Постановка задачи:

Задание 1.

1. Написать программу для решения указанной задачи, используя явную разностную схему (на 6 баллов).

2. Написать программу для решения указанной задачи, используя разностную схему с заданной в варианте погрешностью аппроксимации (на 2 балла).

3. Сравнить результаты пункта 2 при $h = \tau = 0.05$ и пункта 1, когда $h = 0.05$, а τ выбирается из условия устойчивости (на 2 балла).

Вариант 1.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + x^3 - 6xt, \quad 0 < x < 1, \quad 0 < t < 0.5,$$

$$u(x,0) = -1, \quad 0 < x < 1,$$

$$\left.\frac{\partial u}{\partial x}\right|_{x=0} = t, \quad u(1,t) = 4t - 1, \quad 0 < t < 0.5.$$

$$O(\tau + h^2)$$

Программная реализация:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Заданные параметры
L = 1  # Длина промежутка по x
T = 0.5  # Промежуток времени
nx = 20  # Количество узлов по x
nt = 10  # Количество узлов по времени
hx = L / nx  # Шаг по x
ht = T / nt  # Шаг по времени

# Заданные начальные и граничные условия
u0 = -1
ut = lambda t: 4 * t - 1
ux0 = lambda t: t

# Функция для решения дифференциального уравнения с
использованием явной разностной схемы
def solve_explicit_scheme(L, T, nx, nt, u0, ut, ux0):
    hx = L / nx
    ht = T / nt

    # Инициализация сетки
    x_values = np.linspace(0, L, nx + 1)
    t_values = np.linspace(0, T, nt + 1)
    u = np.zeros((nt + 1, nx + 1))

    # Начальное условие
```

```
    u[0, :] = u0

    # Граничные условия
    u[:, -1] = ut(t_values)

    # Разностная схема
    r = ht / hx ** 2
    for n in range(nt):
        for i in range(1, nx):
            u[n + 1, i] = u[n, i] + r * (u[n, i + 1] - 2 * u[n,
i] + u[n, i - 1]) + ht * (x_values[i] ** 3 - 6 * x_values[i] *
t_values[n])

        # Граничное условие на левом краю (производная)
        u[n + 1, 0] = u[n + 1, 1] - hx * ux0(t_values[n + 1])

    return x_values, t_values, u

# Решение с использованием явной разностной схемы
x_values_explicit, t_values_explicit, u_explicit =
solve_explicit_scheme(L, T, nx, nt, u0, ut, ux0)

# Визуализация результатов
X, T = np.meshgrid(x_values_explicit, t_values_explicit)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, T, u_explicit, cmap='viridis')
ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('u')
ax.set_title('Solution using Explicit Scheme')
plt.show()

# Вывод решения в виде набора двумерных графиков-кривых
fix_times_row1 = [0, 0.1, 0.2]  # Фиксированные моменты времени
для первого ряда
fix_times_row2 = [0.3, 0.4, 0.5]  # Фиксированные моменты
времени для второго ряда

fig, axs = plt.subplots(2, len(fix_times_row1), figsize=(15,
10))

# Первый ряд графиков
for i, t_fix in enumerate(fix_times_row1):
    # Находим ближайший временной шаг к заданному моменту
времени
    idx_time = np.abs(t_values_explicit - t_fix).argmin()
    t_plot = t_values_explicit[idx_time]
    u_plot = u_explicit[idx_time]

    # Построение графика значения функции u(x) в заданный момент
времени
    axs[0, i].plot(x_values_explicit, u_plot, label=f't =
```

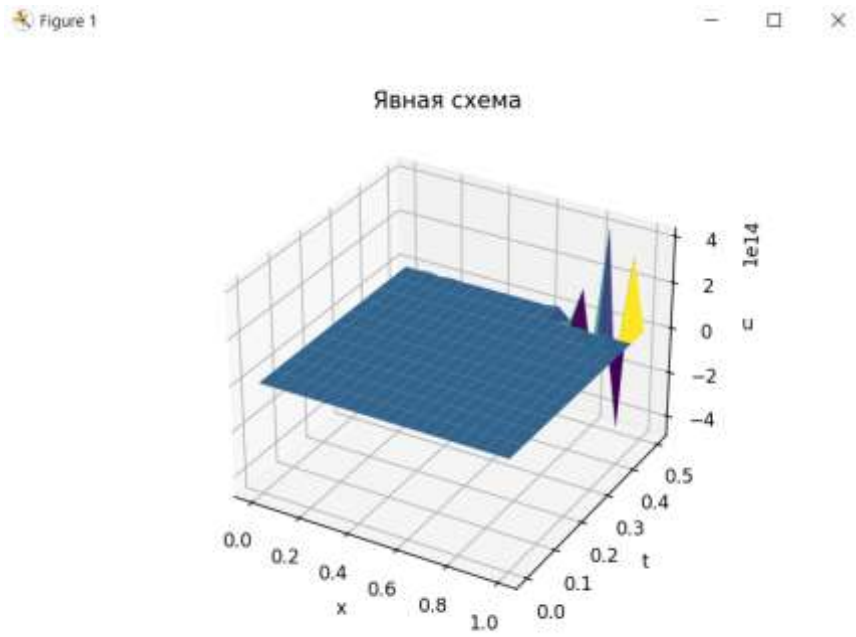
```
{t_plot}')
    axs[0, i].set_xlabel('x')
    axs[0, i].set_ylabel('u(x)')
    axs[0, i].set_title(f'Solution at t = {t_plot}')
    axs[0, i].grid(True)
    axs[0, i].legend()

# Второй ряд графиков
for i, t_fix in enumerate(fix_times_row2):
    # Находим ближайший временной шаг к заданному моменту
времени
    idx_time = np.abs(t_values_explicit - t_fix).argmin()
    t_plot = t_values_explicit[idx_time]
    u_plot = u_explicit[idx_time]

    # Построение графика значения функции u(x) в заданный момент
времени
    axs[1, i].plot(x_values_explicit, u_plot, label=f't =
{t_plot}')
    axs[1, i].set_xlabel('x')
    axs[1, i].set_ylabel('u(x)')
    axs[1, i].set_title(f'Solution at t = {t_plot}')
    axs[1, i].grid(True)
    axs[1, i].legend()

plt.tight_layout()
plt.show()
```

Полученные результаты:



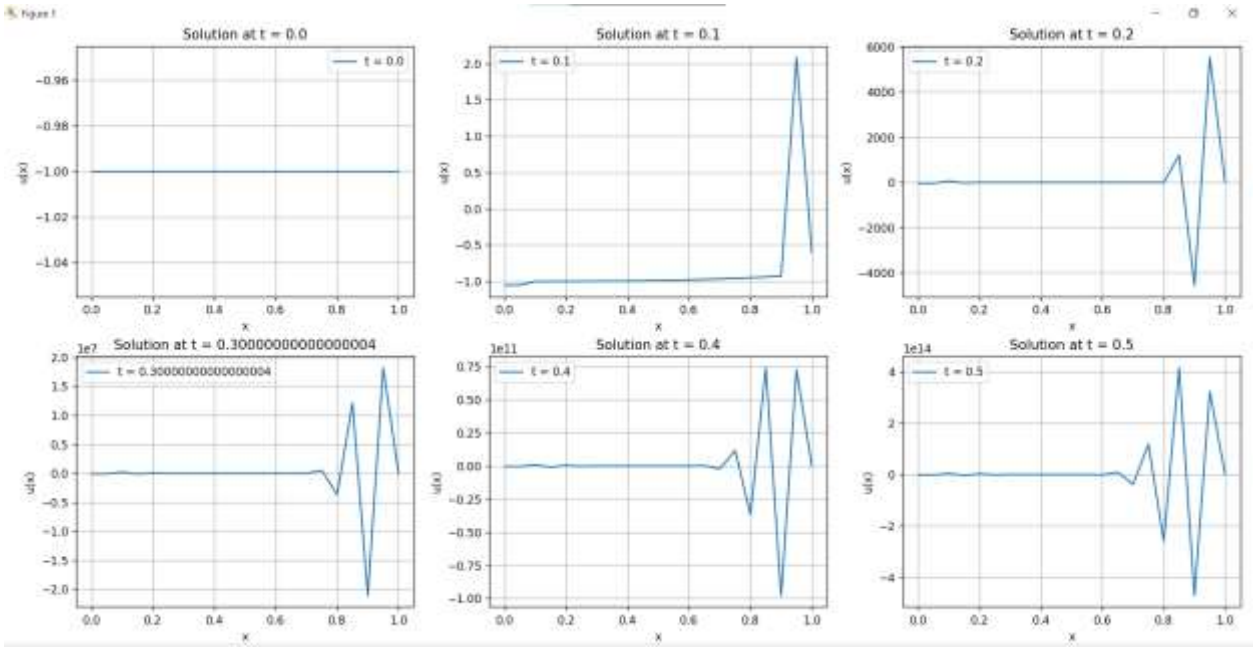


Figure 1
Solution at t = 0.0
t = 0.0
u(x)
x
Solution at t = 0.1
t = 0.1
u(x)
x
Solution at t = 0.2
t = 0.2
u(x)
x
1e7
Solution at t = 0.30000000000000004
t = 0.30000000000000004
u(x)
x
1e11
Solution at t = 0.4
t = 0.4
u(x)
x
1e14
Solution at t = 0.5
t = 0.5
u(x)
x