МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра вычислительной математики

Лабораторная работа №3

«Исследование устойчивости разностных схем»

Вариант 4

Выполнила:

Гут Валерия Александровна
студентка 4 курса 7 группы

Преподаватель:

Репников Василий Иванович

Минск, 2024 г.

Исследование устойчивости разностных схем

Дана дифференциальная задача для уравнения переноса

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = 0,\ 0 < x < \infty,\ t > 0, \\ u(x, 0) = u_0(x),\ x \geq 0 \\ u(0, t) = \mu_0(t),\ t \geq 0 \end{cases} \tag{1}$$

где

- $a = 10$;
- $u_0(x) = x^2$;
- $\mu_0(t) = 100t^2$.

Сразу же проверим условия согласования для корректной постановки задачи:

$$u_0(0) = 0 = \mu_0(0).$$

Определим входные данные компьютерно

[1]:
```python
a = 10

def u_0(x):
    return x**2

def mu_0(t):
    return 100*t**2
```

Построение разностной схемы, погрешность аппроксимации

Для поставленной дифференциальной задачи известно точное решение

$$u(x, t) = \begin{cases} u_0(x - at),\ t \leq \dfrac{x}{a}, \\ \mu_0\left(t - \dfrac{x}{a}\right),\ t \geq \dfrac{x}{a}. \end{cases}$$

Подставляя известные функции, получим точное решение задачи вида

$$u(x, t) = \begin{cases} (x - 10t)^2,\ t \leq \dfrac{x}{10}, \\ 100\left(t - \dfrac{x}{10}\right)^2,\ t \geq \dfrac{x}{10}. \end{cases}$$

Определим компьютерно функцию, соответствующую точному решению.

[2]:
```python
def u(x, t):
    if a * t < x.all():
        return u_0(x - a * t)
    else:
        return mu_0(t - x / a)
```

Пусть задана равномерная сетка узлов

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

где

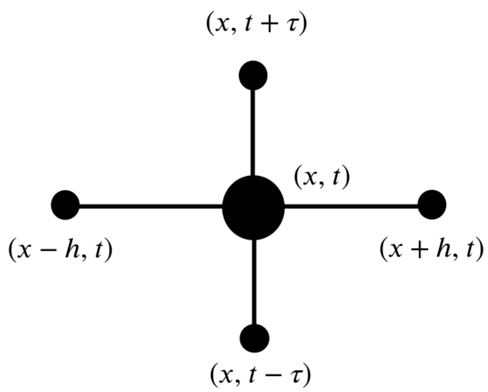
$$\omega_h = \{x_k = kh,\ k = 0, 1, \dots, h > 0\},\ \omega_\tau = \{t_j = j\tau,\ j = 0, 1, \dots,\ \tau > 0\}.$$

Зададим компьютерно сетки узлов. На вход эта функция принимает правую границу для сетки узлов и число разбиений отрезка, а возвращает шаг и узлы сетки.

[3]:
```
import numpy as np

def generate_grid(right_border, num_splits):
    step = right_border / num_splits
    grid = np.linspace(0, right_border, num_splits+1)
    return step, grid
```

По условию также задан следующий шаблон

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



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

$$\begin{cases} y_{\overset{\circ}{t}} + a y_{\overset{\circ}{x}} = 0,\ (x,t) \in \omega_{h\tau}, \\ y(x,0) = u_0(x),\ x \in \omega_h, \\ y(0,t) = \mu_0(t),\ t \in \omega_\tau. \end{cases} \tag{2}$$

Разностная схема также может быть записана в индексной форме в виде

$$\begin{cases} \dfrac{y_k^{j+1} - y_k^{j-1}}{2\tau} + a\dfrac{y_{k+1}^j - y_{k-1}^j}{2h} = 0,\ k = 1, 2, \dots, j = 1, 2 \dots, \\ y_k^0 = u_0(x_k),\ k = 0, 1, \dots, \\ y_0^j = \mu_0(t_j),\ j = 0, 1, \dots, \end{cases} \tag{3}$$

Нужно вычислить погрешность аппроксимации разностной схемы. Начальное и граничное условия аппроксимируются точно, так что погрешность аппроксимации разностной схемы определяется только погрешностью аппроксимации дифференциального уравнения. Поэтому для любой точки $(x,t) \in \omega_{h\tau}$ погрешность аппроксимации будет равна

$$\Psi(x,t) = u_{\overset{\circ}{t}} + a u_{\overset{\circ}{x}} = \frac{\partial u}{\partial t} - \frac{\tau^2}{6} \cdot \frac{\partial^3 u}{\partial^3 t} + O(\tau^4) + \frac{\partial u}{\partial x} - \frac{h^2}{6} \cdot \frac{\partial^3 u}{\partial^3 x} + O(h^4) = O(h^2 + \tau^2),$$

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

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

$$\frac{q^{j+1}e^{ik\varphi} - q^{j-1}e^{ik\varphi}}{2\tau} + a\frac{q^j e^{i(k+1)\varphi} - q^j e^{i(k-1)\varphi}}{2h} = 0.$$

Сокращая общие множители, получим

$$\frac{q - q^{-1}}{2\tau} + a\frac{e^{i\varphi} - e^{-i\varphi}}{2h} = 0.$$

Таким образом, можно выразить

$$q - q^{-1} = \gamma(e^{i\varphi} - e^{-i\varphi}) = 2\gamma i \sin\varphi,\ \gamma = \frac{a\tau}{h}.$$

Домножим на q, перенесем все в одну сторону и получим

$$q^2 - 2q\gamma i \sin\varphi - 1 = 0.$$

Получили квадратное уравнение относительно q. Найдем его корни

$$q = \frac{2\gamma i \sin\varphi \pm \sqrt{-4\gamma^2 \sin^2\varphi + 4}}{2}$$

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

$$|q|^2 = 1 - \gamma^2 \sin^2\varphi + (\gamma \sin\varphi)^2 = 1.$$

Таким образом, по спектральному методу разностная схема устойчива для любого γ.

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

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

Можем записать коэффициенты, которые требуются для проверки условий устойчивости

$$A(x) = \frac{1}{\tau},\ B_1 = \frac{1}{\tau},\ B_2 = -\frac{a}{h},\ B_3 = \frac{a}{h},$$
$$D(x) = A(x) - (B_1 + B_2 + B_3) \equiv 0,\ F(x) \equiv 0.$$

Из-за того, что

$$B_2 = -\frac{a}{h},\ B_3 = \frac{a}{h},$$

а a не может быть одновременно разных знаков, то одно либо B_2, либо B_3 будут отрицательны. Следовательно, по принципу максимума мы не можем утверждать, что разностная схема устойчива.

Реализация разностной схемы

[4]:
```python
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

Для реализации разностной схемы представим ее в виде

$$\begin{cases} y_k^{j+1} = y_k^{j-1} - \gamma(y_{k+1}^j - y_{k-1}^j),\ k = 1, 2, \ldots, j = 1, 2\ldots, \\ y_k^0 = u_0(x_k),\ k = 0, 1, \ldots, \\ y_0^j = \mu_0(t_j),\ j = 0, 1, \ldots, \end{cases} \qquad \gamma = \frac{a\tau}{h}.$$

При реализации сначала строятся значения из начального и граничного условий. То есть мы задаем y_k^j при $j = 0$, $\forall k$, а затем при $k = 0$, $\forall j$. Остальные значения функций можно вычислять по формуле выше.

[5]:
```python
def diff_scheme_solve(x, t, h, tau, u_0, mu_0, a):
    gamma = a * tau / h

    y = np.zeros((len(x), len(t)))

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

    for j in range(len(t)):
        y[0, j] = mu_0(t[j])

    for j in range(len(t)-1):
        for k in range(len(x)-1):
```

```
            y[k, j+1] = y[k, j-1] - gamma * (y[k+1, j] - y[k-1, j])

    return y
```

Теперь сгенерируем сетку с $\tau < h$.

[6]:
```
h, x_grid = generate_grid(0.1, 5)
tau, t_grid = generate_grid(0.01, 5)
```

[7]:
```
h
```

[7]: 0.02

[8]:
```
tau
```

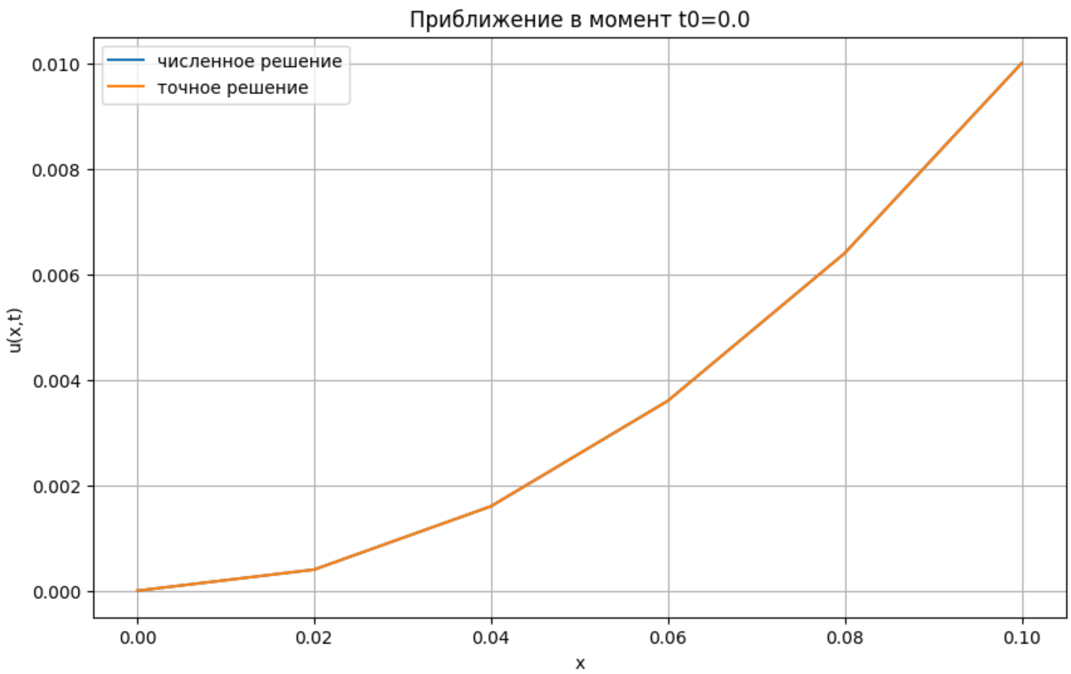
[8]: 0.002

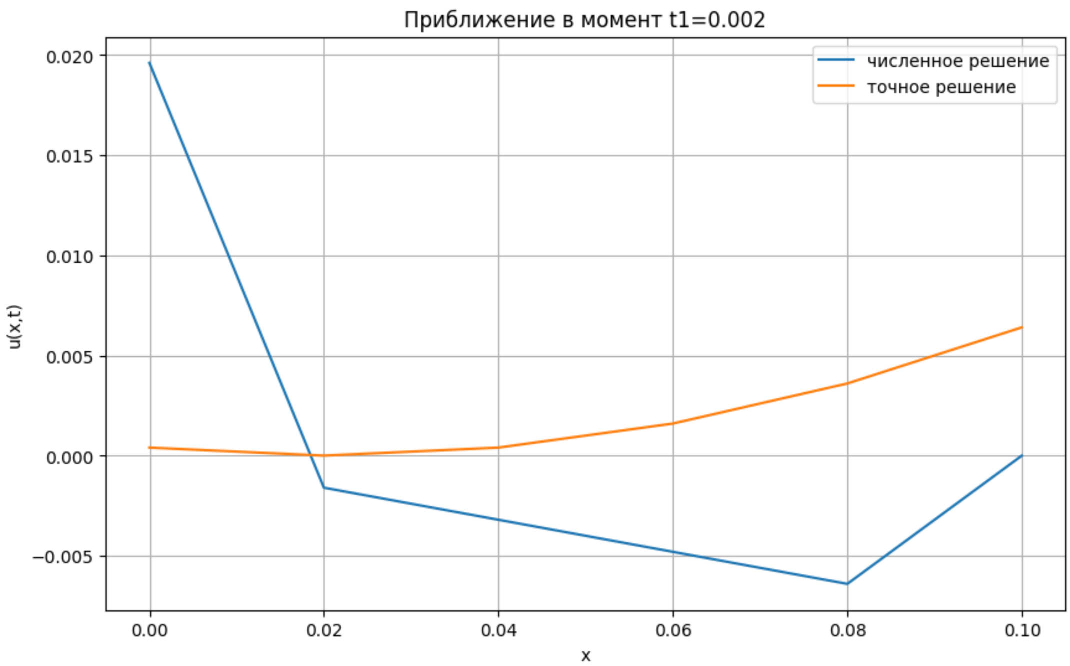
Получим приближенное решение из разностной схемы

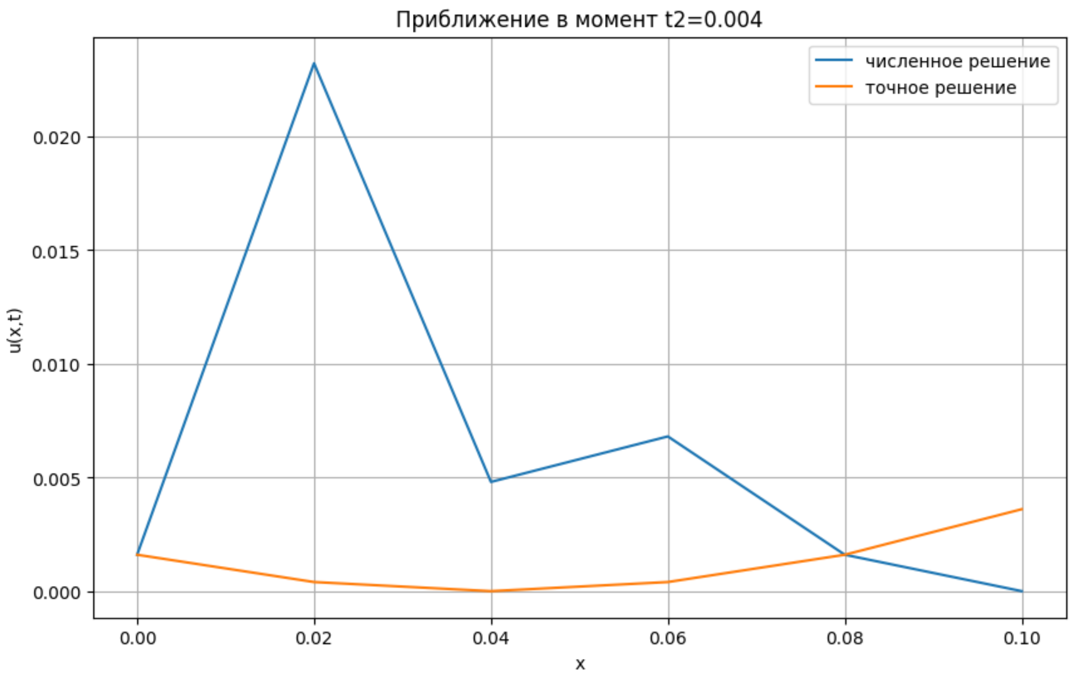
[9]:
```
y = diff_scheme_solve(x_grid, t_grid, h, tau, u_0, mu_0, a)
```

Выведем двумерный график

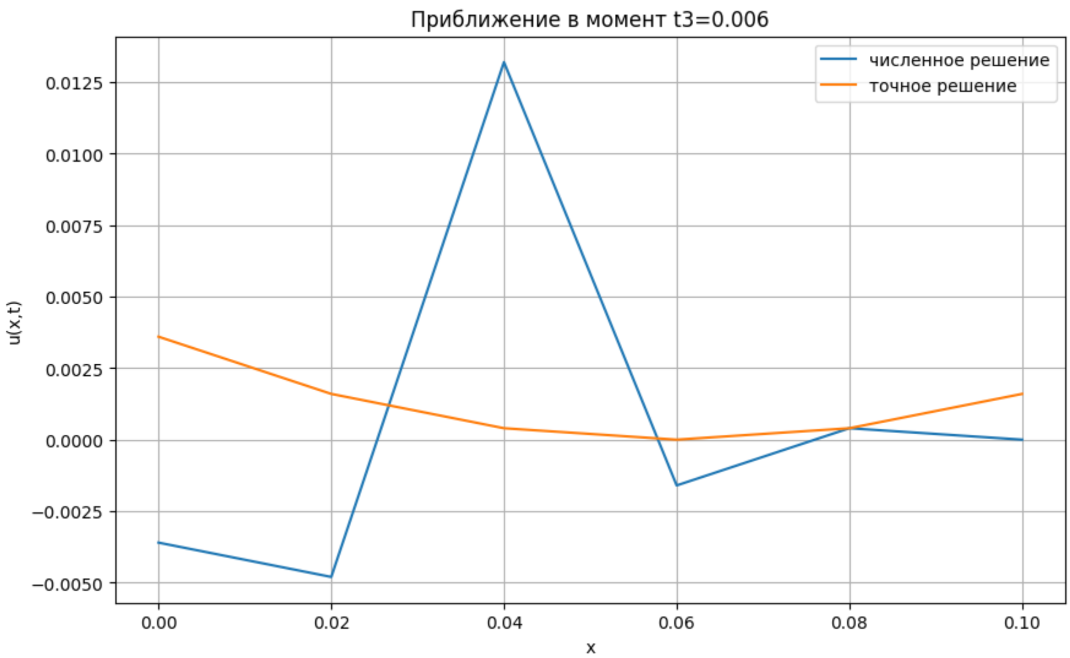
[10]:
```
for j, t in enumerate(t_grid):
    plt.figure(figsize=(10, 6))
    plt.plot(x_grid[:], y[:, j], label='численное решение')
    plt.plot(x_grid, u(x_grid, t), label='точное решение')
    plt.grid(True)
    plt.xlabel('x')
    plt.ylabel('u(x,t)')
    plt.title('Приближение в момент t' + str(j) + '=' + str(round(t,␣
 ↪3)))
    plt.legend()
    plt.show()
```


Приближение в момент t0=0.0
численное решение
точное решение
u(x,t)
x
0.000
0.002
0.004
0.006
0.008
0.010
0.00
0.02
0.04
0.06
0.08
0.10

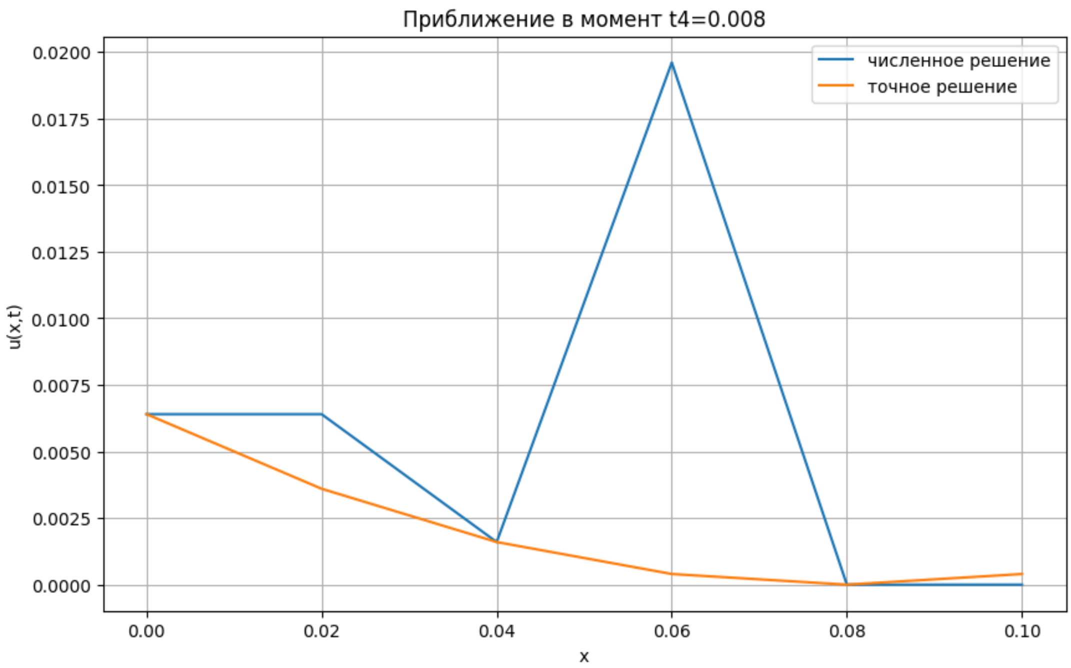

Приближение в момент t1=0.002
численное решение
точное решение
u(x,t)
x
−0.005
0.000
0.005
0.010
0.015
0.020
0.00
0.02
0.04
0.06
0.08
0.10

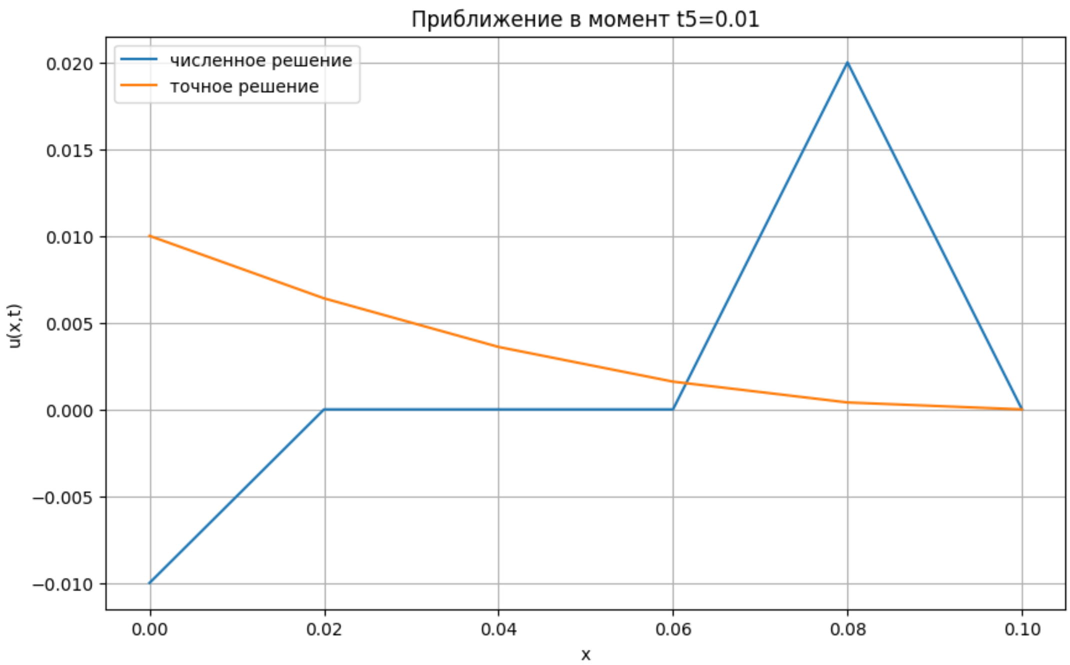


Приближение в момент t2=0.004
численное решение
точное решение
u(x,t)
x



Приближение в момент t3=0.006
численное решение
точное решение
u(x,t)
x

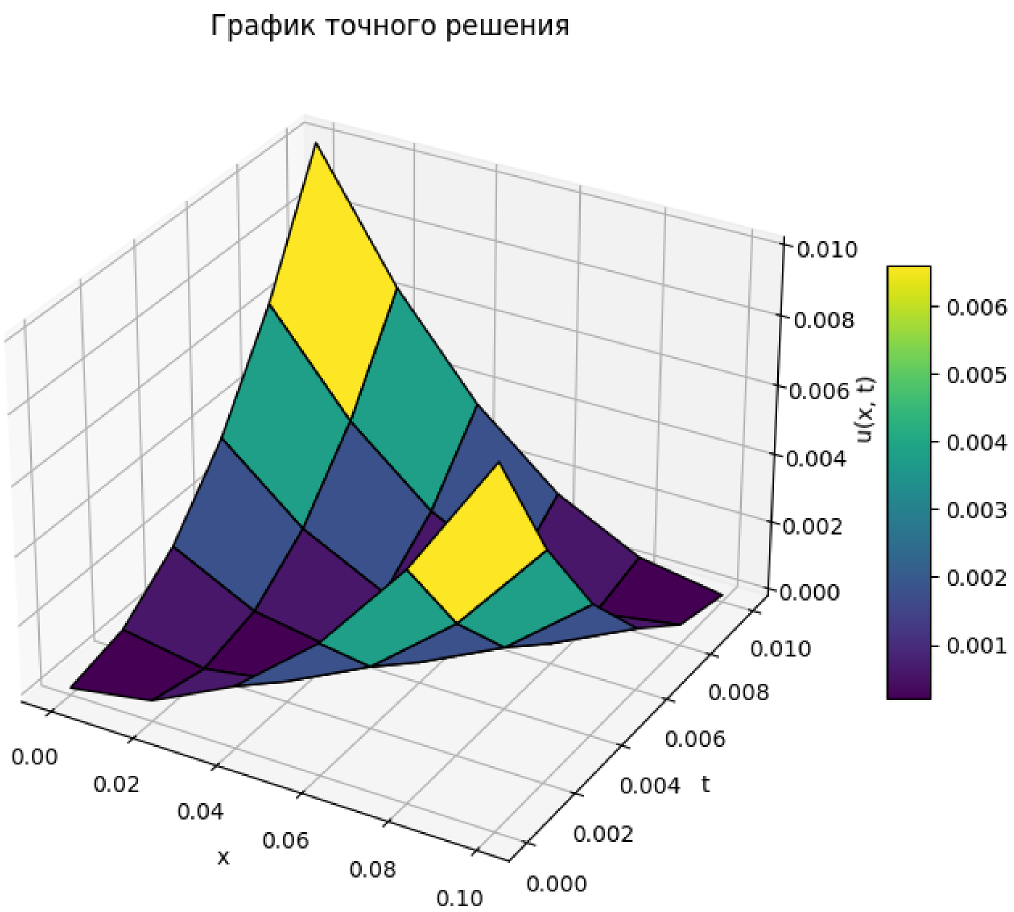




Выведем трехмерные графики точного и приближенного решений

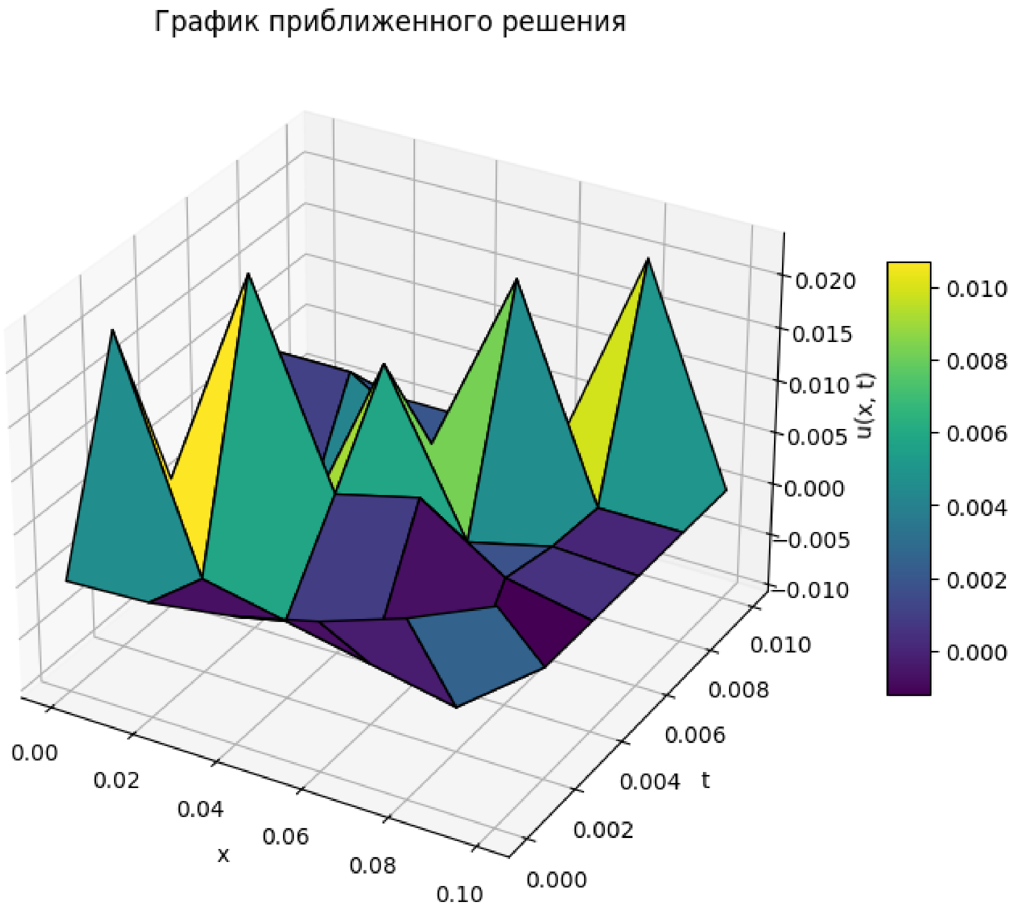
[15]:
```
X, T = np.meshgrid(x_grid, t_grid)
U = np.zeros_like(X)
for i in range(T.shape[0]):
    U[i, :] = u(X[i, :], T[i, 0])
```

```
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

surf = ax.plot_surface(X, T, U, cmap='viridis', edgecolor='k')

ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('u(x, t)')

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10)

plt.title('График точного решения')
plt.show()
```



[16]:
```
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

X, T = np.meshgrid(x_grid, t_grid)
```

```
Y = y.T

surf = ax.plot_surface(X, T, Y, cmap='viridis', edgecolor='k')

ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('u(x, t)')

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10)

plt.title('График приближенного решения')
plt.show()
```



Как можно видеть, по поведению приближенного решения мы можем утверждать, что полученная разностная схема неустойчива.

Рассмотрим теперь случай $h < \tau$.

[18]:
```
h, x_grid = generate_grid(0.01, 5)
tau, t_grid = generate_grid(0.1, 5)
```

[19]:
```
h
```

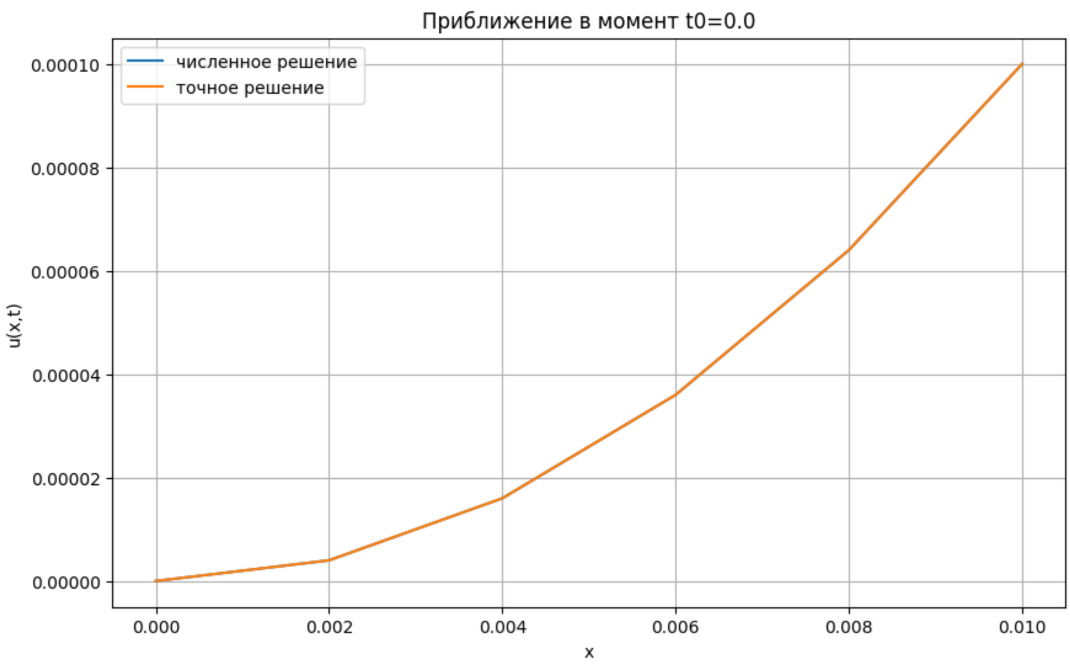
[19]: 0.002

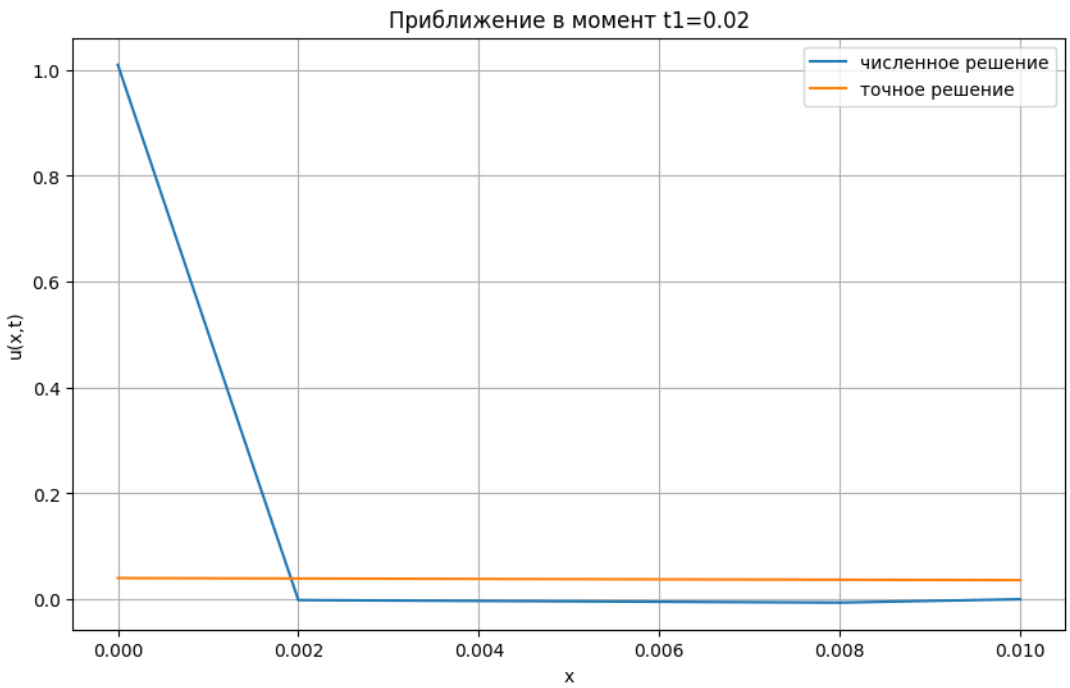
[20]:
```
tau
```

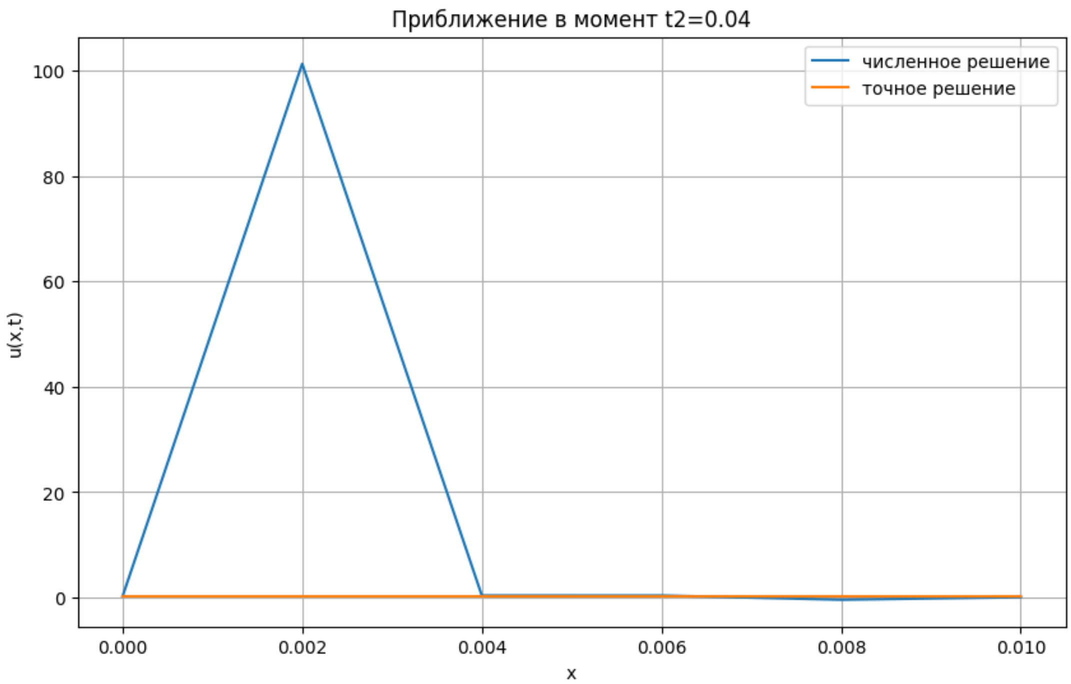
[20]: 0.02

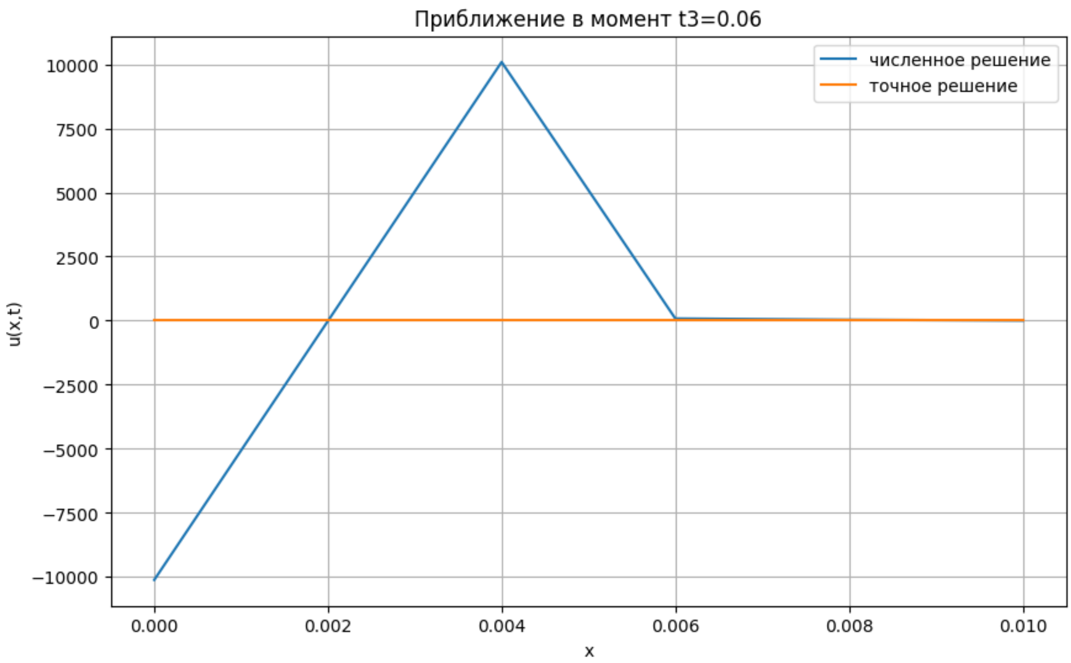
[21]:
```
y = diff_scheme_solve(x_grid, t_grid, h, tau, u_0, mu_0, a)
```

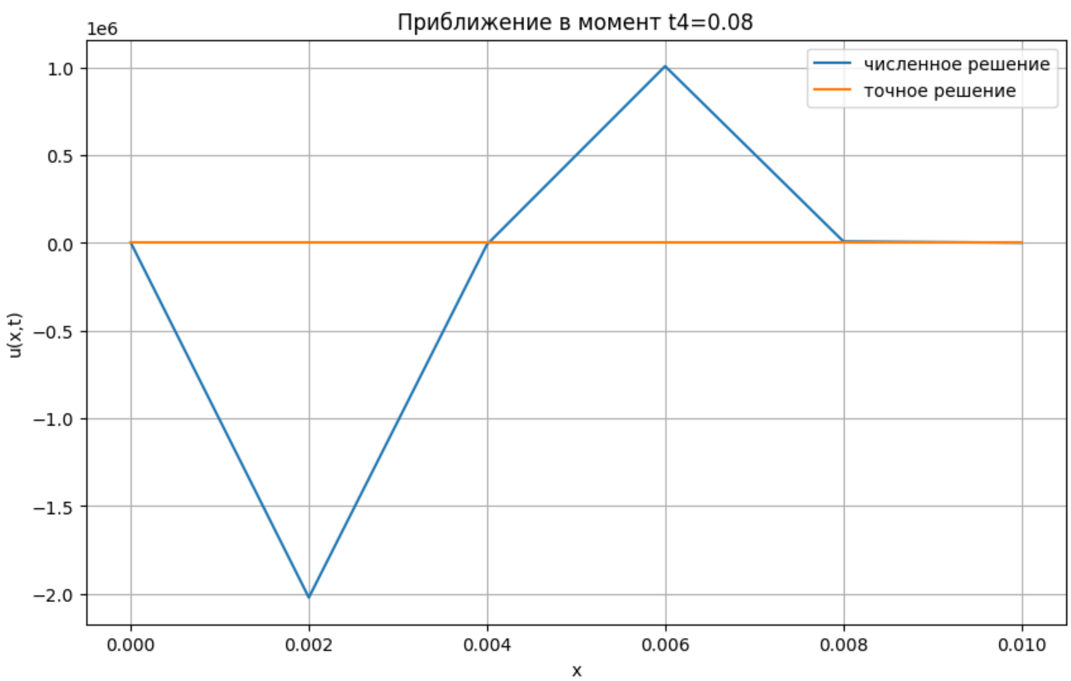
[22]:
```
for j, t in enumerate(t_grid):
    plt.figure(figsize=(10, 6))
    plt.plot(x_grid[:], y[:, j], label='численное решение')
    plt.plot(x_grid, u(x_grid, t), label='точное решение')
    plt.grid(True)
    plt.xlabel('x')
    plt.ylabel('u(x,t)')
    plt.title('Приближение в момент t' + str(j) + '=' + str(round(t,␣
 ↪3)))
    plt.legend()
    plt.show()
```

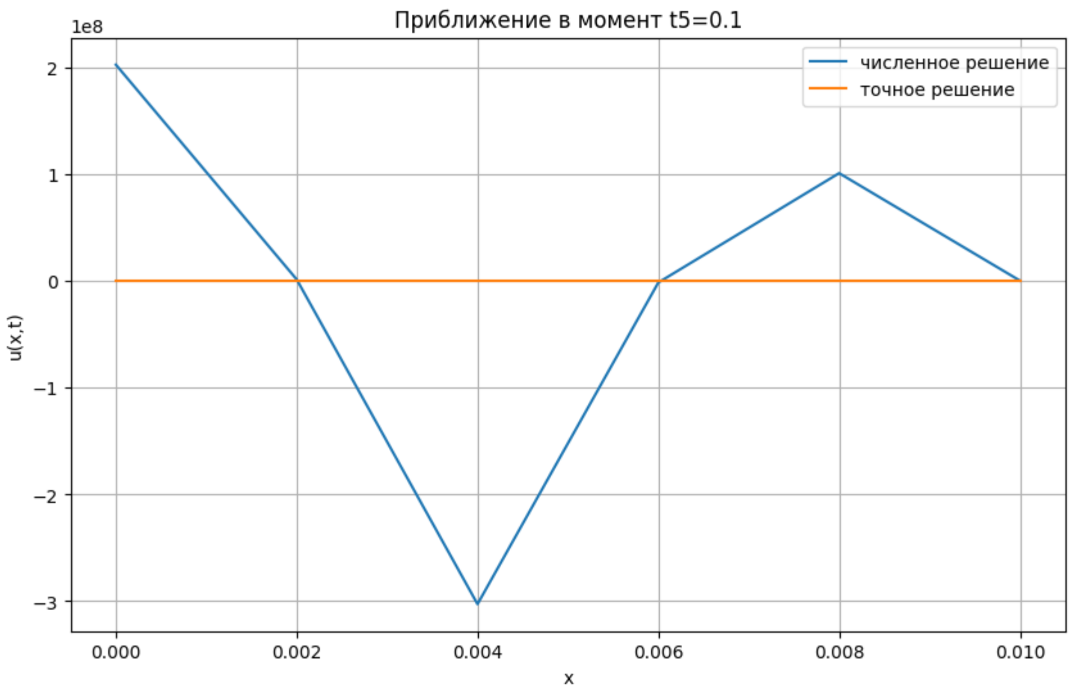



Приближение в момент t1=0.02
численное решение
точное решение
1.0
0.8
0.6
0.4
0.2
0.0
u(x,t)
0.000
0.002
0.004
0.006
0.008
0.010
x

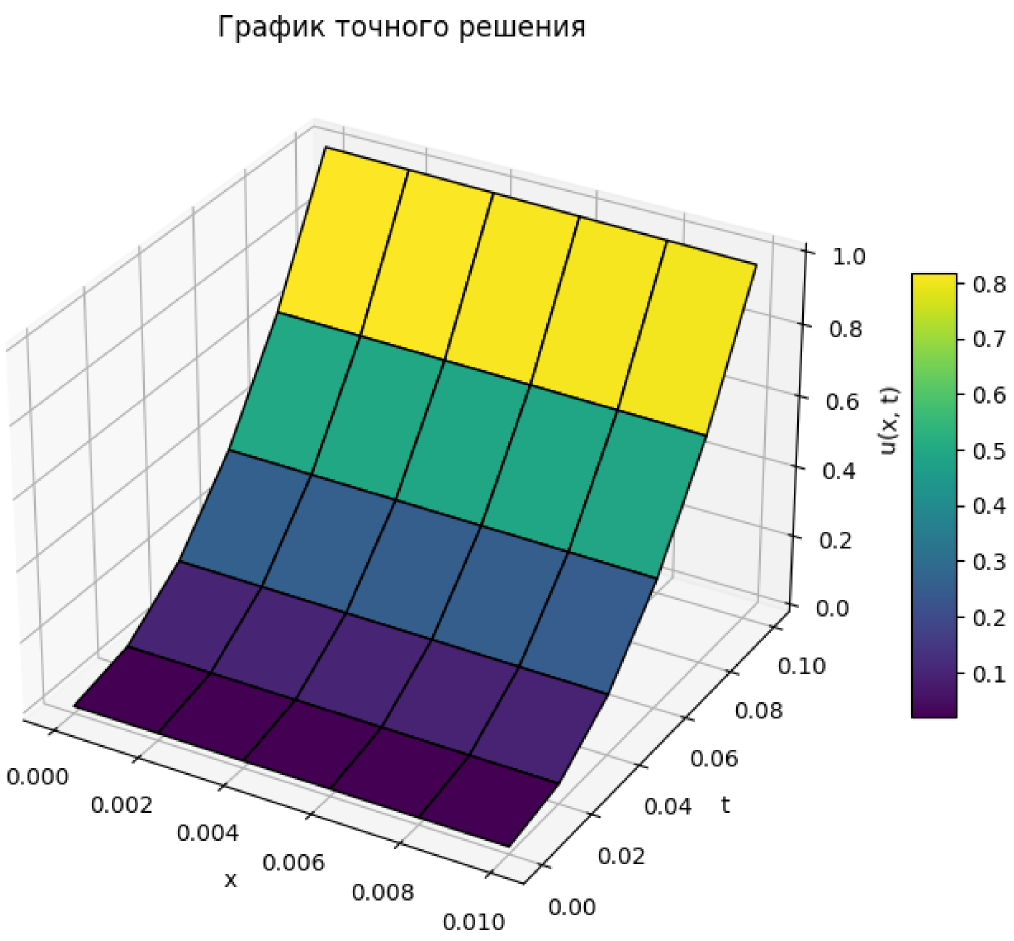

Приближение в момент t2=0.04
численное решение
точное решение
100
80
60
40
20
0
u(x,t)
0.000
0.002
0.004
0.006
0.008
0.010
x


Приближение в момент t3=0.06
численное решение
точное решение
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
u(x,t)
0.000
0.002
0.004
0.006
0.008
0.010
x

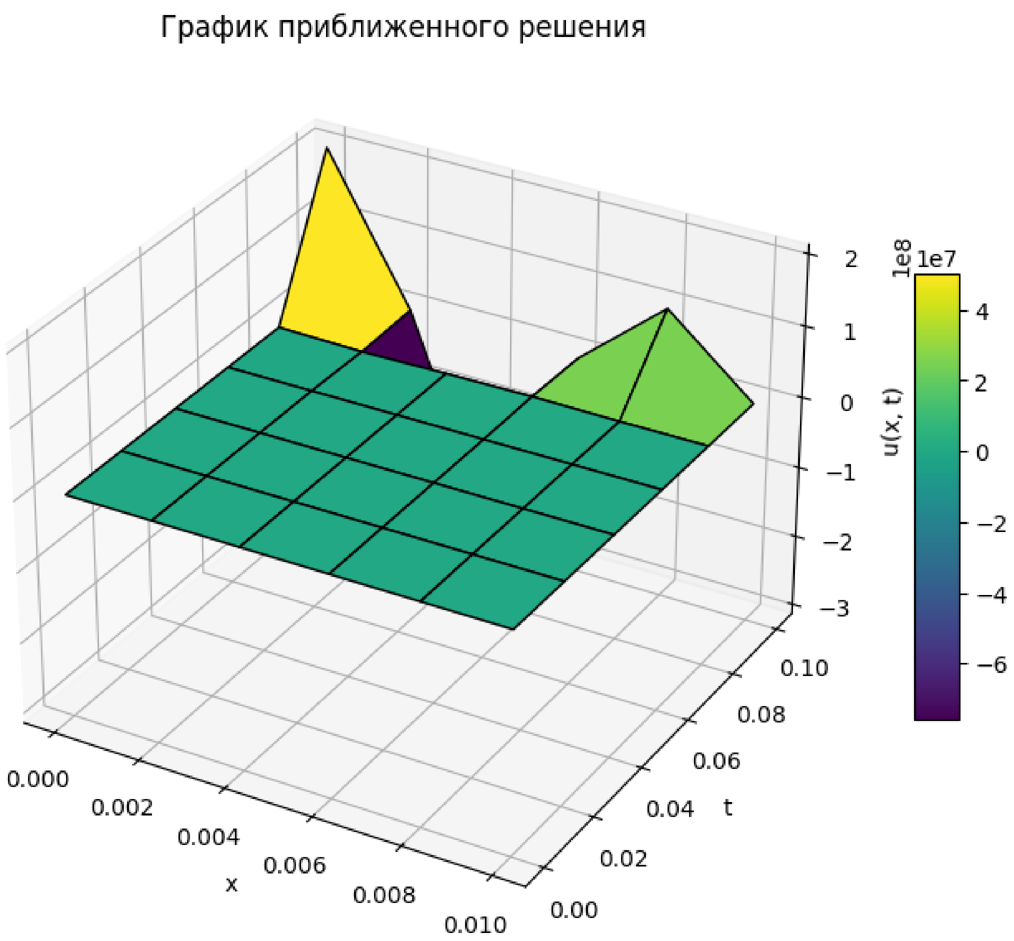

Приближение в момент t4=0.08
1e6
численное решение
точное решение
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
u(x,t)
0.000
0.002
0.004
0.006
0.008
0.010
x



[23]:
```
X, T = np.meshgrid(x_grid, t_grid)
U = np.zeros_like(X)
for i in range(T.shape[0]):
    U[i, :] = u(X[i, :], T[i, 0])

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

surf = ax.plot_surface(X, T, U, cmap='viridis', edgecolor='k')

ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('u(x, t)')

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10)

plt.title('График точного решения')
plt.show()
```



[24]:
```
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

X, T = np.meshgrid(x_grid, t_grid)

Y = y.T

surf = ax.plot_surface(X, T, Y, cmap='viridis', edgecolor='k')

ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('u(x, t)')

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10)

plt.title('График приближенного решения')
plt.show()
```



В данном в случае, как можно видеть, разностная схема также получается неустойчивой.

В итоге, несмотря на то, что по методу разделения переменных разностная схема оказалась устойчивой, практически можно увидеть, что схема является неустойчивой для любых h и τ.