

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ

**Лабораторная работа №4**

**«Разностная схема для уравнения теплопроводности»**

**Вариант 2**

Выполнил:

Бовт Тимофей Анатольевич  
студент 4 курса 7 группы

Преподаватель:

Репников Василий Иванович

Минск, 2024 г.

# Постановка задачи 1

Поставлена третья краевая задача для уравнения теплопроводности с постоянными коэффициентами

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \sqrt{2} \sin\left(\frac{\pi}{4} - x - t\right), & t > 0, 0 < x < 1, \\ u(x, 0) = \cos x, & 0 \leq x \leq 1, \\ \frac{\partial u(0, t)}{\partial x} - u(0, t) = -\sqrt{2} \sin\left(\frac{\pi}{4} + t\right), & t \geq 0 \\ u(1, t) = \cos(t + 1), & t \geq 0. \end{cases} \quad (1)$$

## Построение разностной схемы

Сперва проверим корректную постановку данной дифференциальной задачи, для чего рассмотрим условия согласования:

$$\begin{aligned} \frac{\partial \cos x}{\partial x} - \cos x|_{x=0} &= -1 = -\sqrt{2} \sin\left(\frac{\pi}{4} + t\right)|_{t=0}, \\ \cos x|_{x=1} &= \cos 1 = \cos(t + 1)|_{t=0}. \end{aligned}$$

Таким образом, условия согласования выполнены. Также для данной задачи выполняется принцип максимума, который позволяет заключить, что если решение задачи существует, то оно единственно. Следовательно, дифференциальная задача является корректно поставленной.

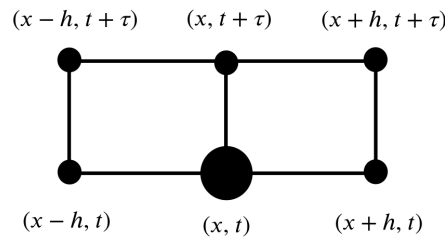
Поставленная дифференциальная задача имеет точное решение

$$u(x, t) = \cos(x + t).$$

Зададим равномерную по каждому направлению сетку

$$\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau, \quad \bar{\omega}_h = \{x_k = kh, \quad k = \overline{0, N}, \quad Nh = 1\}, \quad \bar{\omega}_\tau = \{t_j = j\tau, \quad j = \overline{0, N_1}, \quad N_1\tau = T\}.$$

На этой сетке зададим шеститочечный шаблон



Используя данный шаблон и заменяя дифференциальные операторы разностными, можем записать разностную схему для поставленной задачи в бензиндексной форме

$$\begin{cases} y_t = \Lambda(\sigma \hat{y} + (1 - \sigma)y) + \varphi, & (x, t) \in \omega_{h\tau}, \\ y(x, 0) = \cos x, & x \in \bar{\omega}_h, \\ y_x^{(\sigma)}(0, t) = y^{(\sigma)}(0, t) + \frac{h}{2}y_t(0, t) - \tilde{\mu}_0(t), & t \in \bar{\omega}_\tau, \\ y(1, t) = \cos(t + 1), & t \in \bar{\omega}_\tau. \end{cases} \quad (2)$$

Здесь

$$\Lambda y = y_{\bar{x}x}, \quad \hat{y} = y(x, t + \tau),$$

$\sigma$  – это параметр, а  $\varphi$  и  $\tilde{\mu}_0(t)$  – это сеточные функция, аппроксимирующая функции  $f$  и  $\mu_0(t)$  соответственно, определенную в задаче (1).

# Исследование аппроксимации и устойчивости разностной схемы

## Аппроксимация разностного уравнения

Исследуем погрешность  $\psi(x, t)$  аппроксимации разностного уравнения. Для упрощения записи будем обозначать

$$\dot{u} = \frac{\partial u(x, t)}{\partial t}, \quad u' = \frac{\partial u(x, t)}{\partial x}.$$

Берем любую точку сетки  $(x, t)$  и рассматриваем величину  $\psi(x, t)$  в этой точке

$$\begin{aligned} \psi(x, t) &= \varphi + \sigma \left( \dot{u}'' + \frac{h^2}{12} \dot{u}'''' + O(h^4) \right) + (1 - \sigma) \left( u'' + \frac{h^2}{12} u'''' + O(h^4) \right) - \left( \dot{u} + \frac{\tau}{2} \ddot{u} + O(\tau^2) \right) = \\ &= \varphi + \sigma \left( u'' + \tau \dot{u}'' + \frac{h^2}{12} u'''' + O(\tau^2 + h^4) \right) + (1 - \sigma) \left( u'' + \frac{h^2}{12} u'''' + O(h^4) \right) - \left( \dot{u} + \frac{\tau}{2} \ddot{u} + O(\tau^2) \right) = \\ &= \varphi + (u'' - \dot{u}) + \sigma \tau \dot{u}'' + \frac{h^2}{12} u''' - \frac{\tau}{2} \ddot{u} + O(\tau^2 + h^4). \end{aligned}$$

Учтем вид дифференциального оператора. По условию

$$\dot{u} = u'' + f, \quad \ddot{u} = \dot{u}'' + \dot{f}, \quad \dot{u}'' = u'''' + f'',$$

тогда получим выражение для аппроксимации разностной схемы

$$\psi(x, t) = \varphi - f + \tau \dot{u}'' \left( \sigma - \frac{1}{2} + \frac{h^2}{12\tau} \right) - \frac{\tau}{2} \dot{f} - \frac{h^2}{12} f'' + O(\tau^2 + h^4).$$

## Аппроксимация граничного условия

Исследуем погрешность аппроксимации левого граничного условия

$$\nu(0, t) = u_x^{(\sigma)}(0, t) - u^{(\sigma)}(0, t) - \frac{h}{2} u_t(0, t) + \tilde{\mu}_0(t).$$

Так как

$$\begin{aligned} u^{(\sigma)}(0, t) &= \sigma u(0, t + \tau) + (1 - \sigma) u(0, t) = \sigma [u(0, t) + \tau \dot{u}(0, t) + O(\tau^2)] + (1 - \sigma) u(0, t) = \\ &= u(0, t) + \sigma \tau \dot{u}(0, t) + O(\tau^2), \end{aligned}$$

то

$$\begin{aligned} u_x^{(\sigma)}(0, t) &= u'(0, t) + \frac{h}{2} u''(0, t) + \frac{h^2}{6} u'''(0, t) + \frac{h^3}{24} u^{(IV)}(0, t) + O(h^4) + \\ &+ \sigma \tau \left[ u'(0, t) + \frac{h}{2} u''(0, t) + O(h^2) \right] + O(\tau^2) \end{aligned}$$

и, следовательно,

$$\begin{aligned} \nu(0, t) &= u'(0, t) + \frac{h}{2} u''(0, t) + \frac{h^2}{6} u'''(0, t) + \frac{h^3}{24} u^{(IV)}(0, t) + O(h^4) + \sigma \tau \left[ u'(0, t) + \frac{h}{2} u''(0, t) + O(h^2) \right] + \\ &+ O(\tau^2) - \beta_0 [u(0, t) + \sigma \tau \dot{u}(0, t) + O(\tau^2)] - \left[ \frac{h}{2} \dot{u}(0, t) + \frac{\tau}{2} \ddot{u}(0, t) + O(\tau^2) \right] + \tilde{\mu}_0(t). \end{aligned}$$

Тогда, используя выражения из исходного уравнения и граничного условия, получим

$$\begin{aligned}
\nu(0, t) &= \tilde{\mu}_0(t) + \beta_0 u(0, t) - \mu_0(t) + \frac{h}{2} [\dot{u}(0, t) - f(0, t)] + \frac{h^2}{6} [\beta_0 \dot{u}(0, t) - \dot{\mu}_0(t) - f'(0, t)] + \\
&+ \frac{h^3}{24} [\ddot{u}(0, t) - f''(0, t)] + \sigma\tau [\beta_0 \dot{u}(0, t) - \dot{\mu}_0(t)] + \sigma\tau \frac{h}{2} \ddot{u}(0, t) - \\
&- \beta_0 u(0, t) - \beta_0 \sigma\tau \dot{u}(0, t) - \frac{h}{2} \dot{u}(0, t) - \frac{\tau}{2} \cdot \frac{h}{2} [\ddot{u}(0, t) + f'(0, t)] + O(\tau^2 + h^4) = \\
&= \tilde{\mu}_0(t) - \mu_0(t) - \frac{h}{2} f(0, t) - \sigma\tau \dot{\mu}_0(t). \\
&+ \frac{h^2}{6} \beta_0 \dot{u}(0, t) - \frac{h^2}{6} \dot{\mu}_0(t) - \frac{h^2}{6} f'(0, t) - \frac{h^3}{24} f''(0, t) + \frac{\tau h}{2} \left( \frac{h^2}{12\tau} + \sigma - \frac{1}{2} \right) \ddot{u}(0, t) - \\
&- \frac{\tau}{2} \cdot \frac{h}{2} f'(0, t) + O(\tau^2 + h^4).
\end{aligned}$$

### Устойчивость по начальным данным

Рассмотрим разностную схему с однородным уравнением

$$\begin{cases} y_t = \Lambda(\sigma \hat{y} + (1 - \sigma)y), & (x, t) \in \omega_{h\tau}, \\ y(x, 0) = \cos x, & x \in \bar{\omega}_h, \\ y_x^{(\sigma)}(0, t) = y^{(\sigma)}(0, t) + \frac{h}{2} y_t(0, t) - \tilde{\mu}_0(t), & t \in \bar{\omega}_\tau, \\ y(1, t) = \cos(t + 1), & t \in \bar{\omega}_\tau. \end{cases} \quad (3)$$

Исследовать данную разностную схему на устойчивость будем методом разделения переменных. Учитывая граничные условия задачи, разложим решение соответствующей этой разностной задачи по собственным функциям  $\mu_k(x)$  разностного оператора  $\Lambda y = y_{\bar{x}x}$ , которые являются решениями разностной задачи Штурма-Лиувилля

$$\begin{cases} y_{\bar{x}x} - \lambda y = 0, \\ y_x(0) - y(0) = 0, \\ y(1) = 0. \end{cases}$$

или в индексной форме

$$\begin{cases} \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} - \lambda y_k = 0, & k = \overline{1, N-1}, \\ \frac{y_1 - y_0}{h} - y_0 = 0, \\ y_N = 0. \end{cases}$$

Решение этой разностной задачи можно найти с помощью вычислительных методов алгебры. Нас же интересует ограничение на собственные значения  $\lambda_k$ , которое мы можем получить, составив характеристическое уравнение для разностного уравнения этой задачи:

$$r^2 - (2 + \lambda h^2)r + 1 = 0.$$

Причем при наличии у характеристического уравнения вещественных корней построить нетривиальное решение разностной задачи Штурма-Лиувилля не удастся. Поэтому

необходимо, чтобы дискриминант этого квадратного уравнения был отрицательным, а следовательно

$$\frac{D}{4} = \left(1 - \frac{\lambda h^2}{2}\right) - 1 < 0,$$

откуда получаем

$$\lambda h^2 \left(\frac{\lambda h^2}{4} - 1\right) < 0.$$

Таким образом,

$$\lambda \in \left(0, \frac{4}{h^2}\right).$$

Тогда решение разностной задачи можно представить в виде

$$y(x, t) = \sum_{k=1}^{N-1} T_k(t) \mu_k(x). \quad (4)$$

Любая гармоника является решением задачи (13) при начальном условии

$$u_0(x) = T_k(0) \mu_k(x).$$

Подставим выражение (3) в разностное уравнение, тогда получим соотношение

$$\sum_{k=1}^{N-1} (T_k)_t \mu_k = \sum_{k=1}^{N-1} (\sigma \hat{T}_k + (1 - \sigma) T_k) \Lambda \mu_k,$$

при этом, поскольку  $\mu_k$  – это собственные функции разностного оператора  $\Lambda$ , то

$$\Lambda \mu_k = \lambda_k \mu_k.$$

Воспользуемся ортонормированностью системы функций  $\mu_k$ . Тогда из последнего равенства у нас останутся лишь слагаемые, относящиеся скалярному произведению при совпадающих индексах, то есть

$$\frac{\hat{T}_k - T_k}{\tau} = -\lambda_k (\sigma \hat{T}_k + (1 - \sigma) T_k), \quad k = \overline{1, N-1}.$$

Отсюда выразим  $\hat{T}_k$  на верхнем слое через  $T_k$  на нижнем слое:

$$\hat{T}_k = q_k T_k, \quad q_k = \frac{1 - (1 - \sigma) \lambda_k \tau}{1 + \sigma \lambda_k \tau}, \quad k = \overline{1, N-1}. \quad (5)$$

Как следует из формулы (5) устойчивость  $k$ -ой гармоники будет иметь место при выполнении неравенства

$$|q_k| \leq 1,$$

отсюда следует, что

$$\sigma \geq \frac{1}{2} - \frac{1}{\lambda_k \tau}.$$

Нам нужно также оценить то, как себя ведут  $\lambda_k$ . Для собственных значений  $\lambda_k$  справедлива оценка

$$\lambda_k \leq \lambda_{N-1} < \frac{4}{h^2}.$$

Отсюда следует, что

$$-\frac{1}{\lambda_k \tau} < -\frac{h^2}{4\tau}.$$

Тогда условие устойчивости по начальным данным в сеточной норме  $L_2$  будет выполняться, если

$$\sigma \geq \sigma_0 = \frac{1}{2} - \frac{h^2}{4\tau}. \quad (6)$$

## Устойчивость по правой части

Рассмотрим разностную схему с однородным начальным условием

$$\begin{cases} y_t = \Lambda(\sigma \hat{y} + (1 - \sigma)y) + \varphi, & (x, t) \in \omega_{h\tau}, \\ y(x, 0) = 0, & x \in \bar{\omega}_h, \\ y_x^{(\sigma)}(0, t) = y^{(\sigma)}(0, t) + \frac{h}{2}y_t(0, t) - \tilde{\mu}_0(t), & t \in \bar{\omega}_\tau, \\ y(1, t) = \cos(t + 1), & t \in \bar{\omega}_\tau. \end{cases} \quad (7)$$

Нам нужно исследовать устойчивость задачи (7). Вновь решение будем искать в виде разложения, но у нас также добавится сеточная функция в правой части основного уравнения. Поэтому мы представим эту сеточную функцию в виде разложения по системе базисных функций  $\mu_k$

$$\varphi(x, t) = \sum_{k=1}^{N-1} \varphi_k(t) \mu_k(x).$$

Проделявая те же операции, как и в предыдущем пункте, мы можем получить соотношение между функциями  $T_k$  на верхнем слое и на нижнем слое:

$$\hat{T}_k = q_k T_k + \frac{\tau \varphi_k}{1 + \sigma \lambda_k \tau}. \quad (8)$$

Таким образом, суммируя по гармоникам, мы можем записать решение задачи (7) на верхнем слое

$$\hat{y} = \sum_{k=1}^{N-1} \hat{T}_k \mu_k = \sum_{k=1}^{N-1} q_k T_k \mu_k + \tau \sum_{k=1}^{N-1} \frac{\varphi_k}{1 + \sigma \lambda_k \tau} \mu_k.$$

Оценим это решение по норме

$$\begin{aligned} \|\hat{y}\| &\leq \left\| \sum_{k=1}^{N-1} q_k T_k \mu_k \right\| + \tau \left\| \sum_{k=1}^{N-1} \frac{\varphi_k}{1 + \sigma \lambda_k \tau} \mu_k \right\| = \left[ \|\cdot\| = \sqrt{(\cdot, \cdot)} \right] \leq \max_{1 \leq k \leq N-1} |q_k| \left( \sum_{k=1}^{N-1} T_k^2 \right)^{1/2} + \\ &+ \max_{1 \leq k \leq N-1} \frac{\tau}{|1 + \sigma \lambda_k \tau|} \left( \sum_{k=1}^{N-1} \phi_k^2 \right)^{1/2} = \max_{1 \leq k \leq N-1} |q_k| \cdot \|y\| + \max_{1 \leq k \leq N-1} \frac{\tau}{|1 + \sigma \lambda_k \tau|} \|\phi\|. \end{aligned}$$

Если одновременно будут выполняться условия  $\sigma \geq \sigma_0$ ,  $\sigma \geq 0$ , то будет выполняться  $1 + \sigma \lambda_k \tau \geq 1$ ,  $\forall k$  и  $|q_k| \leq 1$ . А тогда

$$\|\hat{y}\| \leq \|y\| + \tau \|\phi\|.$$

Если мы будем суммировать это неравенство по всем временным слоям, то мы получим оценку всего решения на любом слое

$$\|y^{j+1}\| \leq \|y^0\| + \tau \sum_{j'=0}^j \|\phi^{j'}\|.$$

Отсюда следует окончательная оценка

$$\|y^{j+1}\| \leq \|\phi\|_{\omega_{h\tau}}. \quad (9)$$

Условие (9) доказывает устойчивость разностной схемы по правой части при одновременном выполнении условий  $\sigma \geq \sigma_0$ ,  $\sigma \geq 0$ .

## Вариации параметра $\sigma$

Из аппроксимации разностного уравнения следует

1. Пусть  $\sigma$  – это любое число. Если мы в качестве  $\varphi$  выберем

$$\varphi = f + O(\tau + h^2),$$

то тогда погрешность аппроксимации будет равна

$$\Psi = O(\tau + h^2), \quad u(x, t) \in C_2^4(\overline{D}).$$

2. Пусть  $\sigma = \sigma_\alpha = \frac{1}{2} + \alpha \frac{h^2}{\tau}$ , где  $\alpha$  – любое число. Если в этом случае мы возьмем

$$\varphi = f + \frac{\tau}{2} \dot{f} + O(\tau^2 + h^2) = f\left(x, t + \frac{\tau}{2}\right) + O(\tau^2 + h^2).$$

Тогда мы получим значение погрешности равное

$$\Psi = O(\tau^2 + h^2).$$

Таким образом, мы получим семейство схем второго порядка. В частности при  $\alpha = 0$  мы получим схему Кранка-Николсона.

3. Пусть  $\sigma = \sigma_* = \frac{1}{2} - \frac{h^2}{12\tau}$ , то есть мы выбрали  $\alpha = -\frac{1}{12}$ . Если мы возьмем

$$\varphi = f + \frac{\tau}{2} \dot{f} + \frac{h^2}{12} f'' + O(\tau^2 + h^4),$$

то получим порядок аппроксимации

$$\Psi(x, t) = O(\tau^2 + h^4).$$

Из аппроксимации левого краевого условия

1. При любых  $\sigma$  и

$$\tilde{\mu}_0(t) = \mu_0(t) + \frac{h}{2} f(0, t),$$

получим

$$\nu(0, t) = O(\tau + h^2).$$

2. При любых  $\sigma$  и

$$\tilde{\mu}_0(t) = \mu_0(t) + \frac{h}{2} f(0, t) + \sigma \tau \dot{\mu}_0(t),$$

получим

$$\nu(0, t) = O(\tau^2 + h^2).$$

3. При  $\sigma = \sigma_* = \frac{1}{2} - \frac{h^2}{12\tau}$  и

$$\tilde{\mu}_0(t) = \mu_0(t) + \frac{h}{2} f(0, t) + \left(\frac{\tau}{2} + \frac{h^2}{12}\right) \dot{\mu}_0(t) - \frac{h^2}{6} \beta_0 y_t(0, t) + \frac{h^2}{2} f'(0, t) + \frac{h^3}{24} f''(0, t) + \frac{\tau h}{4} \dot{f}(0, t)$$

получим

$$\nu(0, t) = O(\tau^2 + h^4).$$

Из условий устойчивости по начальным данным и правой части

1. Если  $\sigma = 0$ , то схема будет явная и условие устойчивости имеет вид

$$\tau \leq \frac{h^2}{2}.$$

2. Если  $\sigma \geq \frac{1}{2}$ , то схема устойчива при любых шагах  $\tau$  и  $h$ . В частности при  $\sigma = 1$  мы получим чисто неявную схему, которая является абсолютно устойчивой для любых шагов  $\tau, h$ .
3. Если  $0 < \sigma < \frac{1}{2}$ , то получим оценку устойчивости

$$\tau \leq \frac{h^2}{2(1 - \tau\sigma)}.$$

4. Если  $\sigma = \sigma_\alpha = \frac{1}{2} + \alpha \frac{h^2}{\tau}$ , то для выполнения условия устойчивости нужно взять  $\alpha \geq -\frac{1}{4}$ . При этом в данном случае схема будет устойчива при любых шагах  $\tau, h$ .
5. Если  $\sigma = \sigma_* = \frac{1}{2} - \frac{h^2}{12\tau}$ , то схема будет устойчива при любых шагах  $\tau, h$ .

Таким образом, в итоге

1. при  $\sigma = 0$ , выбирая

$$\varphi = f + O(\tau + h^2), \quad \tilde{\mu}_0(t) = \mu_0(t) + \frac{h}{2}f(0, t),$$

получим разностную схему порядка аппроксимации

$$\Psi(x, t) = O(\tau + h^2)$$

и устойчивую при

$$\tau \leq \frac{h^2}{2};$$

2. при  $\sigma = 1$ , выбирая

$$\varphi = f + O(\tau + h^2), \quad \tilde{\mu}_0(t) = \mu_0(t) + \frac{h}{2}f(0, t),$$

получим разностную схему порядка аппроксимации

$$\Psi(x, t) = O(\tau + h^2)$$

и устойчивую при любых шагах  $\tau, h$ ;

3. при  $\sigma = \sigma_\alpha = \frac{1}{2} + \alpha \frac{h^2}{\tau}$ , где  $\alpha$  – любое число, выбирая

$$\varphi = f + \frac{\tau}{2}\dot{f} + O(\tau^2 + h^2) = f\left(x, t + \frac{\tau}{2}\right) + O(\tau^2 + h^2), \quad \tilde{\mu}_0(t) = \mu_0(t) + \frac{h}{2}f(0, t) + \sigma\tau\dot{\mu}_0(t),$$

получим разностную схему порядка аппроксимации

$$\Psi(x, t) = O(\tau^2 + h^2)$$

и устойчивую при любых шагах  $\tau, h$ , если  $\alpha \geq -\frac{1}{4}$ ;



4. при  $\sigma = \sigma_* = \frac{1}{2} - \frac{h^2}{12\tau}$ , выбирая

$$\varphi = f + \frac{\tau}{2}\dot{f} + \frac{h^2}{12}f'' + O(\tau^2 + h^4),$$

$$\tilde{\mu}_0(t) = \mu_0(t) + \frac{h}{2}f(0, t) + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\dot{\mu}_0(t) - \frac{h^2}{6}\beta_0 y_t(0, t) + \frac{h^2}{2}f'(0, t) + \frac{h^3}{24}f''(0, t) + \frac{\tau h}{4}\dot{f}(0, t),$$

получим разностную схему порядка аппроксимации

$$\Psi(x, t) = O(\tau^2 + h^4)$$

и устойчивую при любых шагах  $\tau, h$ ;

где

$$f(x, t) = \sqrt{2} \sin\left(\frac{\pi}{4} - x - t\right), \quad \mu_0(t) = \sqrt{2} \sin\left(\frac{\pi}{4} + t\right), \quad \beta_0 = 1.$$

## Реализация разностной схемы

Алгоритм реализации будет следующий:

1. нулевой слой заполняем по формуле

$$y_k^0 = \cos(x_k), \quad k = \overline{0, N}; \quad (10)$$

2. для всех  $j = \overline{0, N_1 - 1}$  будем вычислять  $y_k^j$ , решая систему линейных алгебраических уравнений с трехдиагональной матрицей, а именно

$$\begin{cases} \frac{\sigma}{h^2}y_{k-1}^{j+1} - \left(\frac{2\sigma}{h^2} + \frac{1}{\tau}\right)y_k^{j+1} + \frac{\sigma}{h^2}y_{k+1}^{j+1} = -F_k^j, & k = \overline{1, N-1}, \\ \frac{\sigma}{h}y_1^{j+1} - \left(\sigma + \frac{\sigma}{h} + \frac{h}{2\tau}\right)y_0^{j+1} = -\frac{(1-\sigma)}{h}y_1^j + \left((1-\sigma)\left(\frac{1}{h} + 1\right) - \frac{h}{2\tau}\right)y_0^j - \tilde{\mu}_0(t_{j+1}), \\ y_N^{j+1} = \cos(t_{j+1} + 1), \end{cases} \quad (11)$$

где

$$F_k^j = \frac{1}{\tau}y_k^j + (1-\sigma)\frac{y_{k-1}^j - 2y_k^j + y_{k+1}^j}{h^2} + \varphi_k^j.$$

Очевидно, что наиболее эффективным методом решения системы (11) будет метод прогонки, а сам он будет устойчивый, что следует из вида коэффициентов.

```
[174]: import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

По аналогии с предыдущими лабораторными определим функцию для создания сеток узлов

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

Функция для метода прогонки

```
[264]: def tridiagonal_algorithm(a,b,c,f):
    a, b, c, f = tuple(map(lambda k_list: list(map(float, k_list)), (a, b,
↪c, f)))

    alpha = [-b[0] / c[0]]
    beta = [f[0] / c[0]]
    n = len(f)
    x = [0]*n

    for i in range(1, n):
        alpha.append(-b[i]/(a[i]*alpha[i-1] + c[i]))
        beta.append((f[i] - a[i]*beta[i-1])/(a[i]*alpha[i-1] + c[i]))

    x[n-1] = beta[n - 1]

    for i in range(n - 1, 0, -1):
        x[i - 1] = alpha[i - 1]*x[i] + beta[i - 1]

    return x
```

Функция, определяющая точное решение

```
[265]: def u(x,t):
    return np.cos(x+t)
```

Функции, определяющие входные данные нашей дифференциальной задачи

```
[266]: def u_0(x):
    return np.cos(x)

def f(x,t):
    return np.sqrt(2) * np.sin(np.pi/4 - x - t)

def mu_0(t):
    return np.sqrt(2) * np.sin(np.pi/4 + t)

def mu_1(t):
    return np.cos(t + 1)
```

Функция, определяющая реализацию разностной схемы

```
[267]: def diff_scheme_solve(x, t, h, tau, sigma, u_0, phi, mu_01, mu_1):

    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)-1):
        gamma = [-(sigma+sigma/h+h/(2*tau))]
        beta = [sigma/h]
```

```

        g = [-(1-sigma)/h*y[1,j]+((1-sigma)*(1/h + 1)-h/(2*tau))*y[0,j] -
mu_01(t[j+1])]
        alpha = [0]
        for k in range(1, len(x)-1):
            alpha.append(sigma/h**2)
            gamma.append(-(2*sigma / h**2+1/tau))
            beta.append(sigma/h**2)
            g.append(-(1/tau*y[k,j] + (1-sigma)*(y[k-1, j] - 2*y[k,j] +
y[k+1,j])/h**2 + phi(x[k], t[j])))
        alpha.append(0)
        gamma.append(1)
        g.append(mu_1(t[j+1]))
        beta.append(0)
        y[:, j+1] = tridiagonal_algorithm(alpha,beta,gamma,g)

    return y

```

Создание сетки узлов:

```
[268]: h, x_grid = generate_grid(1, 10)
tau, t_grid = generate_grid(0.1, 10)
```

```
[269]: print('tau=' + str(tau) + '\nh=' + str(h))
```

```
tau=0.01
```

```
h=0.1
```

Проверка неравенства Куранта

```
[270]: tau < h**2 / 2
```

```
[270]: False
```

Определяем вариационные параметры, влияющие на порядок аппроксимации

```
[271]: sigma = 0
phi = f
def mu_01(t):
    return mu_0(t) + h/2 * f(0,t)
```

Находим приближенное решение

```
[272]: y = diff_scheme_solve(x_grid, t_grid, h, tau, sigma, u_0, phi, mu_01, mu_1)
```

Имеем  $\sigma = 0$ ,  $\tau > h^2/2$ , то есть разностная схема должна оказаться неустойчивой

```
[273]: def print_exact():
    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)')

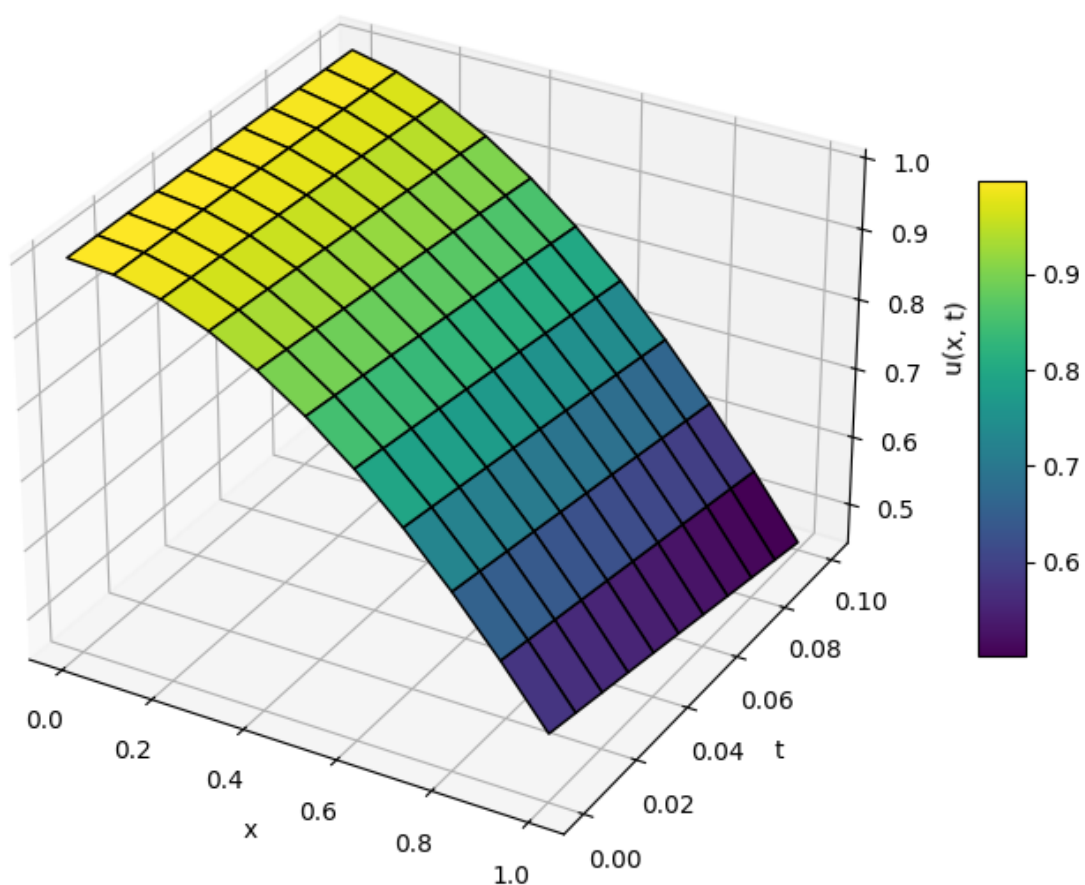
fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10)

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

print_exact()

```

График точного решения



```

[274]: def print_approx():
        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)')

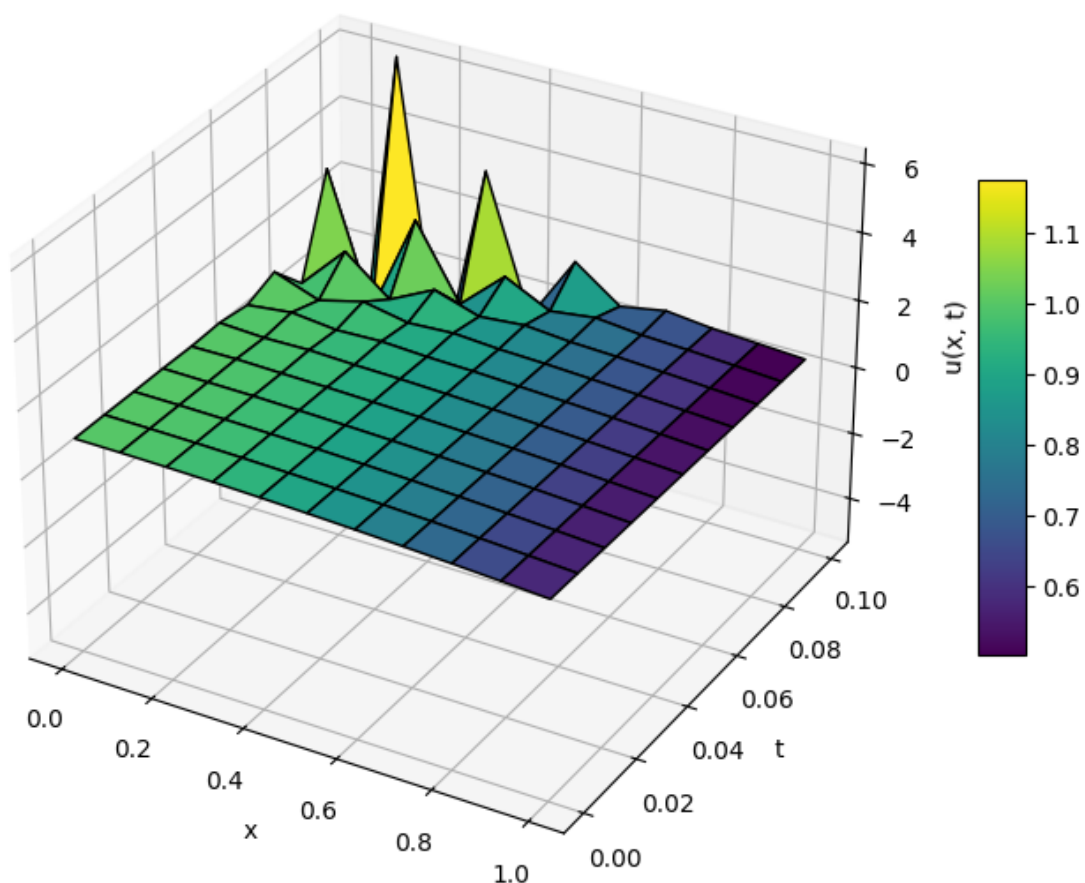
fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10)

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

print_approx()

```

График приближенного решения



Действительно, решение оказалось неустойчиво.

Построим сетку узлов по-другому

```
[275]: h, x_grid = generate_grid(1, 10)
       tau, t_grid = generate_grid(0.1, 100)
```

```
[276]: print('tau=' + str(tau) + '\nh=' + str(h))
```

tau=0.001

h=0.1

Проверка неравенства Куранта

```
[277]: tau < h**2 / 2
```

[277]: True

Находим приближенное решение

```
[278]: y = diff_scheme_solve(x_grid, t_grid, h, tau, sigma, u_0, phi, mu_01, mu_1)
```

Имеем  $\sigma = 0$ ,  $\tau < h^2/2$ , то есть разностная схема должна оказаться устойчивой

```
[279]: print_exact()
       print_approx()
```

График точного решения

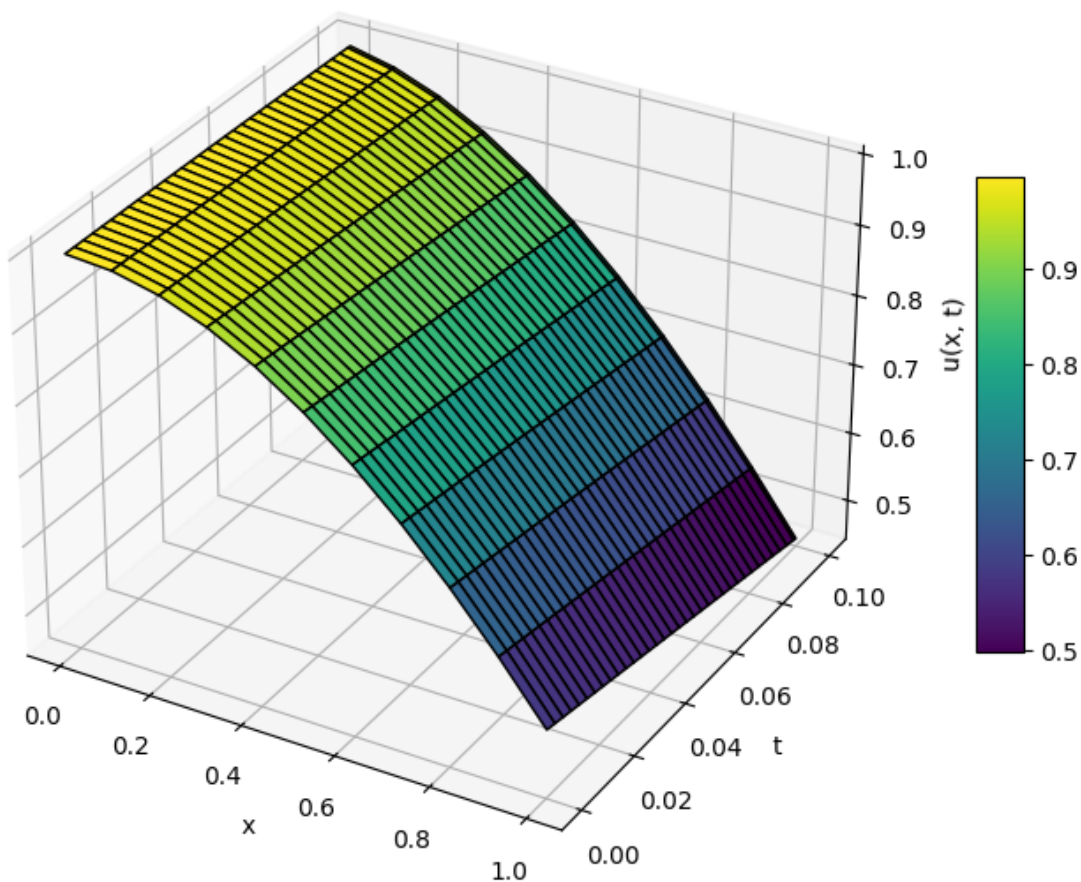
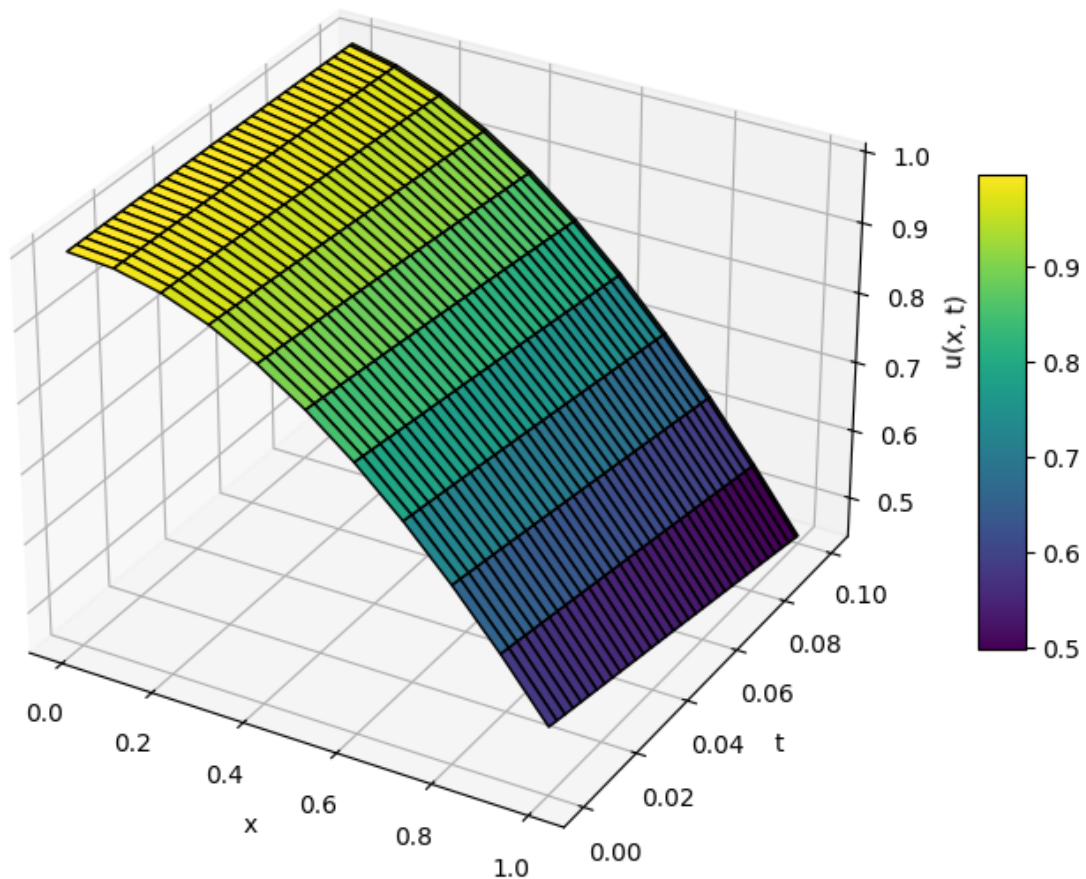


График приближенного решения



Аналогично, изменяя вариационные параметры, можем построить приближенные решения с более высокой погрешностью аппроксимации.

## Постановка задачи 2

Поставлена третья краевая задача для уравнения теплопроводности с непрерывными коэффициентами

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( (\cos(x+t) + 1) \frac{\partial u}{\partial x} \right) + 2 \cos(x+t) + \sqrt{2} \cos\left(\frac{\pi}{4} + x+t\right), & t > 0, 0 < x < 1, \\ u(x, 0) = \cos x, & 0 \leq x \leq 1, \\ u(0, t) = \cos t, & t \geq 0 \\ -\frac{\partial u(1, t)}{\partial x} = \sin(t+1), & t \geq 0. \end{cases} \quad (12)$$

## Построение разностной схемы

Сперва проверим корректную постановку данной дифференциальной задачи, для чего рассмотрим условия согласования:

$$\cos x|_{x=0} = 1 = \cos t|_{t=0},$$

$$-\frac{\partial \cos x}{\partial x}\Big|_{x=1} = \sin 1 = \sin(t+1)|_{t=0}.$$

Таким образом, условия согласования выполнены. Также для данной задачи выполняется принцип максимума, который позволяет заключить, что если решение задачи существует, то оно единственно. Следовательно, дифференциальная задача является корректно поставленной.

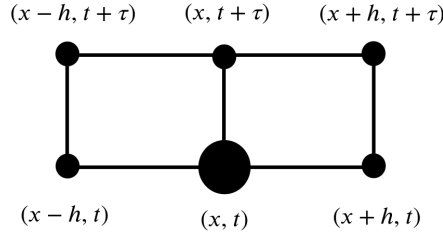
Поставленная дифференциальная задача имеет точное решение

$$u(x, t) = \cos(x + t).$$

Зададим равномерную по каждому направлению сетку

$$\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau, \quad \bar{\omega}_h = \{x_k = kh, \quad k = \overline{0, N}, \quad Nh = 1\}, \quad \bar{\omega}_\tau = \{t_j = j\tau, \quad j = \overline{0, N_1}, \quad N_1\tau = T\}.$$

На этой сетке зададим шеститочечный шаблон



В соответствии с методом баланса, используя данный шаблон и заменяя дифференциальные операторы разностными, можем записать разностную схему для поставленной задачи в бензиндексной форме

$$\begin{cases} y_t = \Lambda(\sigma \hat{y} + (1 - \sigma)y) + \varphi, & (x, t) \in \omega_{h\tau}, \\ y(x, 0) = \cos x, & x \in \bar{\omega}_h, \\ y(0, t) = \cos t, & t \in \bar{\omega}_\tau, \\ -y_x^{(\sigma)}(1, t) = \sin(t + 1), & t \in \bar{\omega}_\tau. \end{cases} \quad (13)$$

Здесь

$$\Lambda y = (\bar{a}y_{\bar{x}})_x, \quad \hat{y} = y(x, t + \tau),$$

$\sigma$  – это параметр, а из метода баланса

$$\bar{a}_i = \left[ \frac{1}{h} \int_{x_{i-1}}^{x_i} \frac{dx}{k\left(x, t_j + \frac{\tau}{2}\right)} \right]^{-1}, \quad \text{где } k(x, t) = \cos(x + t) + 1,$$

$\bar{a}$ ,  $\varphi$  – это некоторые сеточные функции. В частности, можно выбирать

$$a_i^j = k\left(x_i - \frac{h}{2}, t_j + \frac{\tau}{2}\right), \quad \text{или } a_i^j = \frac{1}{2} \left[ k\left(x_i - h, t_j + \frac{\tau}{2}\right) + k\left(x_i, t_j + \frac{\tau}{2}\right) \right],$$

$$\varphi_i^j = f_i^{j+\frac{1}{2}}, \quad \text{или } \varphi_i^j = \frac{1}{2}(f_i^j + f_i^{j+1}).$$



## Реализация разностной схемы

Пусть

$$a_i^j = k \left( x_i - \frac{h}{2}, t_j + \frac{\tau}{2} \right) = k_{i-\frac{1}{2}}^{j+\frac{1}{2}}, \quad \varphi_i^j = f_i^{j+\frac{1}{2}}$$

Алгоритм реализации будет следующий:

1. нулевой слой заполняем по формуле

$$y_i^0 = \cos(x_i), \quad i = \overline{0, N}; \quad (14)$$

2. для всех  $j = \overline{0, N_1 - 1}$  будем вычислять  $y_k^j$ , решая систему линейных алгебраических уравнений с трехдиагональной матрицей, а именно

$$\begin{cases} \frac{\sigma}{h^2} k_{i-\frac{1}{2}}^{j+\frac{1}{2}} y_{i-1}^{j+1} - \left( \frac{\sigma}{h^2} (k_{i-\frac{1}{2}}^{j+\frac{1}{2}} + k_{i+\frac{1}{2}}^{j+\frac{1}{2}}) + \frac{1}{\tau} \right) y_i^{j+1} + \frac{\sigma}{h^2} k_{i+\frac{1}{2}}^{j+\frac{1}{2}} y_{i+1}^{j+1} = -F_i^j, \quad i = \overline{1, N-1}, \\ y_0^{j+1} = \cos(t_{j+1}), \\ -\frac{\sigma}{h} y_N^{j+1} + \frac{\sigma}{h} y_{N-1}^{j+1} = \frac{(1-\sigma)}{h} (y_N^j - y_{N-1}^j) + \sin(t_{j+1} + 1), \end{cases} \quad (15)$$

где

$$F_k^j = \frac{1}{\tau} y_i^j + \frac{(1-\sigma)}{h^2} (k_{i-\frac{1}{2}}^{j+\frac{1}{2}} y_{i-1}^j - k_{i-\frac{1}{2}}^{j+\frac{1}{2}} y_i^j - k_{i+\frac{1}{2}}^{j+\frac{1}{2}} y_i^j + k_{i+\frac{1}{2}}^{j+\frac{1}{2}} y_{i+1}^j) + f_i^{j+\frac{1}{2}}.$$

Очевидно, что наиболее эффективным методом решения системы (15) будет метод прогонки, а сам он будет устойчивый, что следует из вида коэффициентов.

```
[280]: def u_0(x):  
        return np.cos(x)  
  
def f(x,t):  
    return 2*np.cos(x+t) + np.sqrt(2) * np.cos(np.pi/4 + x + t)  
  
def mu_0(t):  
    return np.cos(t)  
  
def mu_1(t):  
    return np.sin(t + 1)  
  
def k(x,t):  
    return np.cos(x+t) + 1
```

```
[281]: def diff_scheme_solve(x, t, h, tau, sigma, f, u_0, mu_0, mu_1, k):  
  
    y = np.zeros((len(x), len(t)))  
  
    for i in range(len(x)):  
        y[i, 0] = u_0(x[i])  
  
    for j in range(len(t)-1):  
        gamma = [1]
```

```

    beta = [0]
    g = [mu_0(t[j+1])]
    alpha = [0]
    for i in range(1, len(x)-1):
        alpha.append(sigma/h**2 * k(x[i] - h/2, t[j] + tau/2))
        gamma.append(-(sigma / h**2*(k(x[i] - h/2, t[j] + tau/2) +
↪k(x[i] + h/2, t[j] + tau/2))+1/tau))
        beta.append(sigma/h**2*k(x[i] + h/2, t[j] + tau/2))
        g.append(-(1/tau*y[i,j] + (1-sigma)*(k(x[i] - h/2, t[j] + tau/
↪2)*y[i-1, j] - k(x[i] - h/2, t[j] + tau/2)*y[i,j] - k(x[i] + h/2, t[j] +
↪tau/2)*y[i,j] + k(x[i] + h/2, t[j] + tau/2)*y[i+1,j])/h**2 + f(x[i],
↪t[j]+tau/2)))
        alpha.append(sigma/h)
        gamma.append(-sigma/h)
        g.append((1-sigma)/h*(y[len(x)-1, j] - y[len(x)-2,j]) + mu_1(t[j+1]))
        beta.append(0)
    y[:, j+1] = tridiagonal_algorithm(alpha,beta,gamma,g)

    return y

```

```

[290]: h, x_grid = generate_grid(1, 10)
       tau, t_grid = generate_grid(0.1, 10)

```

```

[291]: print('tau=' + str(tau) + '\nh=' + str(h))

```

```

tau=0.01
h=0.1

```

```

[298]: sigma = 1

```

Находим приближенное решение

```

[299]: y = diff_scheme_solve(x_grid, t_grid, h, tau, sigma, f, u_0, mu_0, mu_1, k)

```

```

[300]: print_exact()
       print_approx()

```

График точного решения

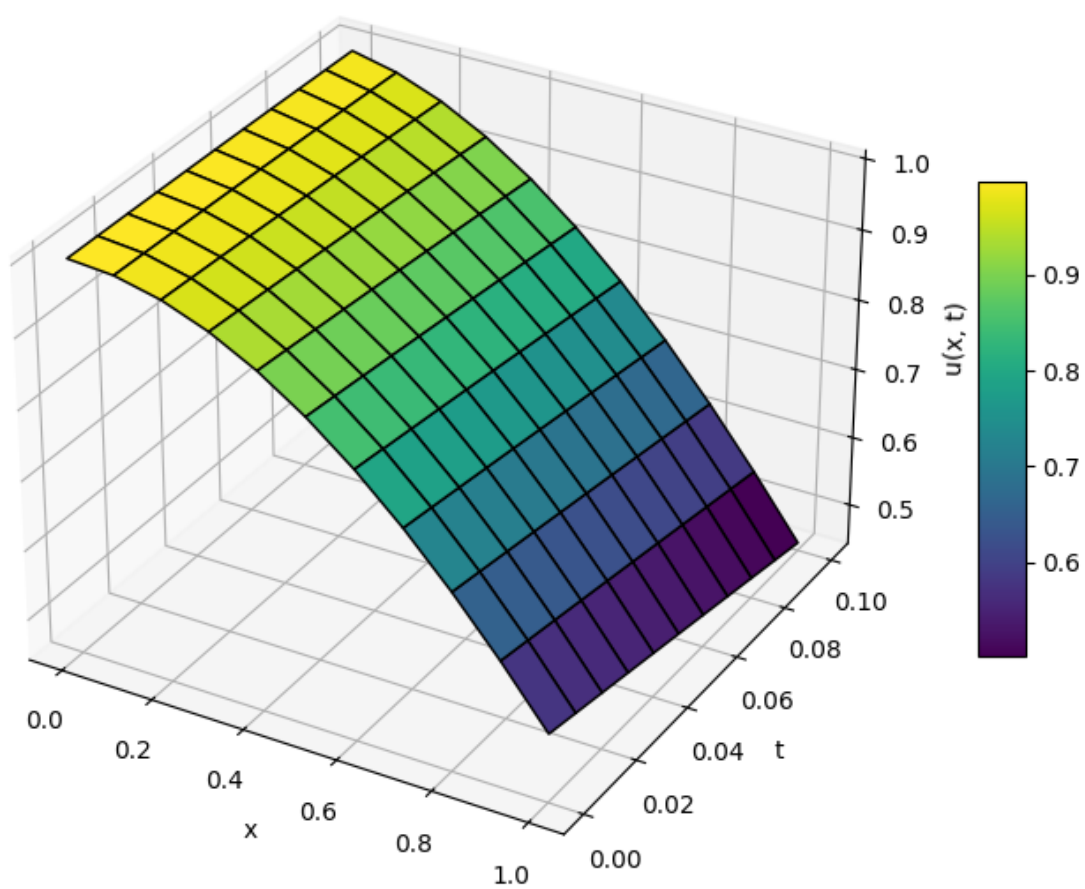


График приближенного решения

