

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей  
Кафедра информатики  
Дисциплина: Методы численного анализа

**ОТЧЁТ**  
к лабораторной работе  
на тему

Метод сеток решения волнового уравнения

Выполнил: студент группы 153501  
Тимофеев Кирилл Андреевич

Проверил: Анисимов Владимир Яковлевич

Минск 2023

## Цели выполнения задания

- изучить метод разностных аппроксимаций для волнового уравнения, составить алгоритмы решения волнового уравнения методом сеток, применимым для организации вычислений на ПЭВМ;
- составить программы волнового уравнения по разработанным алгоритмам;
- выполнить тестовые примеры и проверить правильность работы программ.
- получить численное решение волнового уравнения.

## Краткие теоретические сведения

### Волновое уравнение

Рассмотрим смешанную задачу для волнового уравнения:

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = f(x, t), \quad 0 < x < 1, \quad 0 < t \leq T, \quad (2.63)$$

$$u(x, 0) = \rho(x), \quad u'_t(x, 0) = q(x), \quad 0 \leq x \leq 1, \quad (2.64)$$

$$u(0, t) = 0, \quad u(1, t) = 0, \quad 0 \leq t \leq T, \quad (2.65)$$

где  $f(x, t)$ ,  $\rho(x)$ ,  $q(x)$  – заданные достаточно гладкие функции, причем  $\rho(0) = \rho(1) = q(0) = q(1) = 0$ .

Будем предполагать, что задача (2.63) – (2.65) имеет единственное решение  $u(x, t) \in C_4(\bar{D})$ ,  $\bar{D} = \{(x, t): 0 \leq x \leq 1, \quad 0 \leq t \leq T\}$  – замкнутый прямоугольник.

### Разностная схема.

Будем использовать сетки, построенные на замкнутом прямоугольнике  $\bar{D}$  в лабораторной работе №14, и соответствующие обозначения сеточных **таблица 2.1.**

функций. Заменяем в уравнении (2.63) частную производную  $u''_{tt}$  приближенно второй разностной производной в направлении  $t$ , а частную производную —  $u''_{xx}$  второй разностной производной в направлении  $x$  и, заменив  $u$  на  $y$ , приходим к разностному уравнению

$$\frac{y_k^{v+1} - 2y_k^v + y_k^{v-1}}{\tau^2} + \Lambda y_k^v = f_k^v, \quad (2.66)$$

$k = 1, 2, \dots, N-1, \nu = 1, 2, \dots, M-1$ . Шаблон разностного уравнения (2.66) показан на рис. 2.9.

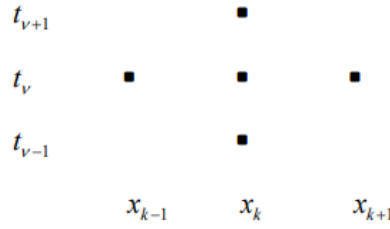


Рис. 2.9

Это уравнение можно разрешить явно относительно  $y_k^{v+1}$ . Но для того, чтобы находить значения разностного решения на  $(\nu+1)$ -м слое, требуется иметь уже вычисленные значения искомого решения на двух предыдущих слоях. Поэтому нужно получить разностное решение сначала отдельно на слоях, отвечающих значениям  $\nu = 0$  и  $\nu = 1$ . В этом нам помогут начальные условия.

Прежде всего, используя первое начальное условие (2.64), задаем

$$y_k^0 = \rho_k, \quad k = 1, 2, \dots, N-1. \quad (2.67)$$

Кроме того, полагаем при  $k = 1, 2, \dots, N-1$

$$y_k^1 = \rho_k + \tau q_k + \frac{\tau^2}{2} (f_k^0 - \Lambda \rho_k). \quad (2.68)$$

Правая часть формулы (2.68) аппроксимирует многочлен Тейлора

$$u(x_k, 0) + \tau u'_t(x_k, 0) + \frac{\tau^2}{2} u''_{tt}(x_k, 0), \text{ поскольку согласно (2.64) } u(x_k, 0) = \rho_k, \quad u'_t(x_k, 0) = q_k,$$

а из уравнения (2.63) для частных производных решения задачи (2.63) –

(2.65) вытекает связь  $u''_{tt}(x_k, 0) = f(x_k, 0) + u''_{xx}(x_k, 0)$ . Наконец, согласно краевым условиям (2.65) имеем

$$y_0^v = 0, \quad y_N^v = 0, \quad \nu = 0, 1, \dots, M. \quad (2.69)$$

Теперь разностная схема (2.63) – (2.65) полностью определена. Эта схема явная трехслойная (см. шаблон на рис. 2.7), условно устойчивая в некоторых естественных нормах.

Если  $h \rightarrow 0, \tau \rightarrow 0$ , причем  $\tau/h \leq c < 1, c = \text{const}$ , то решение  $y$  разностной схемы (2.66) сходится к рассматриваемому решению  $u$  задачи (2.63) – (2.65) в следующем смысле:

$$\|u - y\|_h = O(h^2 + \tau^2). \quad (2.70)$$

$$\text{где } \|u - y\|_h = \max_{0 \leq \nu \leq M} \left( h \sum_{k=1}^{N-1} (u_k^\nu - y_k^\nu)^2 \right)^{1/2}.$$

Схема (2.66) имеет второй порядок точности и по  $h$ , и по  $\tau$ .

### **Понятие о методе прямых.**

Если в задаче (2.63) – (2.65) ввести дискретность только по  $x$ , то мы придем к системе линейных обыкновенных дифференциальных уравнений

$$\frac{d^2 y_k}{dt^2} - \frac{y_{k-1} - 2y_k + y_{k+1}}{h^2} = f(x_k, t), \quad (2.71)$$

где  $k = 1, 2, \dots, N-1$

с начальными условиями

$$y_k(0) = \rho_k, \quad y'_k(0) = q_k, \quad (2.72)$$

причем  $y_0(t) \equiv y_N(t) \equiv 0$ .

При сделанном предположении относительно гладкости решения задачи (2.63) – (2.65) имеем

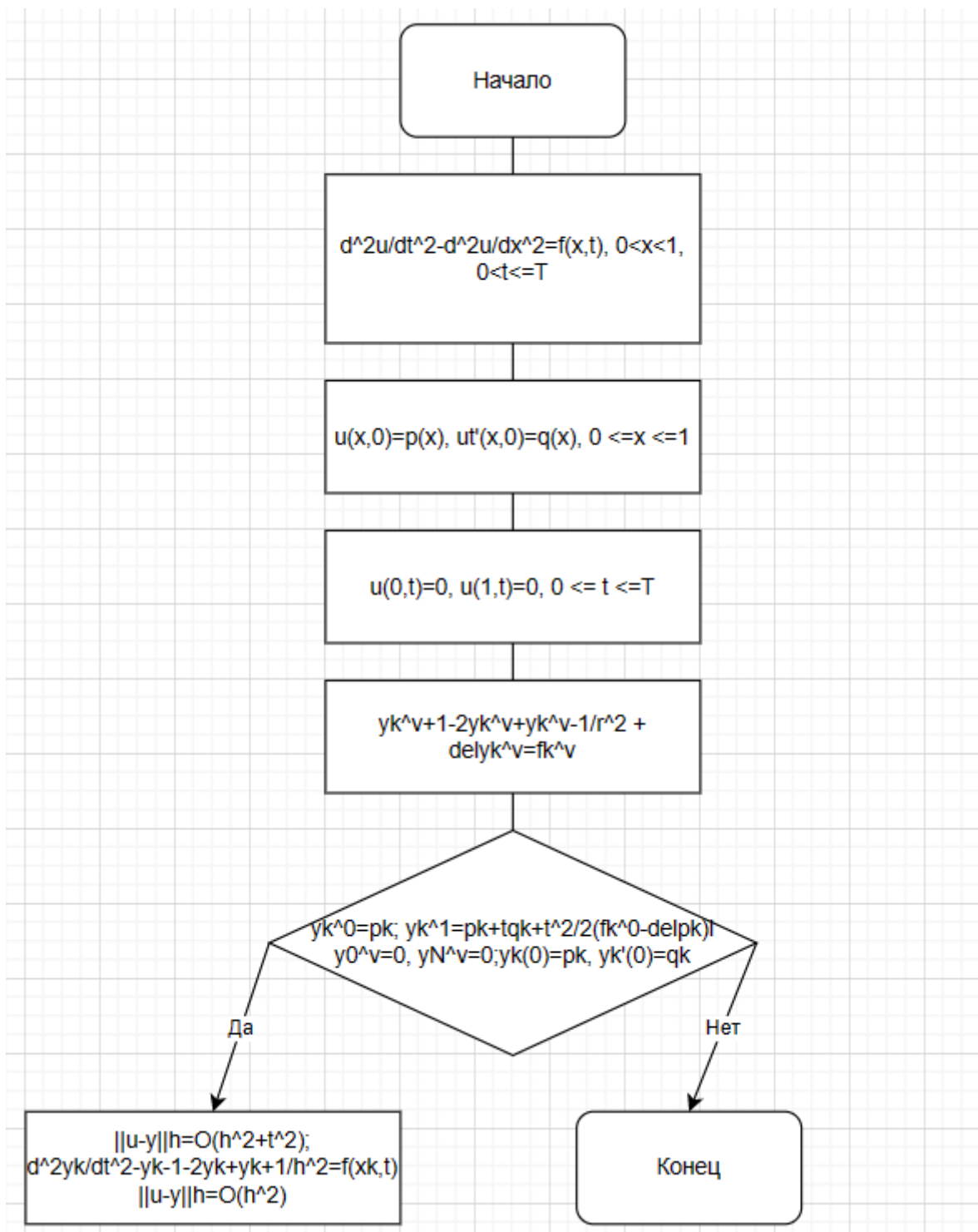
$$\|u - y\|_h = O(h^2), \quad (2.73)$$

$$\text{где } \|u - y\|_h = \max_{0 \leq t \leq T} \left( h \sum_{k=1}^{N-1} (u(x_k, t) - y_k(t))^2 \right)^{1/2}, \quad y_1(t), \quad y_2(t), \quad \dots, \quad y_{N-1}(t) \text{ — решение}$$

задачи Коши (2.63) – (2.65). Данный метод называется *методом прямых*,

### **таблица 2.1.**

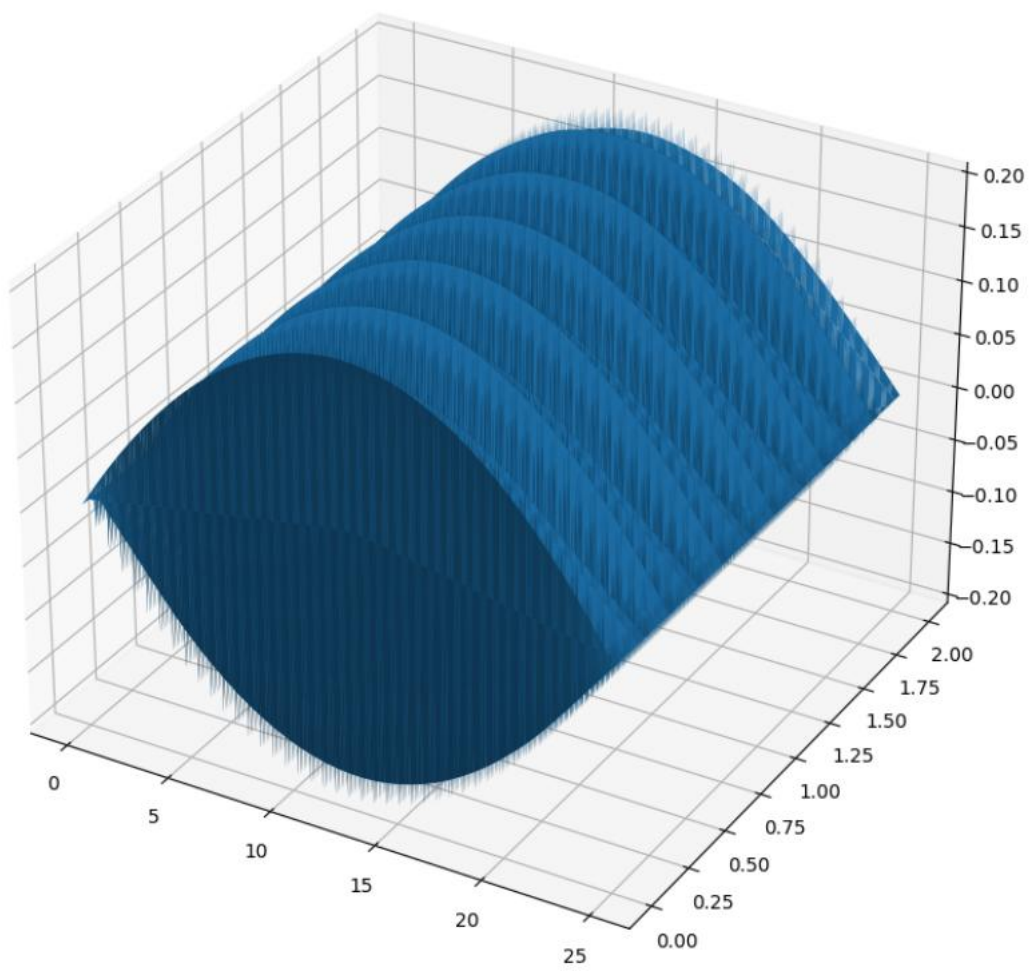
поскольку приближенное решение задачи (2.63) – (2.65) ищется на прямых  $x = x_k, k = 1, 2, \dots, N-1$ , расположенных в плоскости  $xt$ . Разностный же метод часто называется *методом сеток*.

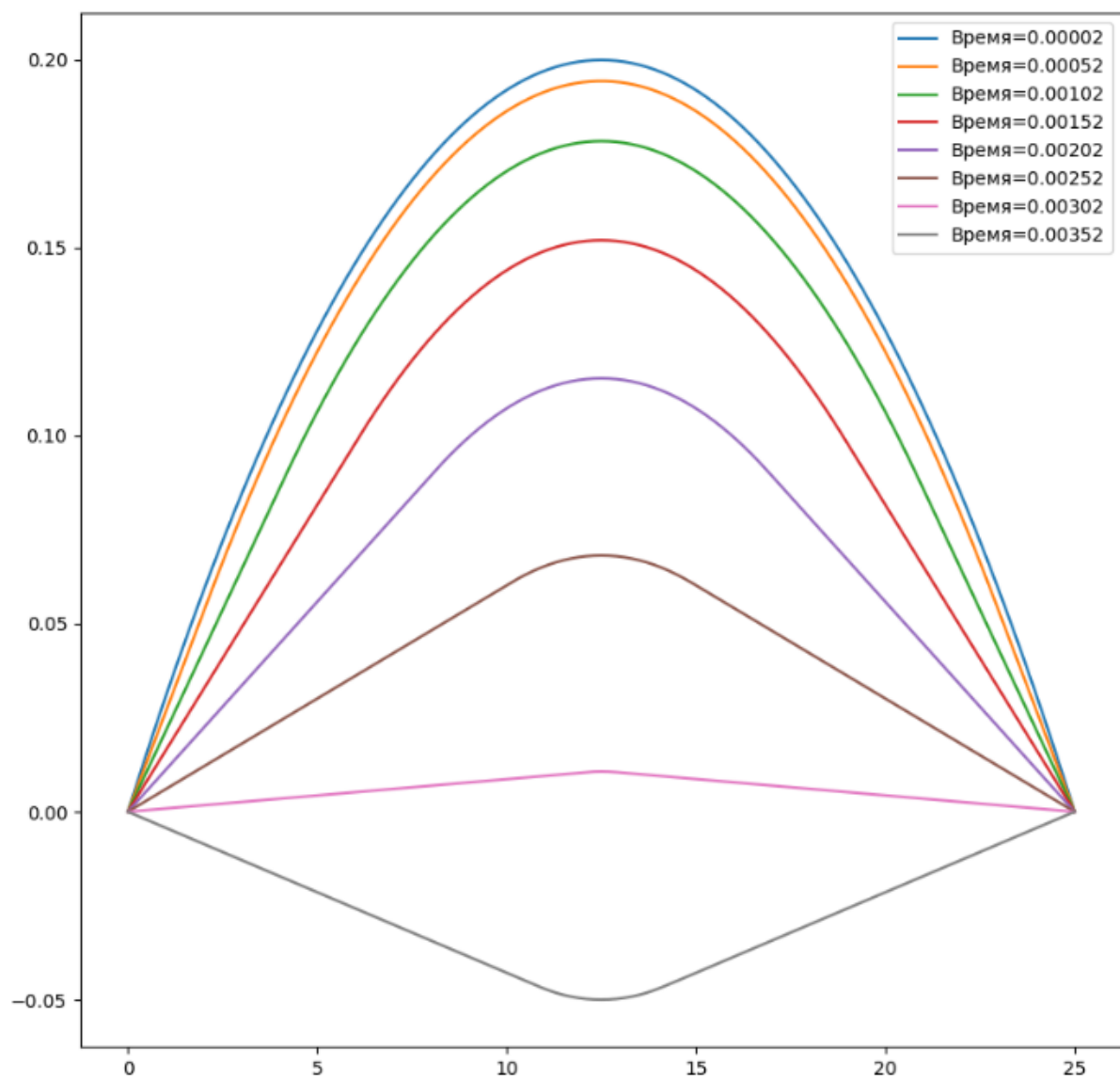


## Тестовые задания

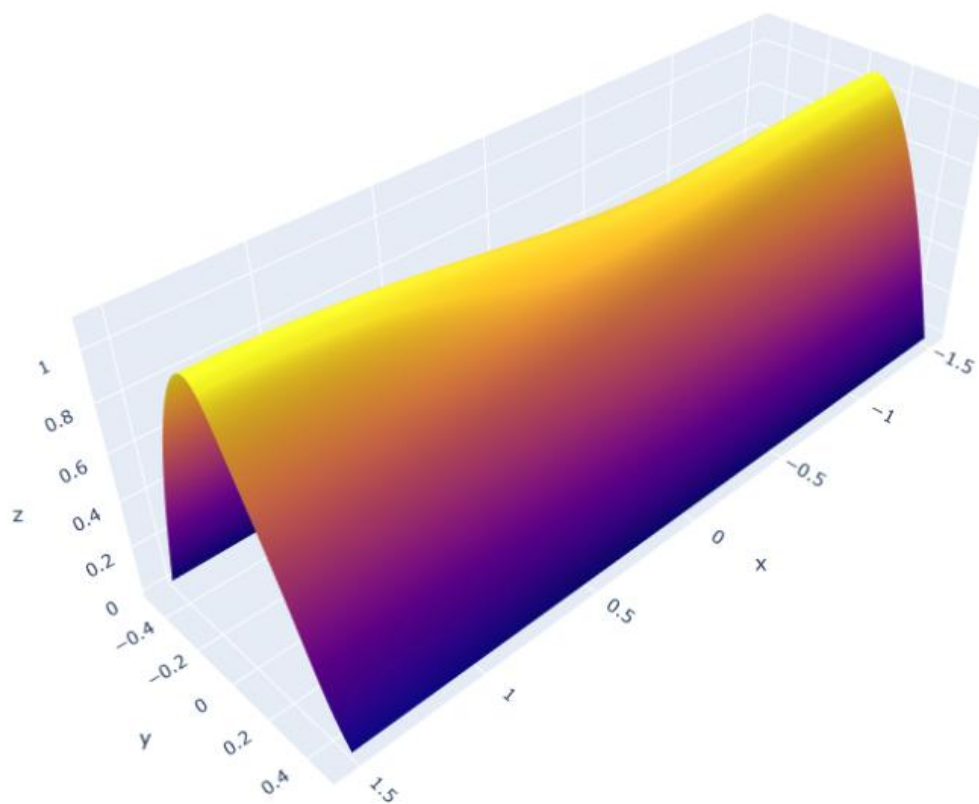
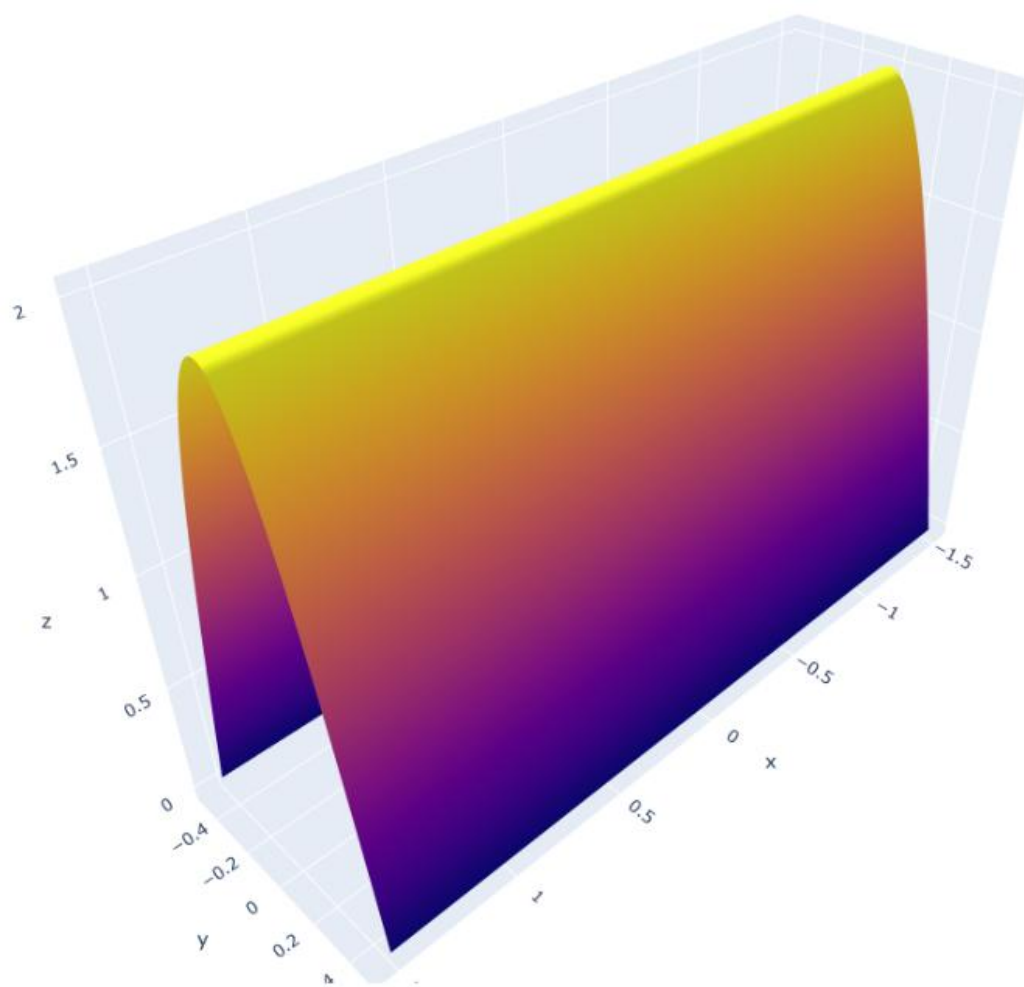
### Тест №1

$L = 25, u = 0.2, E = 120 \cdot 10^9, p = 7.4 \cdot 10^3;$

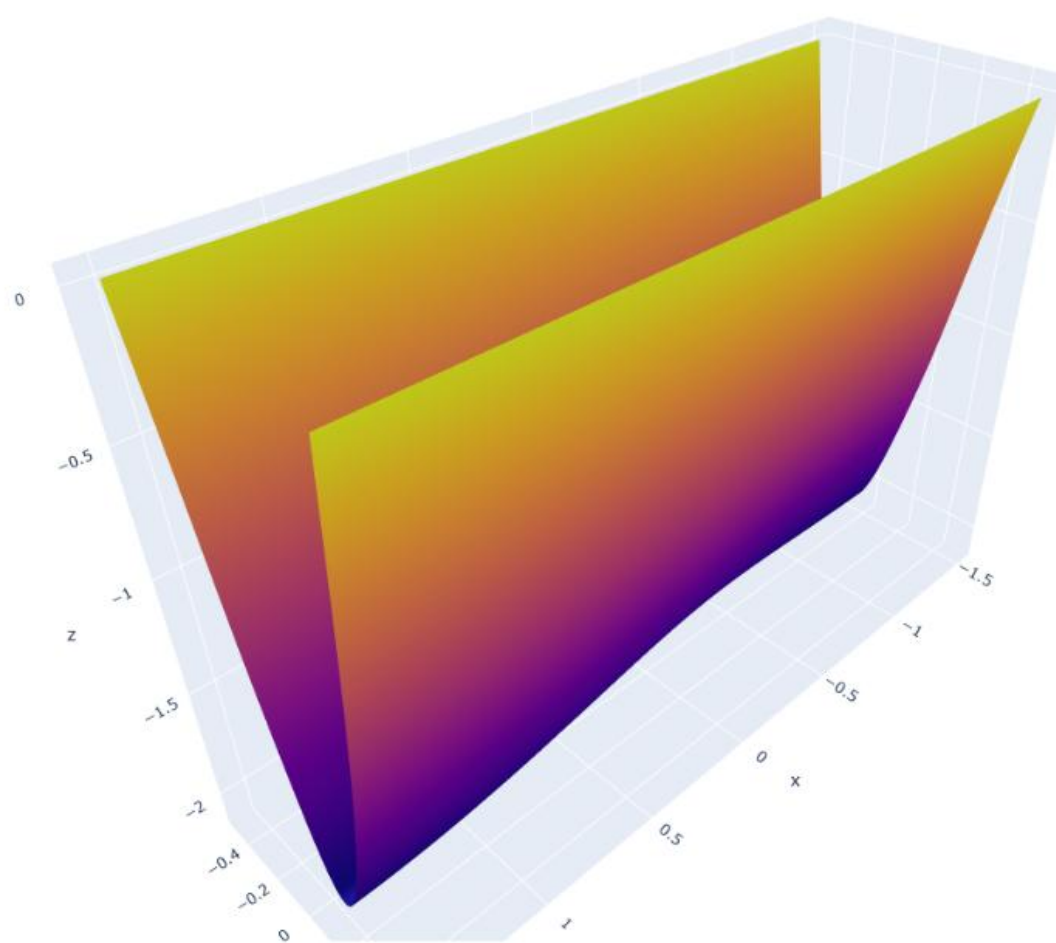
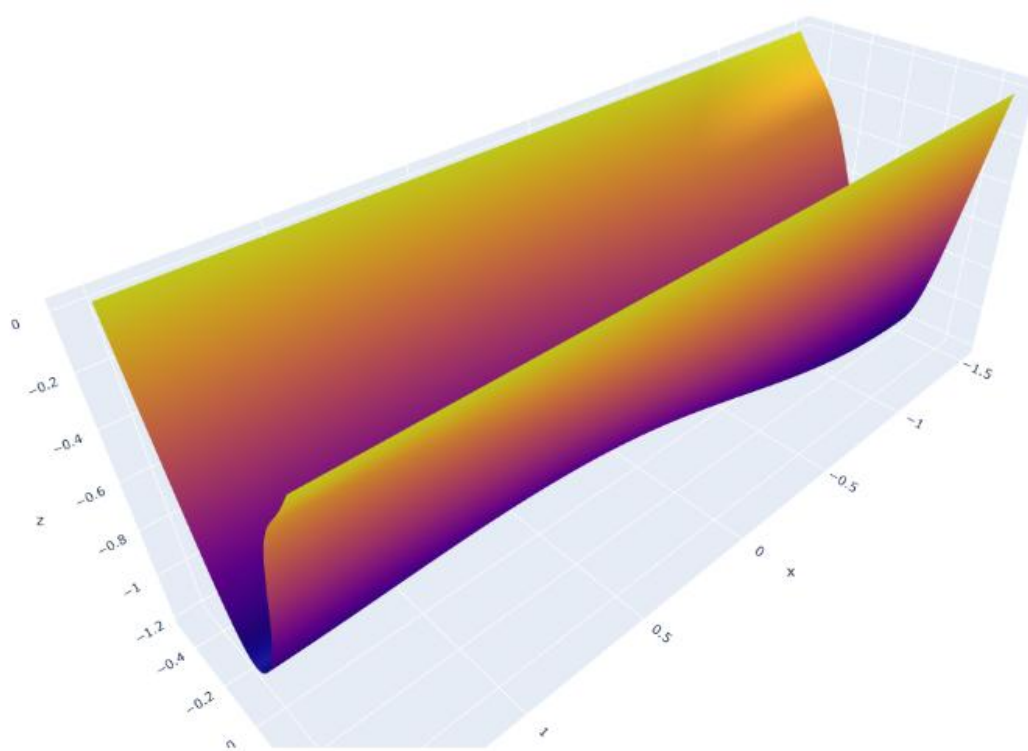


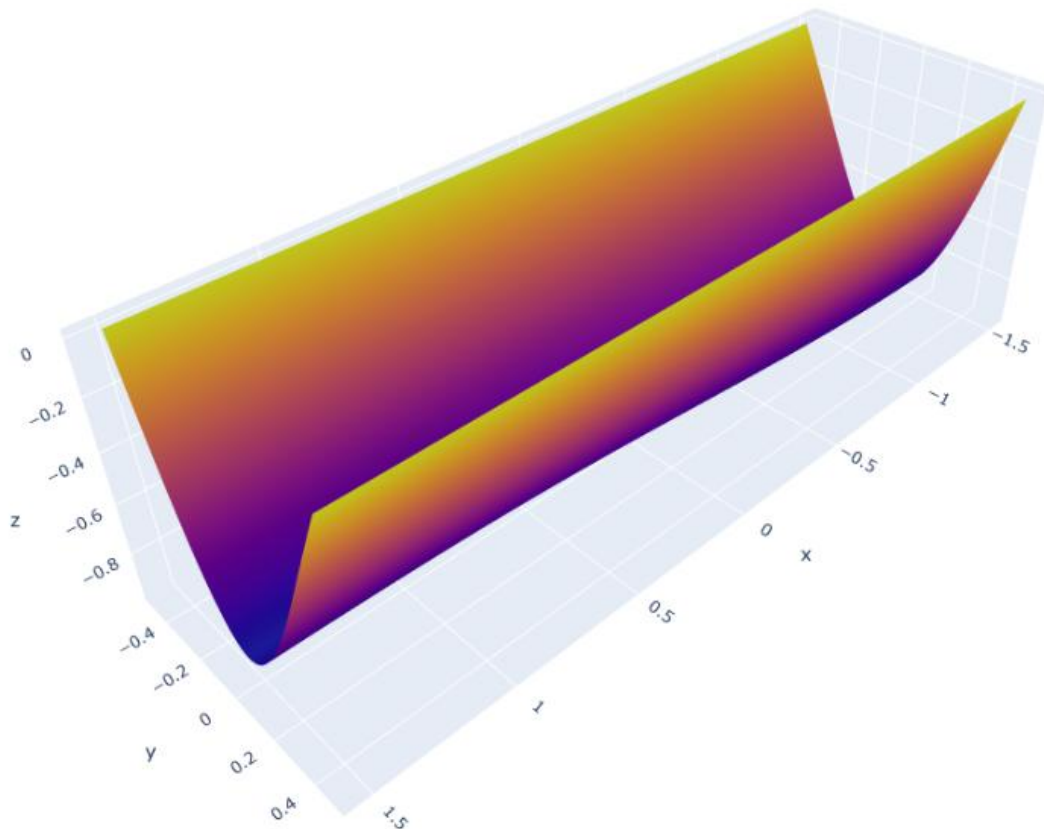


Тест №2









## Программная реализация

```
import math
import numpy as np
# import chart_studio.plotly as py
import plotly.graph_objs as graph_objs
import plotly
import time

a = 3
b = 1
T = 4

p = lambda x, y: 2 * np.cos(np.pi * x / a)
q = lambda x, y: np.tan(np.sin(2 * np.pi * x / a)) * np.sin(np.pi * y / b)

def solve(NX, NY, M, build_plot=False):
    tau = T / M
    hx = a / NX
    hy = b / NY
    C = tau / hx + tau / hy

    x_values = np.linspace(-a / 2, a / 2, NX)
    y_values = np.linspace(-b / 2, b / 2, NY)
    matrix = np.zeros((M, NX, NY))

    for i in range(NX):
        for j in range(NY):
            matrix[0][i][j] = p(x_values[i], y_values[j])

    for i in range(1, NX - 1):
        for j in range(1, NY - 1):
```

```

        matrix[1][i][j] = p(x_values[i], y_values[j]) + q(x_values[i],
y_values[j]) * tau
        matrix[1][i][j] += tau ** 2 / (2 * hx ** 2) * (
            matrix[0][i + 1][j] - 2 * matrix[0][i][j] +
matrix[0][i - 1][j])
        matrix[1][i][j] += tau ** 2 / (2 * hy ** 2) * (
            matrix[0][i][j + 1] - 2 * matrix[0][i][j] +
matrix[0][i][j - 1])

    matrix[1, 1:-1, 0] = matrix[1, 1:-1, 1]
    matrix[1, 1:-1, -1] = matrix[1, 1:-1, -2]

    for t in range(1, M - 1):
        matrix[t + 1, 1:-1, 1:-1] = 2 * matrix[t, 1:-1, 1:-1] - matrix[t -
1, 1:-1, 1:-1]
        matrix[t + 1, 1:-1, 1:-1] += tau ** 2 / hx ** 2 * (
            matrix[t, :-2, 1:-1] - 2 * matrix[t, 1:-1, 1:-1] +
matrix[t, 2:, 1:-1])
        matrix[t + 1, 1:-1, 1:-1] += tau ** 2 / hy ** 2 * (
            matrix[t, 1:-1, :-2] - 2 * matrix[t, 1:-1, 1:-1] +
matrix[t, 1:-1, 2:])
        matrix[t + 1, 1:-1, 0] = matrix[t + 1, 1:-1, 1]
        matrix[t + 1, 1:-1, -1] = matrix[t + 1, 1:-1, -2]

    for i in range(0, M, 999):
        x_grid, y_grid = np.meshgrid(x_values, y_values)
        surface = graph_objs.Surface(x=x_grid, y=y_grid, z=matrix[i])

        fig = graph_objs.Figure([surface])
        plotly.offline.plot(fig, auto_open=True)

        time.sleep(0.5)

solve(100, 100, 4000, build_plot=True)
import numpy as np
import matplotlib.pyplot as plt

# L = 6.0
# u = 0.15
# E = 82 * 10 ** 9
# ro = 9.7 * 10 ** 3
L = 25.0
u = 0.2
E = 120 * 10 ** 9
ro = 7.4 * 10 ** 3

p = lambda x: -4.0 * u / (L ** 2) * x ** 2 + 4 * u / L * x
q = lambda x: 0

def plot_solution_3d(solution: np.array, h: float, theta: float,
upper_bound: float) -> None:
    x = np.arange(0, L + h, h)
    y = np.arange(0, upper_bound + theta, theta)

    X, Y = np.meshgrid(x, y)
    figure = plt.figure(figsize=(10, 10))

    ax = figure.add_subplot(projection='3d')
    ax.plot_surface(X, Y, solution, antialiased=True)

```

```

plt.show()

def plot_solution_2d(solution: np.array, h: float, theta: float,
                    from_layer: int, to_layer: int, step_layer: int) ->
None:
    figure = plt.figure(figsize=(10, 10))
    x = np.arange(0, L + h, h)

    for i in range(from_layer, to_layer, step_layer):
        plt.plot(x, solution[i], label='Время={:01.5f}'.format(theta * i))
    plt.legend()
    plt.show()

def compute_first_and_second_layer(h: float, theta: float) -> tuple:
    x = np.arange(0, L + h, h)
    first_layer = np.zeros(x.size)
    first_layer[1:-1] = np.vectorize(p)(x[1:-1])

    second_layer = np.zeros(x.size)
    second_layer[1:-1] = first_layer[1:-1] + theta * np.vectorize(q)(x[1:-1]) + (theta ** 2 * E) / (2 * h ** 2 * ro) * \
        (first_layer[2:] - 2 * first_layer[1:-1] +
first_layer[:-2])

    return first_layer, second_layer

def compute_next_layer(previous_layer: np.array,
                        previous_previous_layer: np.array,
                        h: float, theta: float) -> np.array:
    new_layer = np.zeros(previous_layer.size)
    new_layer[1:-1] = (theta ** 2 * E) / (h ** 2 * ro) * \
        (previous_layer[2:] - 2 * previous_layer[1:-1] +
previous_layer[:-2]) + 2 * previous_layer[
1:-1] - previous_previous_layer[
1:-1]
    return new_layer

def solve(h: float, theta: float, upper_bound: float) -> np.array:
    layers = []
    previous_layer, current_layer = compute_first_and_second_layer(h,
theta)

    layers.append(previous_layer.copy())
    layers.append(current_layer.copy())
    for _ in np.arange(2 * theta, upper_bound + theta, theta):
        next_layer = compute_next_layer(current_layer, previous_layer, h,
theta)
        layers.append(next_layer.copy())
        previous_layer, current_layer = current_layer, next_layer
    return np.array(layers)

solution = solve(0.1, 0.00001, 2)
plot_solution_3d(solution, 0.1, 0.00001, 2)
plot_solution_2d(solution, 0.1, 0.00001, 2, 400, 50)

```

# Полученные результаты

## Задача 1

Продольные колебания  $u(x,t)$  тяги описываются уравнением

$$\frac{\partial^2 u}{\partial t^2} - \frac{\rho}{E} \frac{\partial^2 u}{\partial x^2} = 0, \quad 0 < x < L, \quad (2.73)$$

$$u(0,t)=0, \quad u(L,t)=0, \quad 0 \leq t \leq T, \quad (2.74)$$

где  $E$  –модуль упругости,  $\rho$  –плотность материала стержня. Тяга имеет длину  $L$  и закреплена на концах. Захватив тягу в центре (см. рис. 2.10),



Рис. 2.10

ее деформируют так, что продольное перемещение становится равным  $\Delta u$ :

$$u(x,0) = -\frac{\rho}{E} \frac{\partial^2 u}{\partial x^2}$$

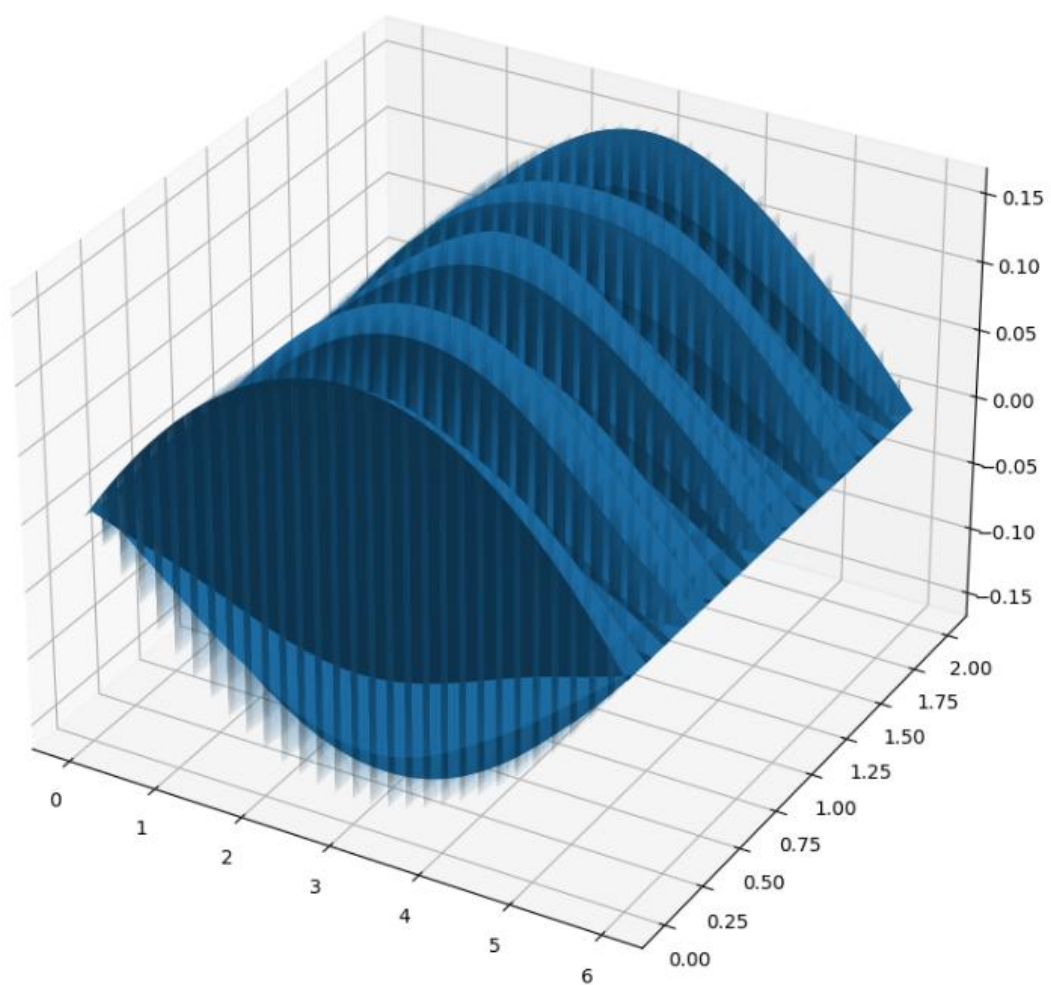
Затем тяга освобождается.

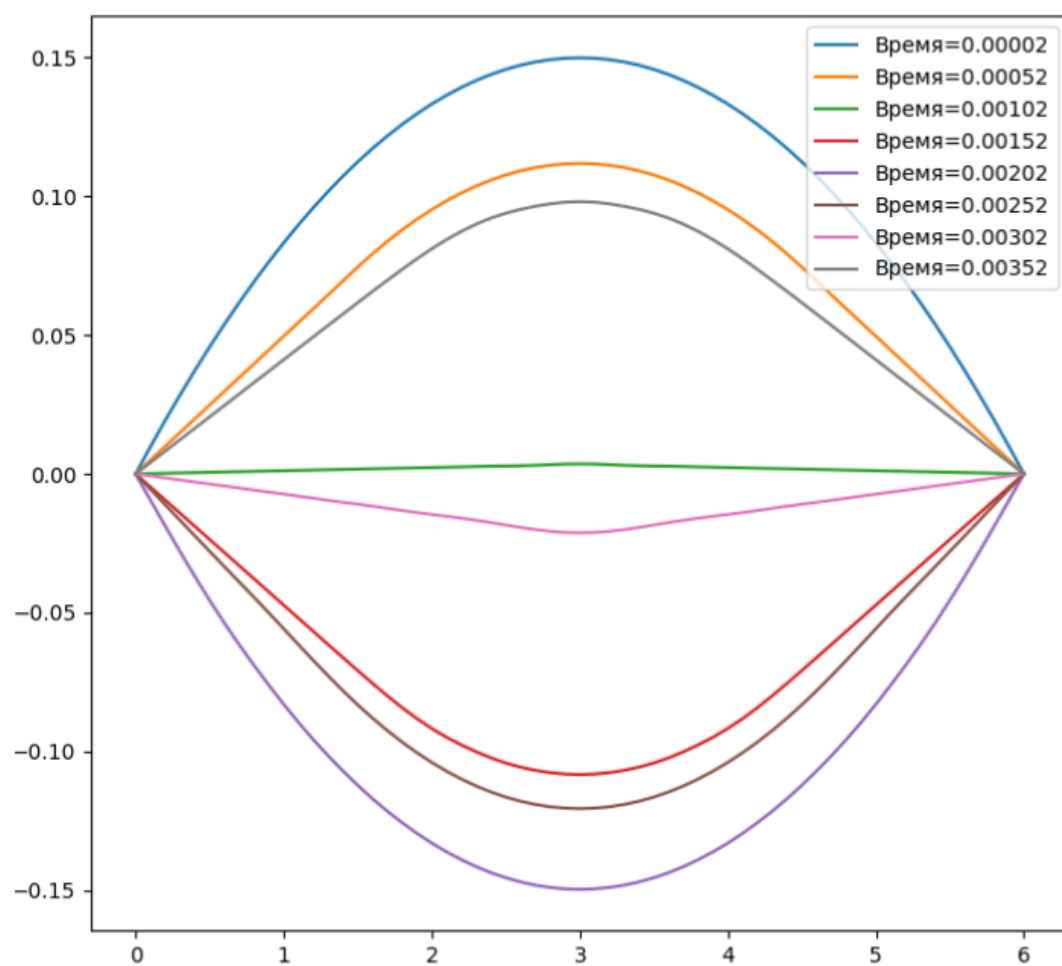
Рассчитайте колебания  $u(x,t)$  при заданных в таблице 2.12 параметрах.

**Таблица исходных данных к задаче 1**

Таблица 2.12

Параметр	В а р и а н т					
	9-1	9-2	9-3	9-4	9-5	9-6
$L$ , см	10	18	32	15	25	6
$\Delta u$ , см	0,1	0,2	0,15	0,1	0,2	0,15
$E$ , Н/м <sup>2</sup>	$110 \cdot 10^9$	$120 \cdot 10^9$	$97 \cdot 10^9$	$86 \cdot 10^9$	$120 \cdot 10^9$	$82 \cdot 10^9$
$\rho$ , кг/м <sup>3</sup>	$4,3 \cdot 10^3$	$5,9 \cdot 10^3$	$6,7 \cdot 10^3$	$8,5 \cdot 10^3$	$7,4 \cdot 10^3$	$9,7 \cdot 10^3$





## Задача 2 Рассчитать колебания тонкой пластины

Колебания тонкой пластины (см. рис.2.11) без учета потерь на трение описываются нормированным волновым уравнением вида

$$\frac{\partial^2 u}{\partial t^2} - \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0$$

где  $u(x, y, t)$  – деформация пластины,  $x, y$  – координаты,  $t$  – время.

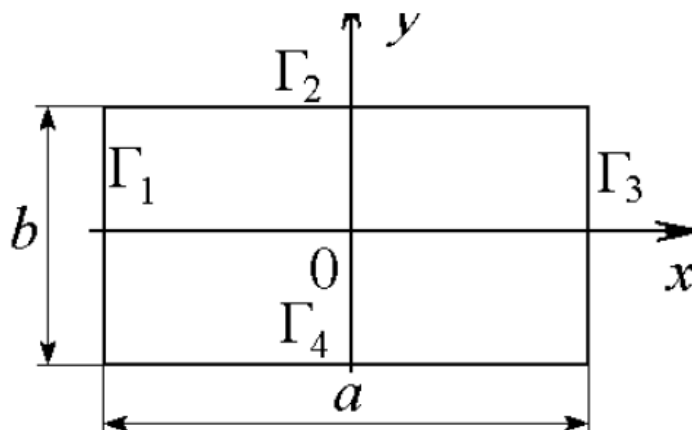


Рис.2.11

Рассчитать колебания пластины при заданных в таблице размерах  $a$  и  $b$ , граничных  $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ , и начальных  $u(x, y, 0)$  и  $\frac{\partial u(x, y, 0)}{\partial t}$  условиях.

таблица 2.1.

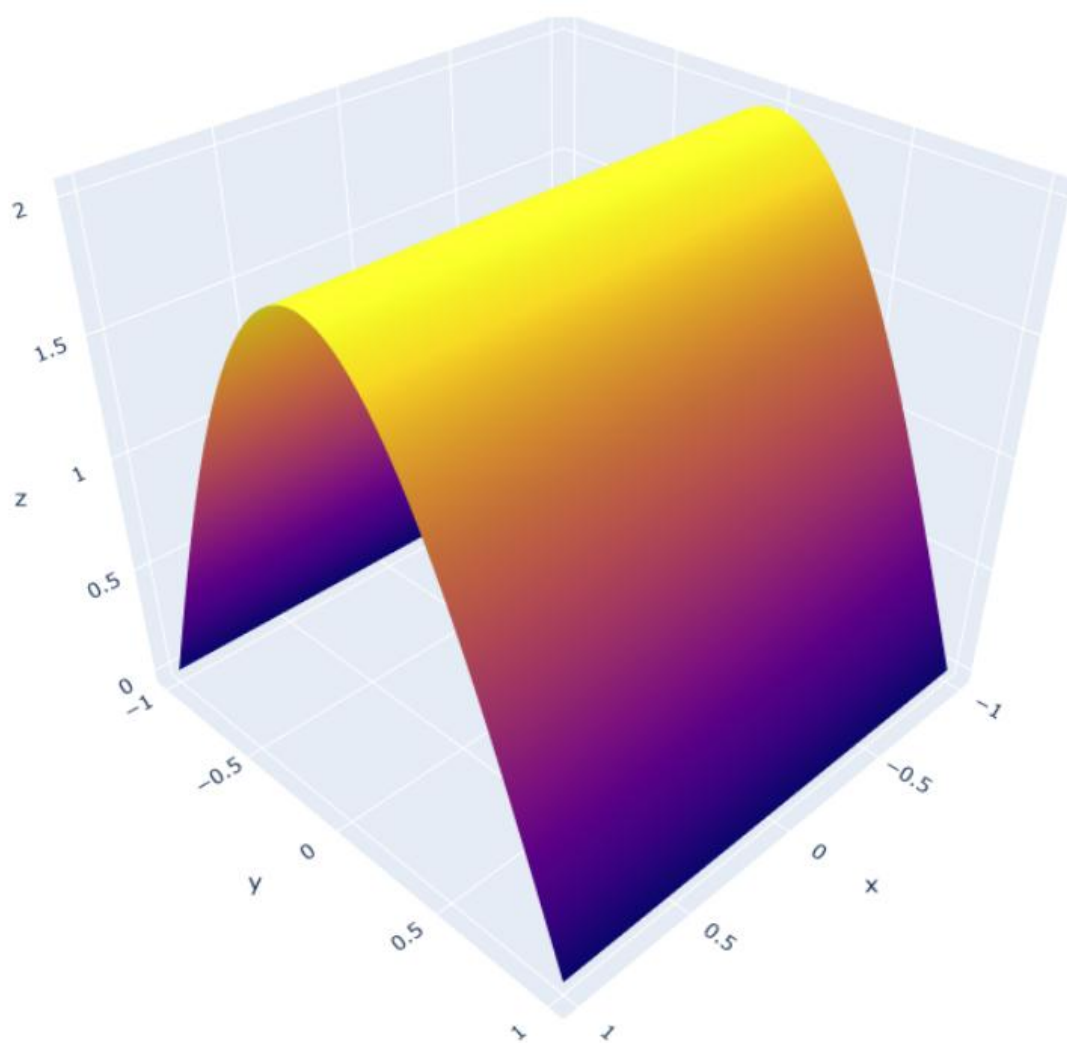
Таблица исходных данных к задаче2

Таблица 2.13

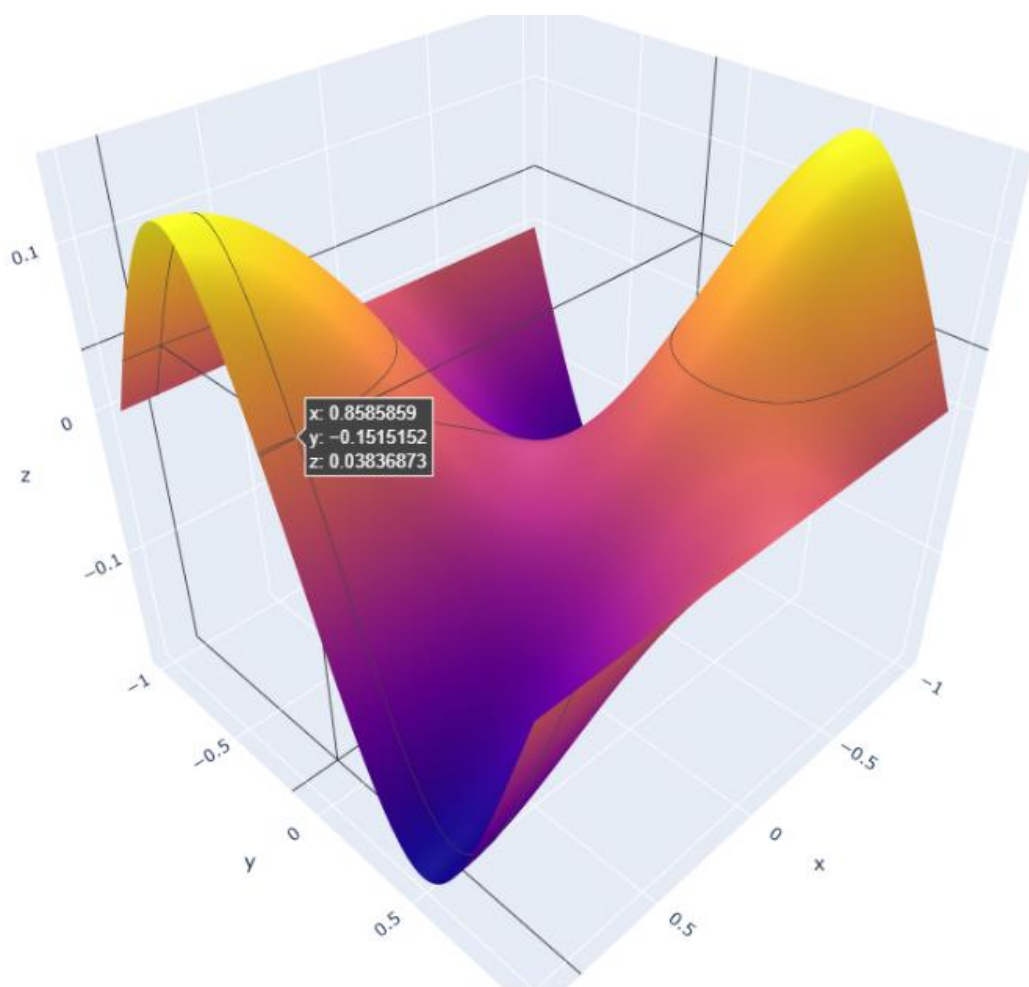
Параметр		В а р и а н т					
		10-1	10-2	10-3	10-4	10-5	10-6
$a$ , см		1	2	3	2	3	2
$b$ , см		2	1	2	3	1	2
Граничные условия	$\Gamma_1$	$u = 0$		$\partial u / \partial n = 0$		$u = 0$	
	$\Gamma_2$	$\partial u / \partial n = 0$		$u = 0$		$\partial u / \partial n = 0$	
	$\Gamma_3$	$u = 0$		$\partial u / \partial n = 0$		$u = 0$	
	$\Gamma_4$	$\partial u / \partial n = 0$		$u = 0$		$\partial u / \partial n = 0$	
$u(t=0)$		$\arctg[\cos(\pi x/a)]$		$\tg[\cos(\pi y/b)]$		$2\cos(\pi x/a)$	
$\partial u / \partial t _{t=0}$		$\sin\left(\frac{2\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$		$\exp\left[\sin\left(\frac{\pi x}{a}\right)\right]\sin\left(\frac{2\pi y}{b}\right)$		$\tg\left[\sin\left(\frac{2\pi x}{a}\right)\right]\sin\left(\frac{\pi y}{b}\right)$	

Момент времени  $T=0$

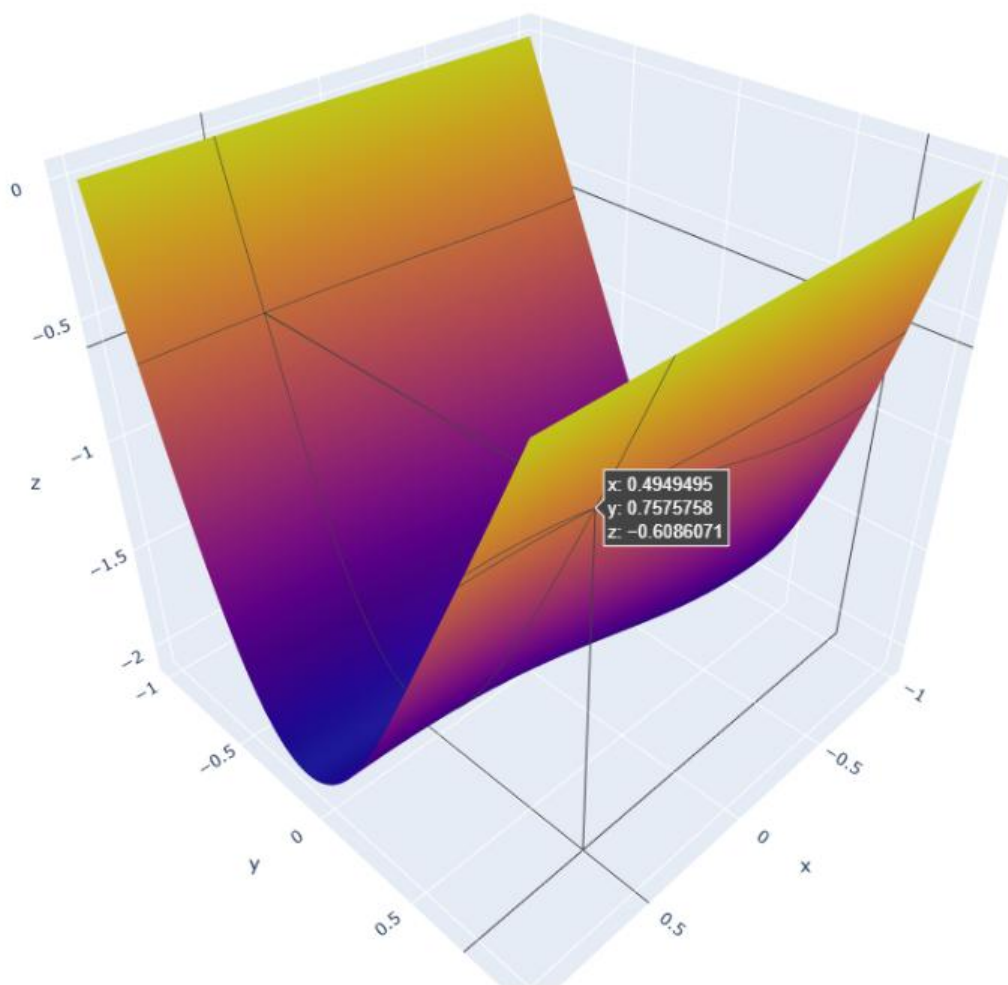




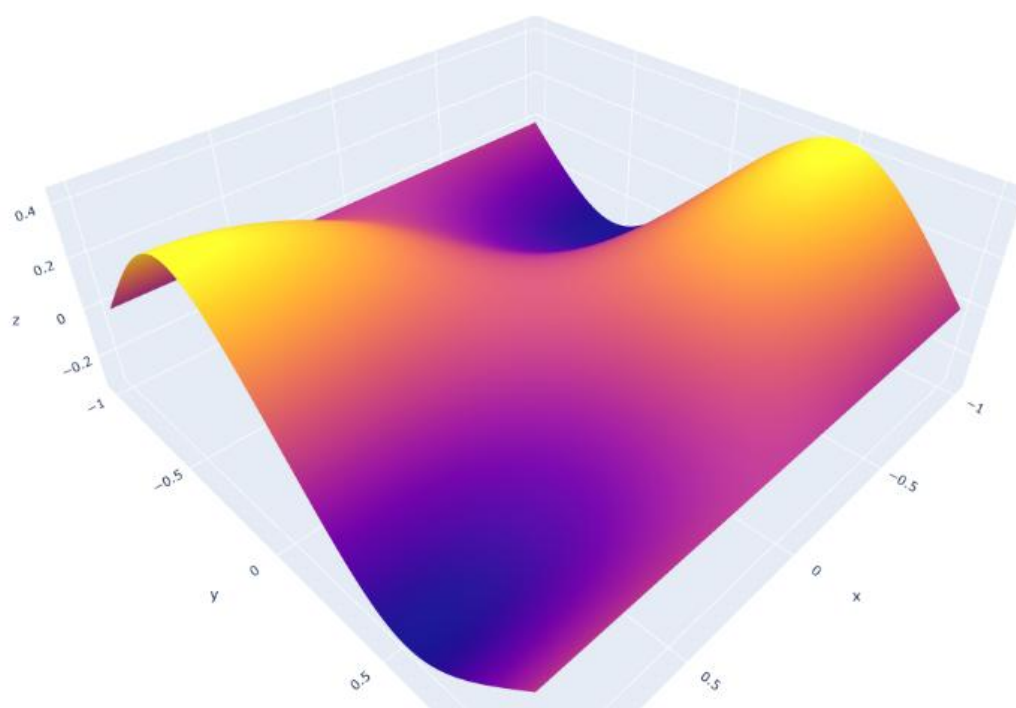
Момент времени  $T=1$



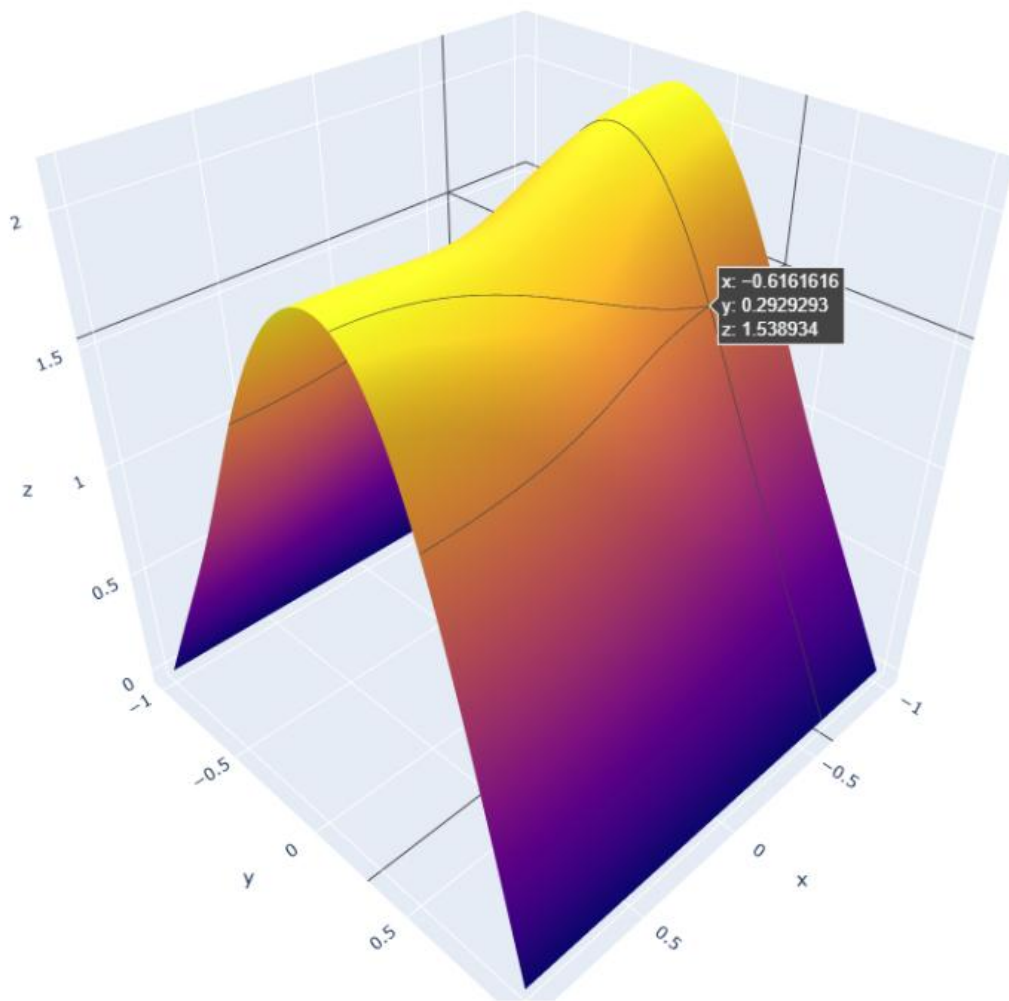
Момент времени  $T=2$



Момент времени  $T=3$



Момент времени  $T=4$



## Выводы

В результате выполнения лабораторной работы был исследован метод разностных аппроксимаций для решения волнового уравнения. Были составлены алгоритмы решения волнового уравнения методом сеток. Написаны программные реализации описанных алгоритмов, выполняющих поставленную задачу, корректность работы которых подтверждена тестовыми примерами.