МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра вычислительной математики

Лабораторная работа №3

«Исследование устойчивости разностных схем»

Вариант 2

Выполнил:

Бовт Тимофей Анатольевич
студент 4 курса 7 группы

Преподаватель:

Репников Василий Иванович

Минск, 2024 г.

Постановка задачи

Поставлена задача Коши для уравнения переноса

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = 0,\ -\infty < x < \infty,\ t > 0, \\ u(x,0) = u_0(x), \end{cases} \tag{1}$$

где

- $a = 1$ – скорость бегущей волны;
- $u_0(x) = x^2$.

1 Построение разностной схемы

Поставленная задача (1) имеет точное решение. Решением задачи (1) является «бегущая волна»

$$u(x,t) = u_0(x - at).$$

Таким образом, в рамках поставленных условий, задача (1) имеет аналитическое решение

$$u(x,t) = (x-t)^2.$$

Пусть задана равномерная сетка узлов

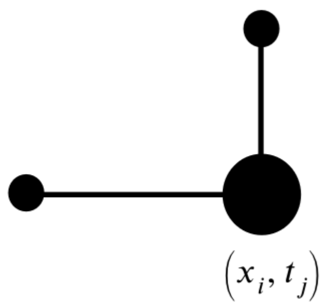
$$\omega_{h\tau} = \omega_h \times \omega_\tau,$$

где

$$\omega_h = \left\{x_k = kh,\ k = 0, \pm 1, \ldots, h > 0\right\},\ \omega_\tau = \left\{t_j = j\tau,\ j = 0, 1, \ldots,\ \tau > 0\right\}.$$

По условию также задан следующий шаблон

$$\text{Ш}(x,t) = \{(x,t),\ (x-h,t),\ (x,t+\tau)\}.$$



Используя предложенный шаблон на заданной сетке узлов построим разностную схему в безиндексной форме, заменяя дифференциальные производные разностными аналогами

$$\begin{cases} y_t + a y_{\overline{x}} = 0,\ (x,t) \in \omega_{h\tau}, \\ y(x,0) = u_0(x),\ x \in \omega_h. \end{cases} \tag{2}$$

Разностная схема (2) также может быть записана в индексной форме в виде

$$\begin{cases} \dfrac{y_k^{j+1} - y_k^j}{\tau} + a\dfrac{y_k^j - y_{k-1}^j}{h} = 0, \ k = \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots, \\ y_k^0 = u_0(x), \ k = 0, \pm 1, \dots . \end{cases} \tag{3}$$

Нужно вычислить погрешность аппроксимации разностной схемы. Поскольку мы имеем одно начальное условие, то погрешность аппроксимации всей схемы будет определяться только погрешностью аппроксимации уравнения. Поэтому для любой точки $(x, t) \in \omega_{h\tau}$ погрешность аппроксимации будет равна

$$\Psi(x,t) = u_t + a u_{\bar{x}} = \frac{\partial u}{\partial t} + \frac{\tau}{2}\frac{\partial^2 u}{\partial t^2} + O(\tau^2) + a\left(\frac{\partial u}{\partial x} + \frac{h}{2}\frac{\partial^2 u}{\partial x^2} + O(h^2)\right) = O(h + \tau),$$

то есть данная разностная схема обладает первым порядком аппроксимации по x и первым порядком аппроксимации по t.

2 Исследование устойчивости разностной схемы спектральным методом

Исследование устойчивости по спектральному методу предусматривает подстановку следующего выражения в разностное уравнение

$$y_k^j = q^j e^{ik\varphi}, \ \varphi \in (0, 2\pi).$$

Итак, подставляя это выражение в разностное уравнение схемы (3), получим

$$\frac{q^{j+1}e^{ik\varphi} - q^j e^{ik\varphi}}{\tau} + a\frac{q^j e^{ik\varphi} - q^j e^{i(k-1)\varphi}}{h} = 0.$$

Сокращая общие множители, получим

$$\frac{q-1}{\tau} + a\frac{1 - e^{-i\varphi}}{h} = 0.$$

Таким образом, можно выразить

$$q = 1 - \gamma(1 - e^{-i\varphi}), \ \gamma = \frac{a\tau}{h}.$$

Далее по спектральному методу для устойчивости необходимо выполнение условия $|q|^2 \leqslant 1$. Рассмотрим это условие

$$\begin{aligned} |q|^2 = |1 - \gamma(1 - \cos\varphi + i\sin\varphi)|^2 = (1 - \gamma(1 - \cos\varphi))^2 + (\gamma\sin\varphi)^2 = \\ = 1 - 2\gamma(1 - \cos\varphi) + \gamma^2(1 - \cos\varphi)^2 + \gamma^2\sin^2\varphi = 1 - 2\gamma(1 - \cos\varphi) + \gamma^2 - 2\gamma^2\cos\varphi + \gamma^2 = \\ = 1 - 2\gamma(1 - \cos\varphi) + 2\gamma^2(1 - \cos\varphi) = 1 + 2\gamma(\gamma - 1)(1 - \cos\varphi) \leqslant 1. \end{aligned}$$

Отсюда

$$2\gamma(\gamma - 1)(1 - \cos\varphi) \leqslant 0.$$

Поскольку $1-\cos\varphi > 0,\ \varphi \in (0, 2\pi)$, то получаем систему условий для устойчивости

$$\begin{cases} \gamma \geqslant 0, \\ \gamma \leqslant 1. \end{cases} \tag{4}$$

То есть при выполнении условий (4) разностная схема будет устойчива по спектральному методу.

Подставляя известное нам значение $a = 1$, получим, что

$$0 \leqslant \frac{\tau}{h} \leqslant 1,$$

или

$$0 \leqslant \tau \leqslant h.$$

3 Исследование устойчивости разностной схемы с помощью принципа максимума

Следуя принципу максимума, в качестве точки для исследования устойчивости возьмем точку (x_i, t_{j+1}). Таким образом, мы можем переписать аппроксимацию основного уравнения переноса

$$\frac{1}{\tau} y_k^{j+1} = \left(\frac{1}{\tau} - \frac{a}{h}\right) y_k^j + \frac{a}{h} y_{k-1}^j.$$

Можем записать коэффициенты, которые требуются для проверки условий устойчивости

$$A(x) = \frac{1}{\tau},\ B_1 = \frac{1}{\tau} - \frac{a}{h},\ B_2 = \frac{a}{h},$$
$$D(x) = A(x) - (B_1 + B_2) \equiv 0,\ F(x) \equiv 0.$$

Проверим, выполняются ли соответствующие условия устойчивости:

$$A(x) = \frac{1}{\tau} > 0, B_1 = \frac{1}{\tau} - \frac{a}{h} \geqslant 0,\ B_2 = \frac{a}{h} > 0,$$

причем второе условие выполняется, когда из условия $B_1 \geqslant 0$ следует, что $\dfrac{a\tau}{h} \leqslant 1$, или, что то же самое,

$$\tau \leqslant \frac{h}{a}, \tag{5}$$

называемое условием Куранта.

Подставляя известное нам значение $a = 1$, получим, что

$$\tau \leqslant h.$$

Таким образом, мы можем считать, что условия устойчивости полученные по спектральному методу и по принципу максимума, совпадают.

4 Машинная реализация разностной схемы

Подключим необходимые библиотеки для работы с векторами и визуализации

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Определим функцию для генерации сеток узлов. Она имеет следующие параметры:

- left_border : левая граница сетки;
- right_border : правая граница сетки;
- num_x_points : число разбиений по x;
- upper_bound : верхняя граница сетки;
- num_t_points : число разбиений по t.

В результате функция возвращает значения:

- nodes_x : сетка узлов ω_h;
- nodes_t : сетка узлов ω_τ;
- h : шаг по x;
- tau : шаг по t.

[2]:
```
def generate_grids(left_border, right_border, num_x_points,␣
 ↪upper_bound, num_t_points):

  h = (right_border-left_border) / num_x_points
  nodes_x = np.linspace(left_border, right_border, num_x_points+1)

  tau = upper_bound / num_t_points
  nodes_t = np.linspace(0, upper_bound, num_t_points+1)

  print('h =', h)
  print('tau =', tau)

  return nodes_x, nodes_t, h, tau
```

Определим функцию

$$u(x,t) = u_0(x - at),$$

соответствующую точному решению поставленной задачи Коши. Зададим $a = 1$ и $u_0(x) = x^2$.

[3]:
```
def u(x, t, a, u_0):
    return u_0(x-a*t)

a = 1
def u_0(x):
    return x**2
```

Определим функцию, реализующую вычисление по разностной схеме в соответствии с рекуррентной формулой

$$\begin{cases} y_k^{j+1} = (1-\gamma)y_k^j + \gamma y_{k+1}^j, \; k = 0 \pm 1, \pm 2 \dots, \; j = 0, 1, \dots, \\ y_k^0 = u_0(x_k), \; k = 0, \pm 1, \dots. \end{cases} \qquad \gamma = \frac{a\tau}{h}.$$

В результате функция возвращает матрицу y, где каждая строка это приближенное решение в каждой точке x в момент времени t, а столбцы – это моменты t.

[4]:
```
def diff_scheme_solve(nodes_x, nodes_t, h, tau, u_0, a):
    gamma = a * tau / h

    y = np.zeros((len(nodes_x), len(nodes_t)))

    for k in range(len(nodes_x)):
        y[k, 0] = u_0(nodes_x[k])

    for k in range(len(nodes_x)-1):
      for j in range(len(nodes_t)-1):
          y[k, j+1] = (1-gamma) * y[k, j] + gamma * y[k+1, j]

    return y
```

Определим следующие сетки узлов. Пусть $x \in [0, 1]$, а отрезок $[0, 1]$ разобъем на 5 частей. Пусть $t \in [0, 0.25]$, а отрезок $[0, 0.25]$ разобъем на 5 частей. Таким образом, получим следующие шаги h и τ.

[5]:
```
nodes_x, nodes_t, h, tau = generate_grids(0, 1, 5, 0.25, 5)
```

```
h = 0.2
tau = 0.05
```

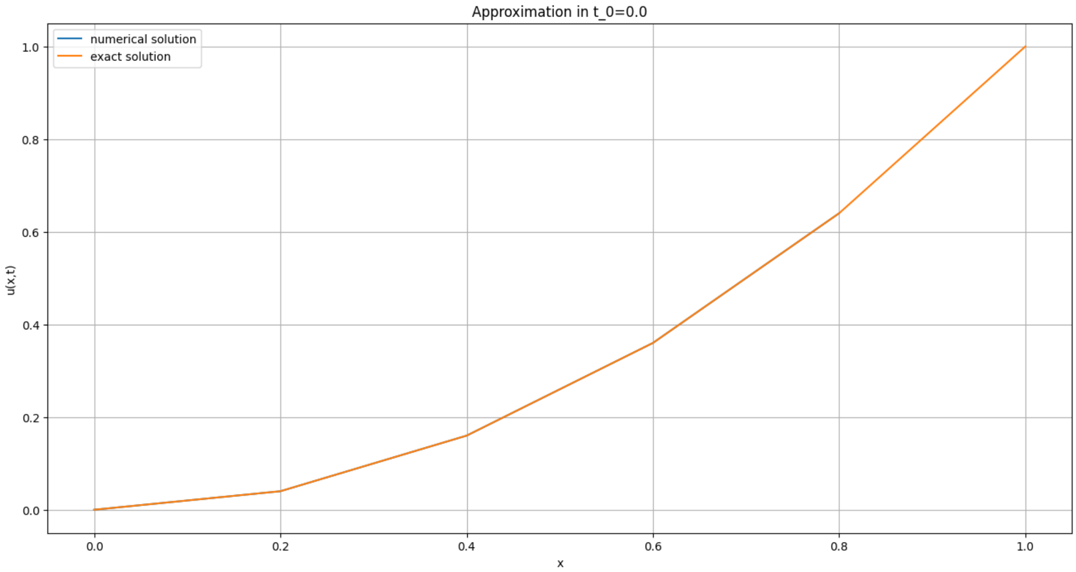
По нашим математическим обоснованиям данная разностная схема должна быть устойчива.

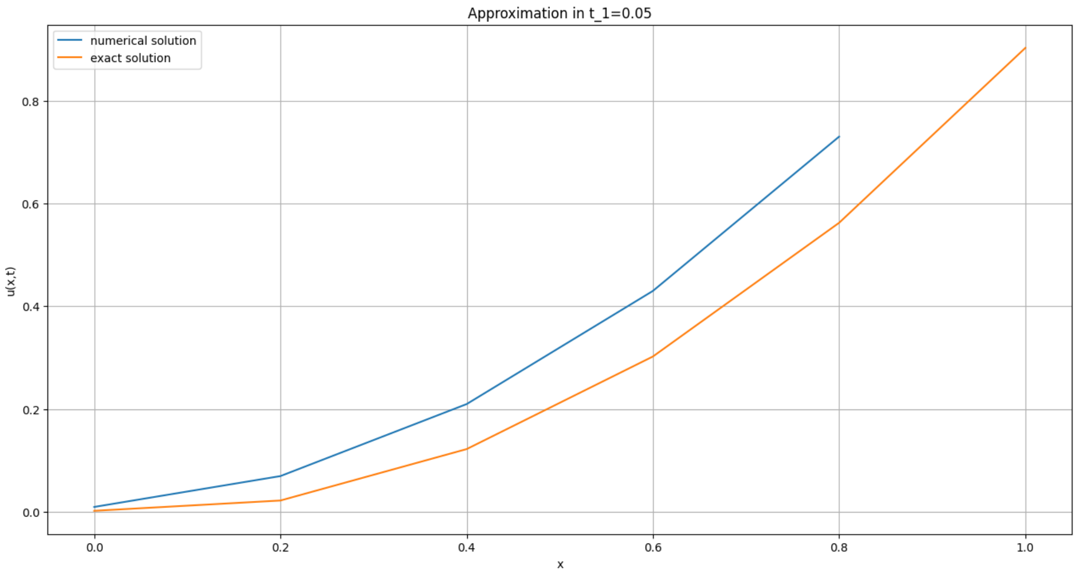
Вычислим значения приближенного решения y по разностной схеме.

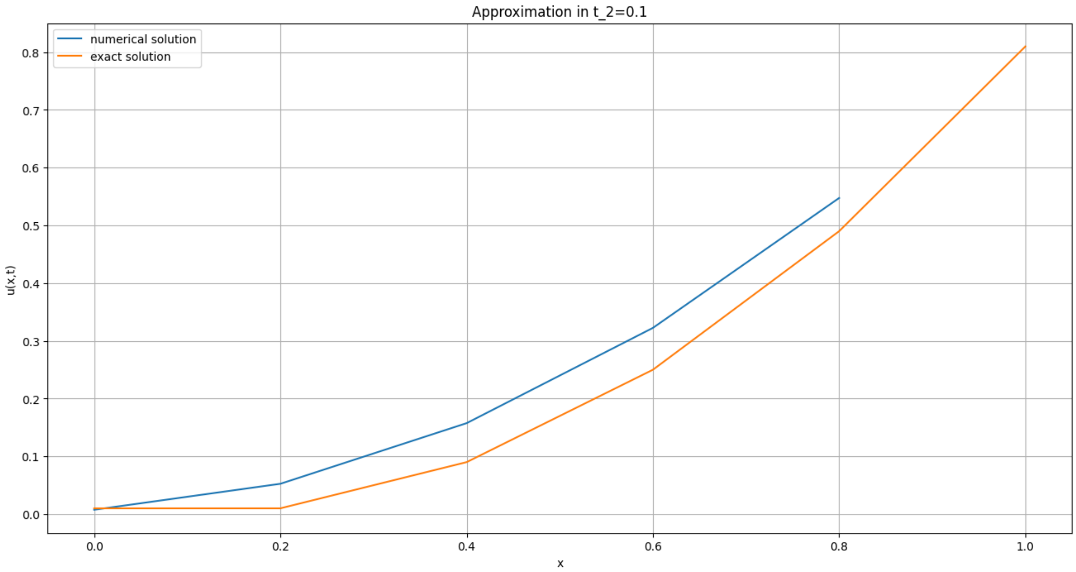
[6]:
```
y = diff_scheme_solve(nodes_x, nodes_t, h, tau, u_0, a)
```

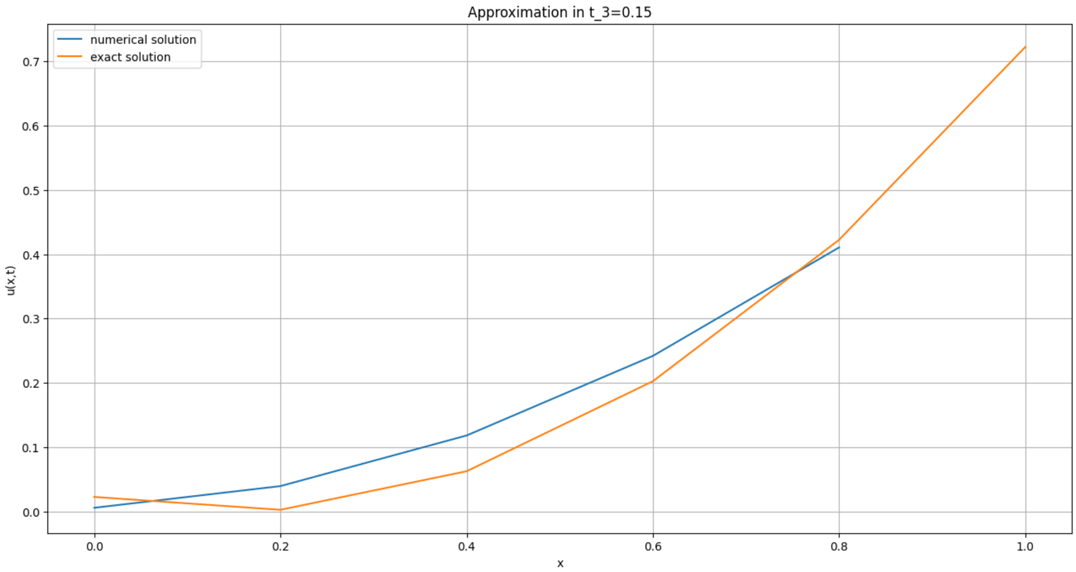
Построим визуализацию полученных результатов. На графике представлены точное и приближенное решение.

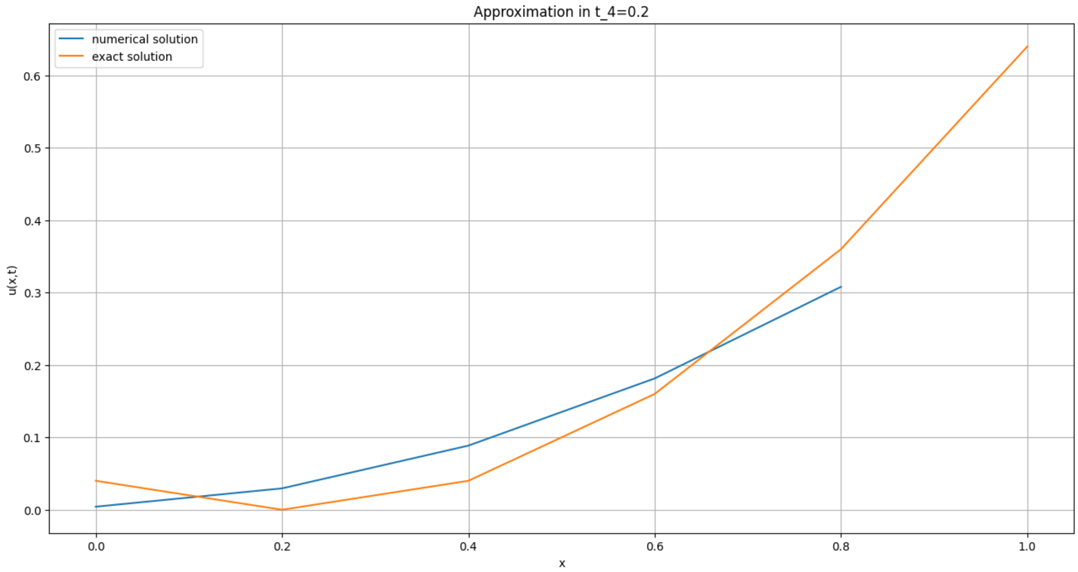
[7]:
```
for j, t in enumerate(nodes_t):
  plt.figure(figsize=(16, 8))
  plt.plot(nodes_x[:-1], y[:-1, j], label='numerical solution')
  plt.plot(nodes_x, u(nodes_x, t, a, u_0), label='exact solution')
  plt.grid(True)
  plt.xlabel('x')
  plt.ylabel('u(x,t)')
  plt.title('Approximation in t_' + str(j) + '=' + str(round(t, 2)))
  plt.legend()
  plt.show()
```

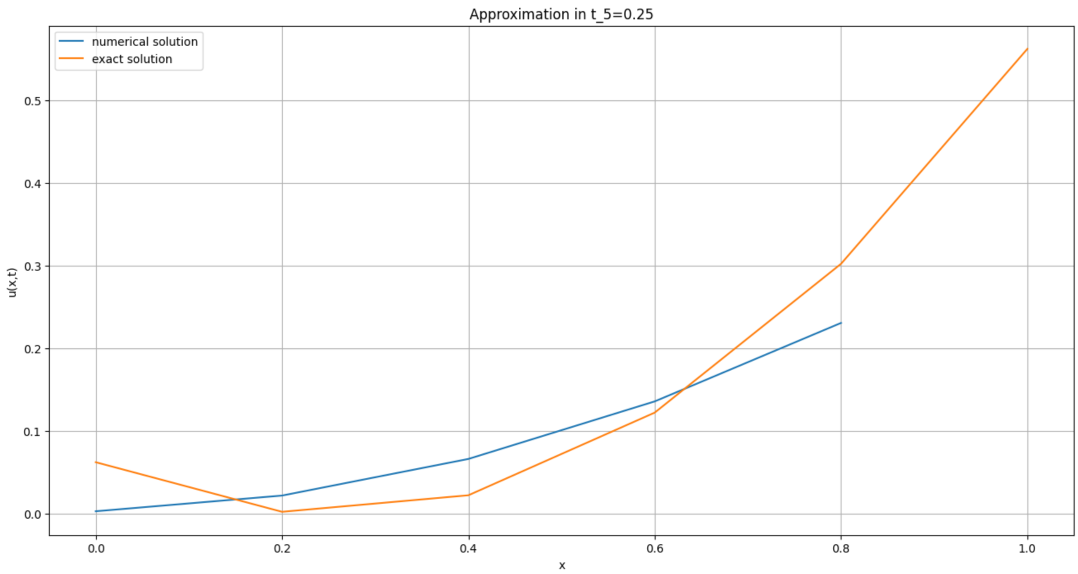

Approximation in t_0=0.0
numerical solution
exact solution
u(x,t)
x
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Approximation in t_1=0.05
numerical solution
exact solution
u(x,t)
x
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Approximation in t_2=0.1
numerical solution
exact solution
u(x,t)
x
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1.0


Approximation in t_3=0.15
numerical solution
exact solution
u(x,t)
x
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1.0





Действительно, построенное по разностной решение приближенно описывает точное решение поставленной дифференциальной задачи. То есть схема действительно оказалась устойчивой.

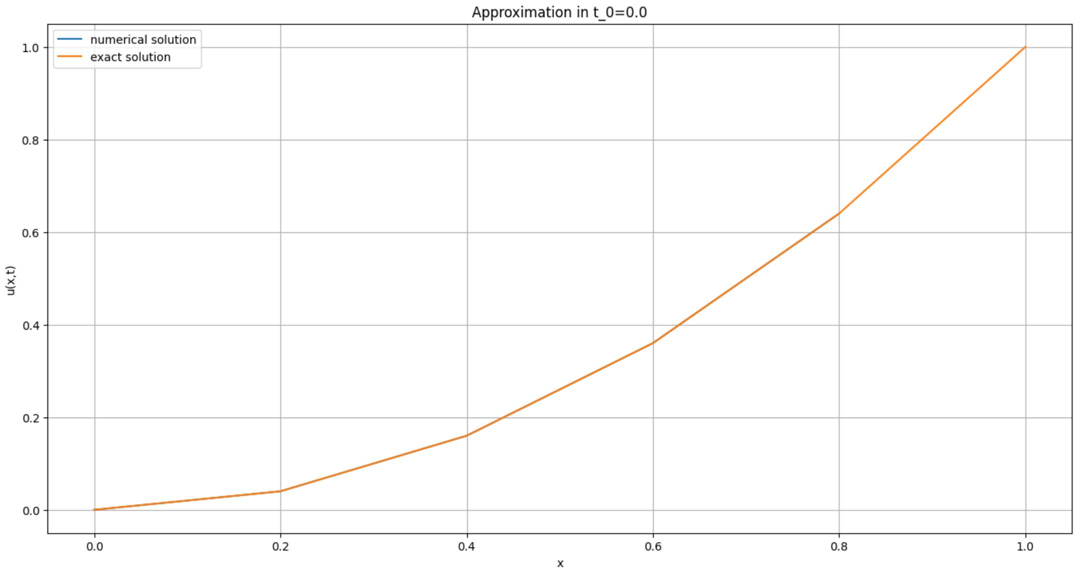
Определим следующие сетки узлов. Пусть $x \in [0, 1]$, а отрезок $[0, 1]$ разобъем на 5 частей. Пусть $t \in [0, 2]$, а отрезок $[0, 2]$ разобъем на 5 частей. Таким образом, получим следующие шаги h и τ.

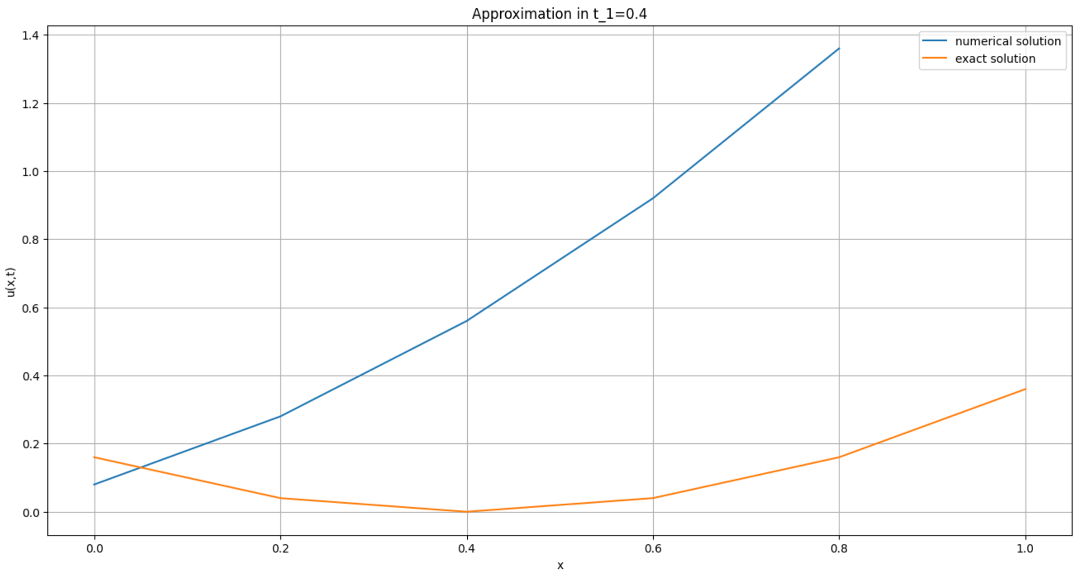
[9]:
```
nodes_x, nodes_t, h, tau = generate_grids(0, 1, 5, 2, 5)
y = diff_scheme_solve(nodes_x, nodes_t, h, tau, u_0, a)
```

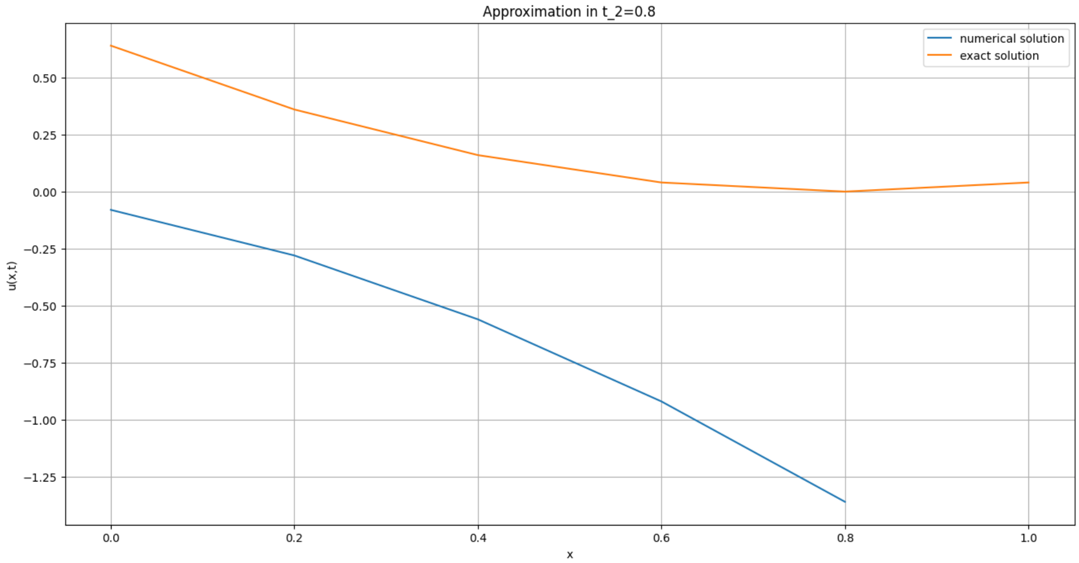
```
h = 0.2
tau = 0.4
```

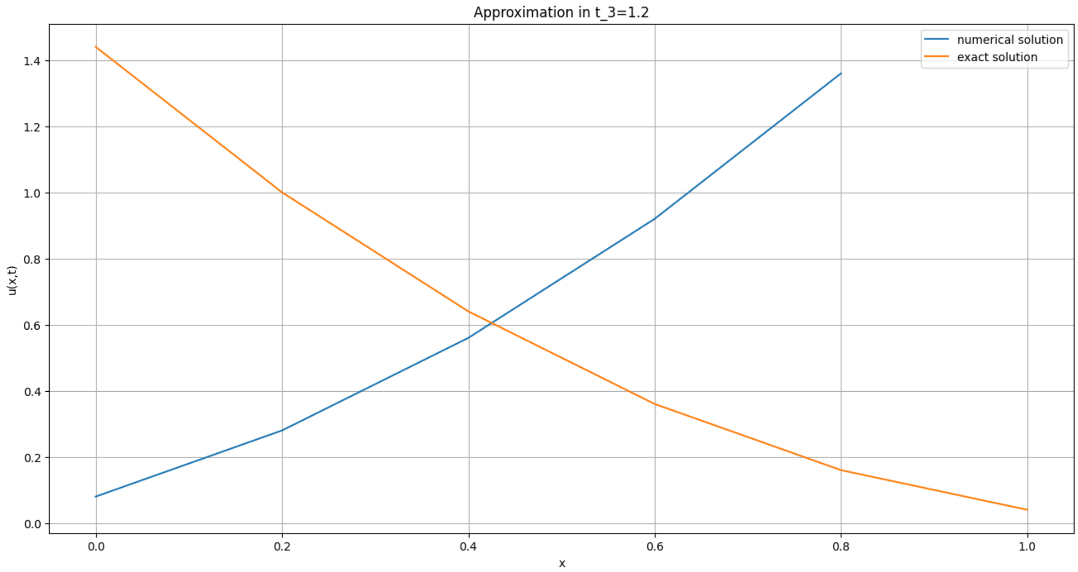
Такая разностная схема должна быть неустойчивой. Рассмотрим графики решений

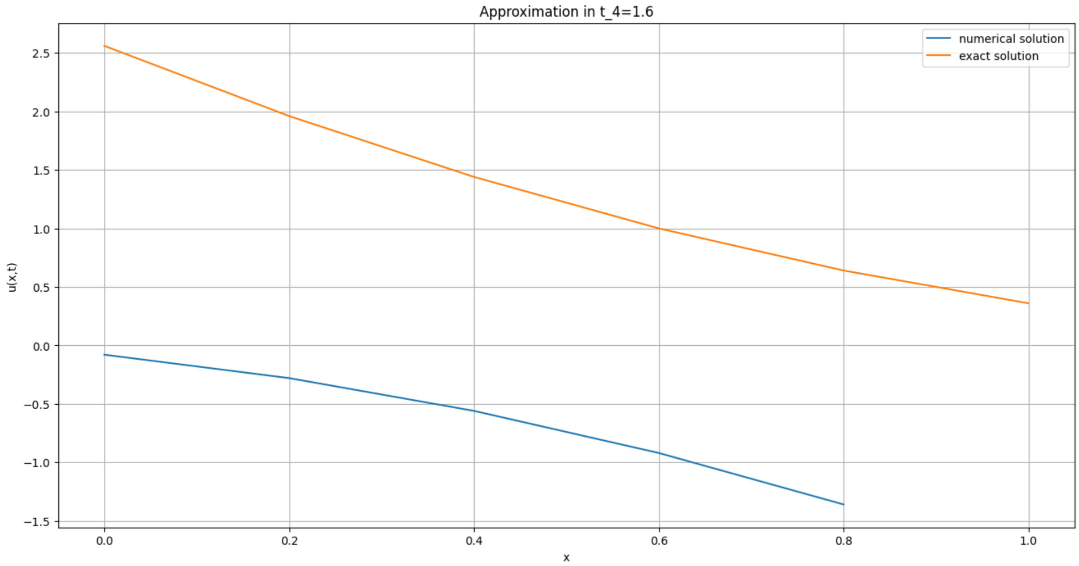
[10]:
```
for j, t in enumerate(nodes_t):
  plt.figure(figsize=(16, 8))
  plt.plot(nodes_x[:-1], y[:-1, j], label='numerical solution')
  plt.plot(nodes_x, u(nodes_x, t, a, u_0), label='exact solution')
  plt.grid(True)
  plt.xlabel('x')
  plt.ylabel('u(x,t)')
  plt.title('Approximation in t_' + str(j) + '=' + str(round(t, 2)))
  plt.legend()
  plt.show()
```

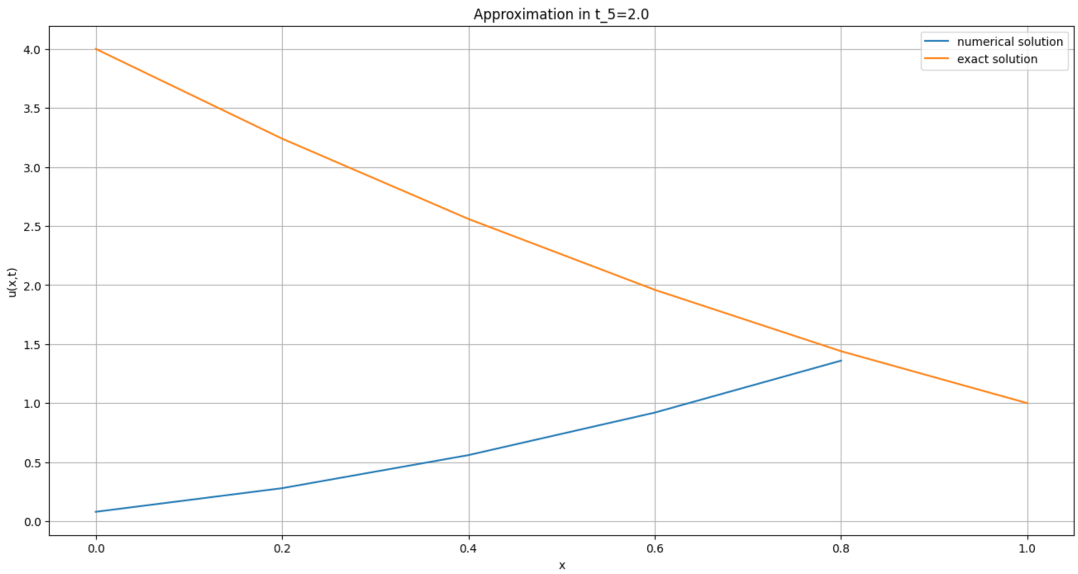





Approximation in t_2=0.8
numerical solution
exact solution
u(x,t)
x
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
0.0
0.2
0.4
0.6
0.8
1.0


Approximation in t_3=1.2
numerical solution
exact solution
u(x,t)
x
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0





Как можно видеть из построенного графика, полученное приближенное решение сильно отклоняется от точного решения, а также на каждом временном слое меняет знак. Из этого мы можем заключить, что разностная схема действительно является неустойчивой.