

ФГАОУ ВО «САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО»

ФИЗИКО-МЕХАНИЧЕСКИЙ ИНСТИТУТ  
ВЫСШАЯ ШКОЛА ТЕОРЕТИЧЕСКОЙ МЕХАНИКИ

**Отчет по индивидуальному заданию**  
**Решение уравнения колебаний струны**  
**методом конечных разностей. Вариант 27**

Студент:

Чеботин А.А.

группа: 5030103/90301

Преподаватель:

Витохин Е.Ю

Санкт-Петербург, 2021 г.

# Содержание

<b>1.</b>	<b>2</b>
1.1. Формулировка задачи. . . . .	2
1.2. Описание численного метода . . . . .	2
1.3. Подготовка контрольных тестов . . . . .	5
1.4. Модульная структура программы . . . . .	5
1.5. Численный анализ решения задачи . . . . .	10
1.6. Выводы . . . . .	13

# Глава 1

## 1.1. Формулировка задачи.

Требуется получить конечно-разностное решение первой начально-краевой задачи для уравнения колебания струны

$$\frac{\partial^2 U}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 U}{\partial t^2}$$

С начальными условиями:

$$U(x, 0) = f(x) = (x^2 + 0.5)\cos(\pi x)$$

$$U_t(x, 0) = \Phi(x) = (x + 0.7)^2$$

И краевыми условиями:

$$U(0, t) = \phi(t) = 0.5$$

$$U(1, t) = \psi(t) = 2t - 1.5$$

Обозначения:

- $c = 1$  - скорость звука в среде (по условию задачи)

Решение получить с шагом  $h = 0.1$  по пространству,  $0 \leq x \leq 1$ ,  $0 \leq t \leq 0.5$ .

## 1.2. Описание численного метода

1. **Дискретизация.** Область непрерывного изменения значения аргументов  $x$  и  $t$  разбивается на конечное число интервалов, то есть на пространственно-временную область  $\Omega \times T = (0 \leq x \leq 1; 0 \leq t \leq 0.5)$  наносится конечно-разностная сетка

$$x_i = ih, \quad x_0 = 0, \quad x_n = 1, \quad i = 0, 1, \dots, n \quad (n = \frac{l}{h} = 11)$$

$$t^k = k\Delta t, \quad t^0 = 0, \quad t^k = 0.5, \quad k = 0, 1, \dots, K \quad (K = \frac{t^*}{\Delta t})$$

где  $k$  - номер слоя по времени,  $\Delta t$  - шаг интегрирования по времени,  $i$  - номер узла по пространству,  $h = 0.1$  - пространственный шаг.

Также вводятся два *временных слоя*. *Нижний*  $t^k = k\Delta t$ , на котором распределение искомой функции  $U(x_i, t^k)$  известно, и *верхний*  $t^{k+1} = (k+1)\Delta t$ , на котором распределение функции  $U(x_i, t^{k+1})$  нужно вычислить. Введем определение:

**Определение 1.1.** *Сеточной функцией*  $U_i^k$  назовем функцию, определенную только в дискретном множестве точек (в узлах) разностной сетки.

Таким образом, мы на сетке введем две сеточные функции, первая  $T_i^k$  из которых известна, а вторую  $T_i^{k+1}$  нужно найти методом конечных разностей.

2. **Аппроксимация.** Идея метода конечных разностей: вместо производных мы используем их конечно-разностные аналоги.

Шаг по времени можно задать с учетом условия Куранта:

$$\Delta t = C \frac{h}{c},$$

где  $C$  - число Куранта,  $c = 1$ .

- (а) **Явная схема.** Для определения  $U_i^{k+1}$  аппроксимируем дифференциалы уравнения колебания струны:

$$\frac{\partial^2 U}{\partial t^2} = \frac{U_i^{k+1} - 2U_i^k + U_i^{k-1}}{\Delta t^2}$$

$$\frac{\partial^2 U}{\partial x^2} = \frac{U_{i+1}^k - U_i^k + U_{i-1}^k}{h^2}$$

Подставляя эти аппроксимации в исходное уравнение, и выражая единственную неизвестную  $U_i^{k+1}$ , мы получим *явное* выражение для его определения:

$$U_i^{k+1} = \frac{c^2 \Delta t^2}{h^2} (U_{i+1}^k - U_i^k + U_{i-1}^k) + 2U_i^k - U_i^{k-1}$$

- (b) **Неявная схема.** Используя неявную схему, производные будут заменяться следующими выражениями

$$\frac{\partial^2 U}{\partial t^2} = \frac{U_i^{k+1} - 2U_i^k + U_i^{k-1}}{\Delta t^2}$$

$$\frac{\partial^2 U}{\partial x^2} = \frac{U_{i+1}^{k+1} - U_i^{k+1} + U_{i-1}^{k+1}}{h^2}$$

То есть мы дифференциал по  $x$  аппроксимировали соотношением на верхнем временном слое, в отличие от аппроксимации по явной схеме. После подставления данных аппроксимаций, приводим подобные слагаемые и получаем следующее разностное уравнение:

$$-AU_{i-1}^{k+1} + BU_i^{k+1} - CU_{i+1}^{k+1} = F_i,$$

$$A = \frac{1}{h^2}, \quad B = \frac{h^2 + 2c^2\Delta t^2}{\Delta t^2 h^2 c^2}, \quad C = \frac{1}{h^2}, \quad F_i = \frac{2U_i^k}{\Delta t^2 c^2} - \frac{U_i^{k-1}}{\Delta t^2 c^2}$$

Разностное уравнение есть СЛАУ с трехдиагональной матрицей, поэтому в случае неявной схемы нужно перейти к ее решению (этап 3).

3. **Решение СЛАУ.** В результате аппроксимации мы свели исходную задачу к системе алгебраических уравнений. Образованная этими уравнениями СЛАУ имеет трехдиагональную матрицу, поэтому будем применять метод прогонки, алгоритм которого имеет сложность  $O(n)$ .

СЛАУ МКР будет иметь следующую структуру:

$$\begin{pmatrix} B & -C & 0 & 0 & \dots & 0 \\ -A & B & -C & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -A & B & -C \\ 0 & \dots & 0 & 0 & -A & B \end{pmatrix} \cdot \begin{pmatrix} U_1^{k+1} \\ U_2^{k+1} \\ \vdots \\ U_{n-1}^{k+1} \\ U_n^{k+1} \end{pmatrix} = \begin{pmatrix} F_1^{k+1} \\ F_2^{k+1} \\ \vdots \\ F_{n-1}^{k+1} \\ F_n^{k+1} \end{pmatrix}$$

Алгоритм метода прогонки. Мы должны найти прогоночные коэффициенты.

(а) Прямой ход.

$$i = 1 : \quad P_1 = \frac{C}{B}, \quad Q_1 = \frac{F_1}{B}$$

$$i = 2, \dots, n : \quad P_i = \frac{C}{B - AP_{i-1}}, \quad Q_i = \frac{F_i + AQ_{i-1}}{B - AP_{i-1}}$$

(б) Обратный ход. Находим искомое решение, используя найденные прогоночные коэффициенты

$$U_i^{k+1} = P_i U_{i+1}^{k+1} + Q_i$$

### 1.3. Подготовка контрольных тестов

Для исследования метода будем проводить следующие тесты:

1. Построим поверхности явной и неявной схем;
2. Построим разрезы, зафиксировав определенный момент времени;
3. Сравним схемы численно.

### 1.4. Модульная структура программы

Программа написана на языке C++. В коде имеются различные пользовательские типы:

- `enum ListOfMediums` - перечисление, состоящее из набора целочисленных констант. В нашем случае перечислителями будут различные среды.
- `struct MediumData` - простейшая структура, имеющая два элемента: (`const int MediumID` - идентификатор среды, `const double cSoundWaveSpeed` - скорость звука в среде);
- `namespace MediumDataBase` - пространство имен. Здесь используется контейнер `vector` из стандартной библиотеки C++. Мы используем вектор структур `MediumData` и при инициализации задаем значения для всех элементов структуры `MediumData`;
- `struct IntegrateParameters` - простейшая структура, содержащая два элемента - шаг интегрирования по времени и по координате  $x$ ;
- `class StringVibrationSolver` - основной класс программы, решающий уравнение колебания струны. Класс имеет свои функции - функции начальных и краевых условий, функции создания сетки, функции получения решения по явной и неявной схеме. Для передачи структур в функции класса `StringVibrationSolver` (например, структуру с определенной средней и структуру с параметрами интегрирования) используется константная ссылка, чтобы повысить эффективность алгоритма и избавиться от дополнительных затрат. Обычные переменные будут передаваться по значению, не затрачивая при этом расходы на дополнительный доступ в функции класса.

Скриншоты программы:

```

WaveEquation.cpp*
Mechanic_lab1 (Глобальная область) main()
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <cmath>
5  #include <vector>
6  #include <array>
7  #include <stdint.h>
8
9  #define _USE_MATH_DEFINES
10
11 #if defined(_USE_MATH_DEFINES) && !defined(_MATH_DEFINES_DEFINED)
12     #define _MATH_DEFINES_DEFINED
13     #define M_PI 3.14159265358979323846
14     #define M_PI_2 1.57079632679489661923
15     #define M_PI_4 0.785398163397448309616
16     #define M_1_PI 0.318309886183790671538
17     #define M_2_PI 0.636619772367581343076
18
19 #endif /* _USE_MATH_DEFINES */
20
21
22 #define N 11 // 0 < x < 1
23 #define K 51 // 0 < t < 0.5
24
25 enum ListOfMediums {
26     DryAir = 0,
27     Hydrogen = 1,
28     Nitrogen = 2,
29     Helium = 3,
30     Oxygen = 4,
31     CarbonDioxide = 5,
32
33     Acetone = 6,
34     Water = 7,
35     Petrol = 8,
36     Vodka = 9,
37
38     One = 10 //by exercise c = 1
39 };
40
41 struct MediumData {
42     const int Medium_ID;
43     const double c_SoundWaveSpeed;
44 };
45
46 namespace MediumDataBase {
47     std::vector<MediumData> mediums = {
48         {0, 331.0},
49         {1, 1284.0},
50         {2, 334.0},
51         {3, 955.0},
52         {4, 316.0},
53         {5, 259.0},
54
55         {6, 1192.0},
56         {7, 1460.0},
57         {8, 1170.0},
58         {9, 1180.0},
59
60         {10, 1}
61     };
62 }
63
64 struct IntegrateParameters {
65     double delta_tStep;
66     double h_xStep;
67 };
68

```

```

69 class StringVibrationSolver {
70
71 private:
72     inline double InitialValueFunction(double x);
73     inline double DerivativeIVF(double x);
74     inline double LeftBoundaryConditionFunction(double t);
75     inline double RightBoundaryConditionFunction(double t);
76
77 public:
78     void SetCoordinateXGrid(std::array<double, N>& x, const IntegrateParameters& params);
79     void SetTimeGrid(std::array<double, K>& t, const IntegrateParameters& params);
80
81     void getSolutionExplicit(const MediumData& medium, double(*U)[N], std::array<double, N>& x,
82                             std::array<double, K>& t, const IntegrateParameters& params);
83     void getSolutionImplicit(const MediumData& medium, double(*U)[N], std::array<double, N>& x,
84                             std::array<double, K>& t, const IntegrateParameters& params);
85 };
86
87
88 //ИУ T(x,0)
89 inline double StringVibrationSolver::InitialValueFunction(double x)
90 {
91     return (x * x + 0.5) * cos(M_PI * x);
92 }
93
94 //derived NU
95 inline double StringVibrationSolver::DerivativeIVF(double x)
96 {
97     return (x + 0.7) * (x + 0.7);
98 }
99
100
101 //ГV1 T(0, t)
102 inline double StringVibrationSolver::LeftBoundaryConditionFunction(double t)
103 {
104     return 0.5;
105 }
106
107 //ГV2 T(l,t)
108 inline double StringVibrationSolver::RightBoundaryConditionFunction(double t)
109 {
110     return 2 * t - 1.5;
111 }
112
113 //Создание сетки по координате x
114 void StringVibrationSolver::SetCoordinateXGrid(std::array<double, N>& x, const IntegrateParameters& params)
115 {
116     x[0] = 0;
117     for (uint8_t i = 1; i < N; ++i) {
118         x[i] = i * params.h_xStep;
119     }
120 }
121

```



```

122 void StringVibrationSolver::SetTimeGrid(std::array<double, K>& t, const IntegrateParameters& params)
123 {
124     t[0] = 0;
125     for (uint8_t k = 1; k < K; ++k) {
126         t[k] = k * params.delta_tStep;
127     }
128 }
129
130 void StringVibrationSolver::getSolutionExplicit(const MediumData& medium, double(*U)[N], std::array<double, N>& x,
131 std::array<double, K>& t, const IntegrateParameters& params)
132 {
133     for (uint8_t i = 0; i < N; ++i)
134     {
135         U[0][i] = InitialValueFunction(x[i]);
136         U[1][i] = InitialValueFunction(x[i]) + DerivativeIVF(x[i]) * params.delta_tStep;
137     }
138
139     for (uint8_t k = 0; k < K; ++k)
140     {
141         U[k][0] = LeftBoundaryConditionFunction(t[k]);
142         U[k][N-1] = RightBoundaryConditionFunction(t[k]);
143     }
144
145     /*Для сокращения выражений*/
146     double h = params.h_xStep;
147     double dt = params.delta_tStep;
148     double c = medium.c_SoundWaveSpeed;
149
150     /*Решение по явной схеме*/
151     for (uint8_t k = 1; k < K-1; ++k)
152     {
153         for (uint8_t i = 1; i < N - 1; ++i)
154         {
155             U[k+1][i] = ((c * c) * (dt * dt) / (h * h)) * (U[k][i+1] - 2 * U[k][i] + U[k][i-1]) +
156                 2 * U[k][i] - U[k-1][i];
157         }
158     }
159
160 void StringVibrationSolver::getSolutionImplicit(const MediumData& medium, double(*U)[N], std::array<double, N>& x,
161 const IntegrateParameters& params)
162 {
163     for (uint8_t i = 0; i < N; ++i)
164     {
165         U[0][i] = InitialValueFunction(x[i]);
166         U[1][i] = InitialValueFunction(x[i]) + DerivativeIVF(x[i]) * params.delta_tStep;
167     }
168
169     for (uint8_t k = 0; k < K; ++k)
170     {
171         U[k][0] = LeftBoundaryConditionFunction(t[k]);
172         U[k][N - 1] = RightBoundaryConditionFunction(t[k]);
173     }
174
175
176
177
178
179

```

```

180     double h = params.h_xStep;
181     double dt = params.delta_tStep;
182     double c = medium.c_SoundWaveSpeed;
183
184     double A = 1.0 / (h * h);
185     double C = 1.0 / (h * h);
186     double B = ((h*h) + 2 * (c*c) * (dt*dt)) / ((h*h) * (c*c) * (dt*dt));
187     std::array<double, N> F, P, Q;
188
189     for (uint8_t k = 1; k < K - 1; ++k)
190     {
191         for (uint8_t i = 0; i < N; i++)
192         {
193             F[i] = -((U[k - 1][i]) / (c * c * dt * dt)) + U[k][i] * (2 / (c * c * dt * dt));
194         }
195         P[0] = C / B;
196         Q[0] = F[0] / B;
197
198         for (uint8_t j = 1; j < N; ++j)
199         {
200             P[j] = C / (B - A * P[j - 1]);
201             Q[j] = (F[j] + A * Q[j - 1]) / (B - A * P[j - 1]);
202         }
203
204         for (uint8_t j = N - 2; j > 0; --j)
205         {
206             U[k + 1][j] = P[j] * U[k + 1][j + 1] + Q[j];
207         }
208     }
209
210
211
212 }
213
214
215 int main()
216 {
217     using namespace MediumDataBase;
218
219     StringVibrationSolver mySolver;
220     IntegrateParameters params;
221
222     params.h_xStep = 0.1;
223     params.delta_tStep = 0.01;
224
225     double U[K][N] = { 0 };
226     std::array<double, N> x;
227     std::array<double, K> t;
228
229     mySolver.SetCoordinateXGrid(x, params);
230     mySolver.SetTimeGrid(t, params);
231
232     /*Здесь можно задать материал (см. enum)*/
233     mySolver.getSolutionImplicit mediums[One], U, x, t, params);
234

```

## 1.5. Численный анализ решения задачи

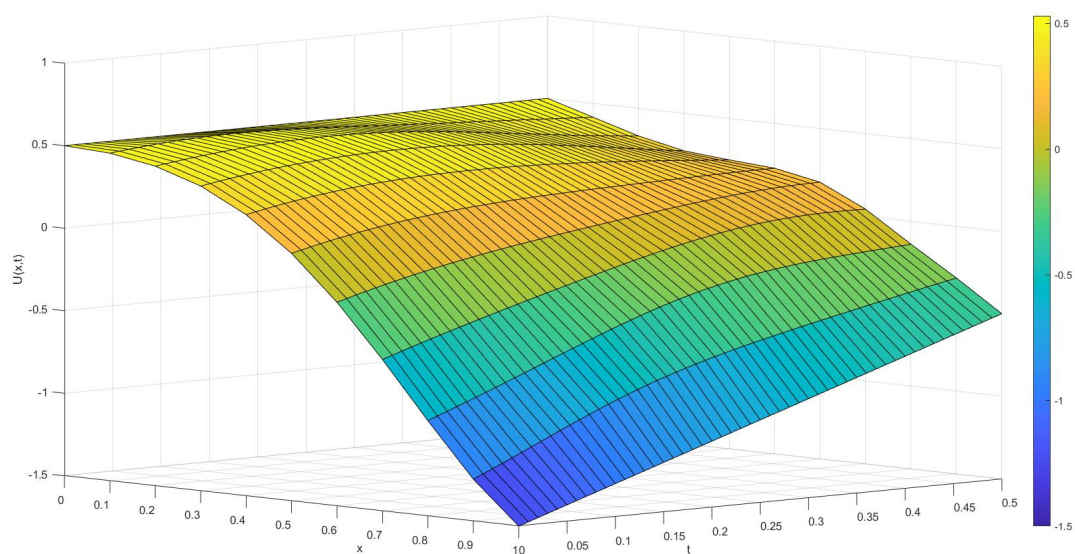


Рис 1.1 Поверхность (явная схема)

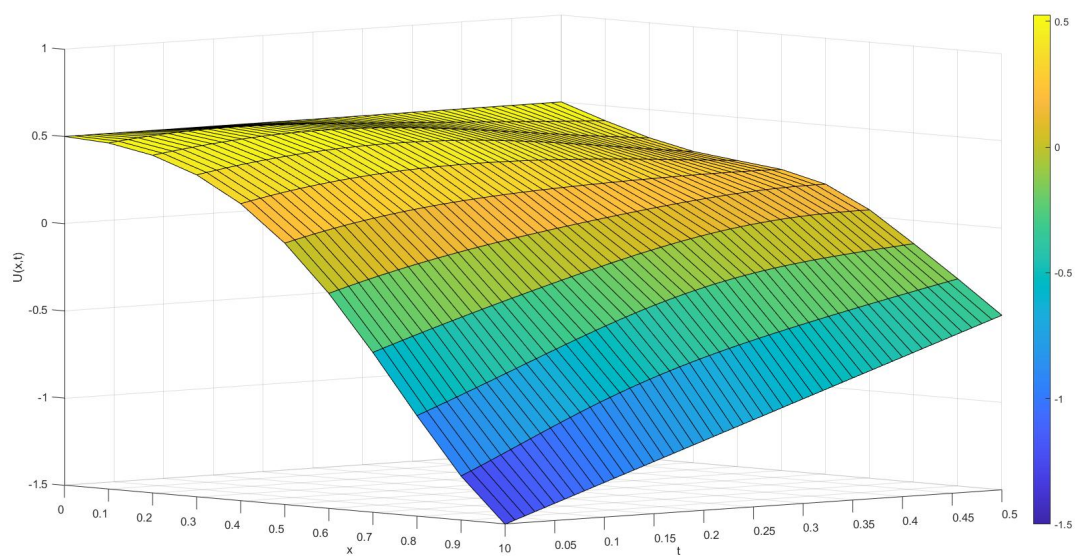
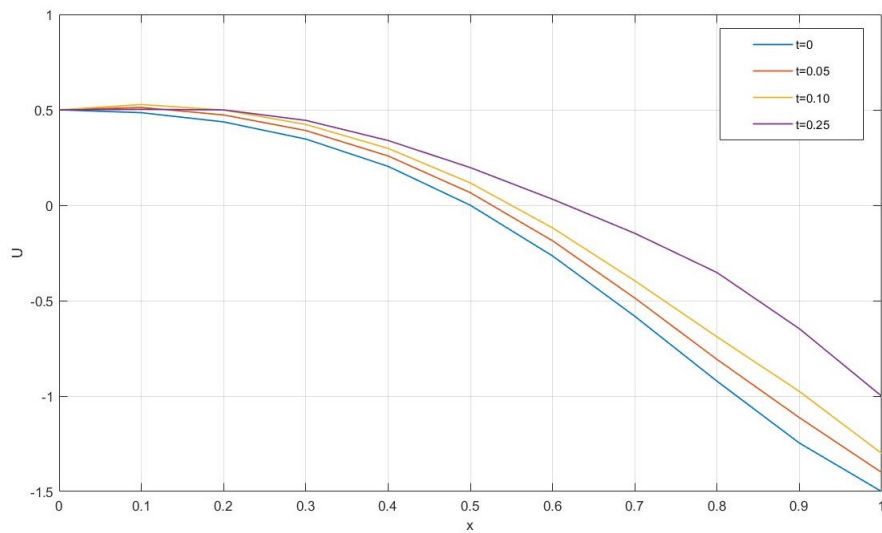
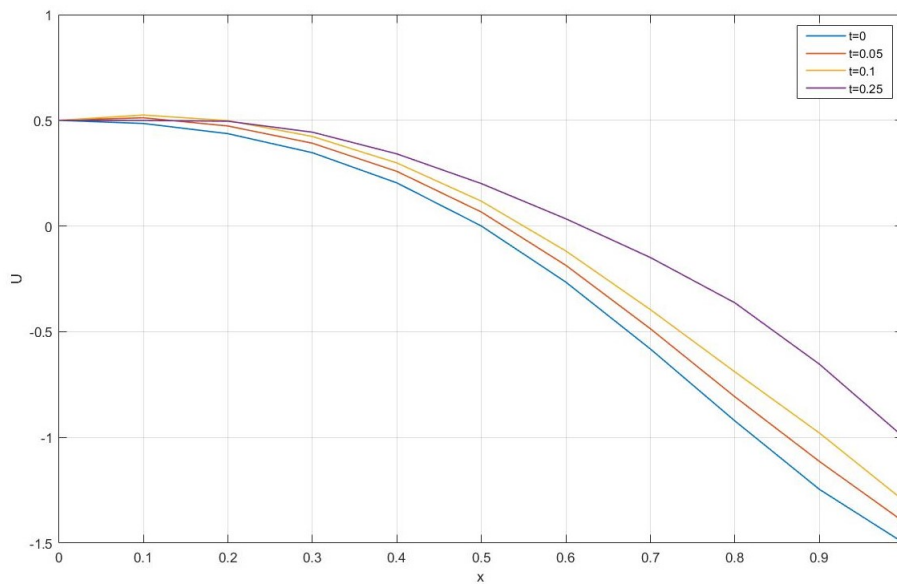
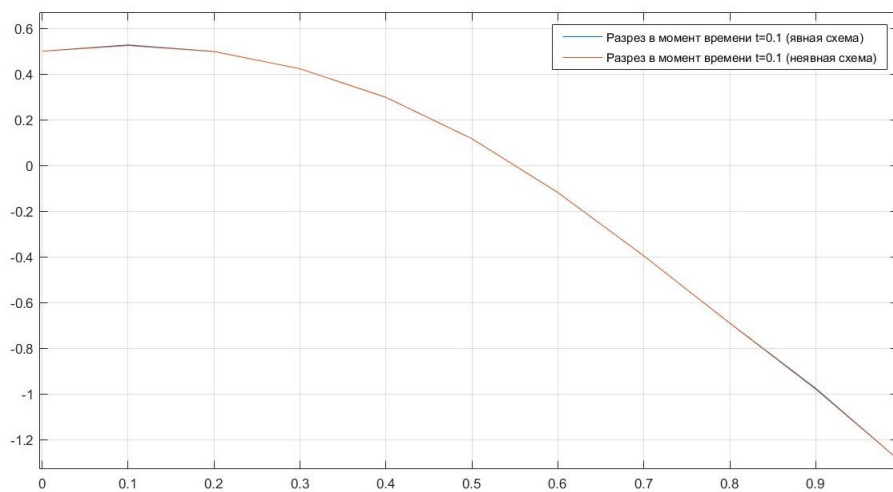


Рис 1.2. Поверхность (неявная схема)

Рис 2.1. Распределение  $U$  в разные моменты времени (явная схема)Рис 2.2. Распределение  $U$  в разные моменты времени (неявная схема)Рис 3. Сравнение схем в момент времени  $t = 0.1$ .

Сравним численно:

Таблица 1. Явная схема

	$x_0 = 0$	$x_1 = 0.1$	$x_2 = 0.2$	$x_3 = 0.3$	$x_4 = 0.4$	$x_5 = 0.5$	$x_6 = 0.6$	$x_7 = 0.7$	$x_8 = 0.8$	$x_9 = 0.9$	$x_{10} = 1$
$t_0 = 0.00$	0.500000	0.485039	0.436869	0.346793	0.203951	0.000000	-0.265755	-0.581907	-0.922279	-1.245884	-1.500000
$t_1 = 0.01$	0.500000	0.491439	0.444969	0.356793	0.216051	0.014400	-0.248855	-0.562307	-0.899779	-1.220284	-1.480000
$t_2 = 0.02$	0.500000	0.497460	0.452652	0.366268	0.227542	0.028184	-0.232457	-0.542948	-0.877110	-1.194076	-1.460000
$t_3 = 0.03$	0.500000	0.503058	0.459919	0.375219	0.238427	0.041355	-0.216557	-0.523824	-0.854268	-1.167358	-1.440000
$t_4 = 0.04$	0.500000	0.508194	0.466771	0.383649	0.248708	0.053918	-0.201151	-0.504933	-0.831253	-1.140235	-1.420000
$t_5 = 0.05$	0.500000	0.512834	0.473205	0.391560	0.258392	0.065878	-0.186232	-0.486267	-0.808064	-1.112820	-1.400000
$t_6 = 0.06$	0.500000	0.516950	0.479220	0.398957	0.267482	0.077242	-0.171793	-0.467819	-0.784705	-1.085229	-1.380000
$t_7 = 0.07$	0.500000	0.520518	0.484809	0.405841	0.275984	0.088018	-0.157823	-0.449579	-0.761183	-1.057581	-1.360000
$t_8 = 0.08$	0.500000	0.523525	0.489965	0.412217	0.283905	0.098215	-0.144313	-0.431538	-0.737508	-1.029993	-1.340000
$t_9 = 0.09$	0.500000	0.525960	0.494680	0.418087	0.291252	0.107844	-0.131249	-0.413684	-0.713699	-1.002580	-1.320000
$t_{10} = 0.10$	0.500000	0.527824	0.498941	0.423455	0.298034	0.116916	-0.118619	-0.396006	-0.689778	-0.975453	-1.300000
$t_{11} = 0.11$	0.500000	0.529120	0.502737	0.428323	0.304258	0.125444	-0.106407	-0.378492	-0.665776	-0.948714	-1.280000
$t_{12} = 0.12$	0.500000	0.529861	0.506052	0.432694	0.309936	0.133442	-0.094598	-0.361130	-0.641731	-0.922459	-1.260000
$t_{13} = 0.13$	0.500000	0.530065	0.508872	0.436572	0.315075	0.140924	-0.083174	-0.343908	-0.617687	-0.896772	-1.240000
$t_{14} = 0.14$	0.500000	0.529757	0.511180	0.439958	0.319689	0.147906	-0.072116	-0.326817	-0.593696	-0.871726	-1.220000
$t_{15} = 0.15$	0.500000	0.528965	0.512962	0.442853	0.323787	0.154406	-0.061405	-0.309848	-0.569817	-0.847383	-1.200000
$t_{16} = 0.16$	0.500000	0.527724	0.514203	0.445259	0.327382	0.160442	-0.051020	-0.292994	-0.546113	-0.823790	-1.180000
$t_{17} = 0.17$	0.500000	0.526070	0.514890	0.447175	0.330486	0.166033	-0.040940	-0.276251	-0.522655	-0.800983	-1.160000
$t_{18} = 0.18$	0.500000	0.524044	0.515012	0.448602	0.333113	0.171199	-0.031144	-0.259620	-0.499517	-0.778982	-1.140000
$t_{19} = 0.19$	0.500000	0.521687	0.514559	0.449537	0.335275	0.175960	-0.021609	-0.243102	-0.476774	-0.757797	-1.120000
$t_{20} = 0.20$	0.500000	0.519042	0.513528	0.449981	0.336987	0.180338	-0.012313	-0.226707	-0.454504	-0.737424	-1.100000
$t_{21} = 0.21$	0.500000	0.516151	0.511916	0.449930	0.338262	0.184357	-0.003235	-0.210445	-0.432786	-0.717847	-1.080000
$t_{22} = 0.22$	0.500000	0.513057	0.509727	0.449382	0.339115	0.188039	0.005648	-0.194335	-0.411695	-0.699042	-1.060000
$t_{23} = 0.23$	0.500000	0.509798	0.506968	0.448335	0.339560	0.191407	0.014354	-0.178398	-0.391304	-0.680972	-1.040000
$t_{24} = 0.24$	0.500000	0.506414	0.503651	0.446786	0.339611	0.194487	0.022903	-0.162663	-0.371680	-0.663596	-1.020000
$t_{25} = 0.25$	0.500000	0.502937	0.499792	0.444735	0.339283	0.197302	0.031312	-0.147163	-0.352886	-0.646865	-1.000000
$t_{26} = 0.26$	0.500000	0.499400	0.495415	0.442179	0.338589	0.199877	0.039597	-0.131935	-0.334974	-0.630725	-0.980000
$t_{27} = 0.27$	0.500000	0.495829	0.490545	0.439120	0.337544	0.202236	0.047769	-0.117022	-0.317989	-0.615121	-0.960000
$t_{28} = 0.28$	0.500000	0.492247	0.485214	0.435559	0.336162	0.204404	0.055838	-0.102471	-0.301966	-0.599994	-0.940000
$t_{29} = 0.29$	0.500000	0.488672	0.479456	0.431501	0.334456	0.206404	0.063809	-0.088331	-0.286928	-0.585287	-0.920000
$t_{30} = 0.30$	0.500000	0.485118	0.473311	0.426952	0.332440	0.208258	0.071685	-0.074657	-0.272888	-0.570944	-0.900000
$t_{31} = 0.31$	0.500000	0.481595	0.466821	0.421922	0.330128	0.209988	0.079463	-0.061501	-0.259846	-0.556910	-0.880000
$t_{32} = 0.32$	0.500000	0.478108	0.460029	0.416423	0.327532	0.211614	0.087137	-0.048919	-0.247791	-0.543137	-0.860000
$t_{33} = 0.33$	0.500000	0.474659	0.452982	0.410470	0.324665	0.213155	0.094695	-0.036965	-0.236701	-0.529579	-0.840000
$t_{34} = 0.34$	0.500000	0.471247	0.445727	0.404085	0.321542	0.214627	0.102121	-0.025692	-0.226542	-0.516196	-0.820000
$t_{35} = 0.35$	0.500000	0.467867	0.438310	0.397291	0.318175	0.216042	0.109394	-0.015149	-0.217272	-0.502955	-0.800000
$t_{36} = 0.36$	0.500000	0.464513	0.430779	0.390116	0.314578	0.217412	0.116488	-0.005382	-0.208837	-0.489827	-0.780000
$t_{37} = 0.37$	0.500000	0.461177	0.423178	0.382592	0.310764	0.218745	0.123373	0.003569	-0.201177	-0.476792	-0.760000
$t_{38} = 0.38$	0.500000	0.457848	0.415552	0.374755	0.306749	0.220044	0.130013	0.011671	-0.194226	-0.463832	-0.740000
$t_{39} = 0.39$	0.500000	0.454519	0.407941	0.366647	0.302546	0.221310	0.136370	0.018897	-0.187912	-0.450938	-0.720000
$t_{40} = 0.40$	0.500000	0.451178	0.400382	0.358310	0.298172	0.222539	0.142402	0.025230	-0.182161	-0.438104	-0.700000
$t_{41} = 0.41$	0.500000	0.447818	0.392911	0.349793	0.293644	0.223723	0.148063	0.030660	-0.176894	-0.425330	-0.680000
$t_{42} = 0.42$	0.500000	0.444430	0.385557	0.341146	0.288977	0.224849	0.153307	0.035189	-0.172037	-0.412618	-0.660000
$t_{43} = 0.43$	0.500000	0.441009	0.378349	0.332421	0.284191	0.225901	0.158086	0.038827	-0.167513	-0.399974	-0.640000
$t_{44} = 0.44$	0.500000	0.437552	0.371307	0.323673	0.279304	0.226858	0.162349	0.041594	-0.163251	-0.387406	-0.620000
$t_{45} = 0.45$	0.500000	0.434057	0.364452	0.314957	0.274337	0.227695	0.166051	0.043521	-0.159181	-0.374922	-0.600000
$t_{46} = 0.46$	0.500000	0.430525	0.357798	0.306330	0.269309	0.228381	0.169143	0.044645	-0.155242	-0.362531	-0.580000
$t_{47} = 0.47$	0.500000	0.426960	0.351356	0.297848	0.264243	0.228885	0.171583	0.045016	-0.151377	-0.350243	-0.560000
$t_{48} = 0.48$	0.500000	0.423370	0.345135	0.289565	0.259158	0.229168	0.173330	0.044688	-0.147536	-0.338063	-0.540000
$t_{49} = 0.49$	0.500000	0.419764	0.339141	0.281533	0.254078	0.229194	0.174349	0.043725	-0.143679	-0.325997	-0.520000
$t_{50} = 0.50$	0.500000	0.416154	0.333378	0.273803	0.249024	0.228920	0.174611	0.042194	-0.139771	-0.314048	-0.500000

Таблица 2. Неявная схема

	$x_0 = 0$	$x_1 = 0.1$	$x_2 = 0.2$	$x_3 = 0.3$	$x_4 = 0.4$	$x_5 = 0.5$	$x_6 = 0.6$	$x_7 = 0.7$	$x_8 = 0.8$	$x_9 = 0.9$	$x_{10} = 1$
$t_0 = 0.00$	0.500000	0.485039	0.436869	0.346793	0.203951	0.000000	-0.265755	-0.581907	-0.922279	-1.245884	-1.500000
$t_1 = 0.01$	0.500000	0.491439	0.444969	0.356793	0.216051	0.014400	-0.248855	-0.562307	-0.899779	-1.220284	-1.480000
$t_2 = 0.02$	0.500000	0.497369	0.452653	0.366270	0.227545	0.028187	-0.232453	-0.542944	-0.877108	-1.194172	-1.460000
$t_3 = 0.03$	0.500000	0.502793	0.459918	0.375225	0.238436	0.041366	-0.216545	-0.523813	-0.854267	-1.167649	-1.440000
$t_4 = 0.04$	0.500000	0.507683	0.466761	0.383663	0.248728	0.053942	-0.201124	-0.504907	-0.831257	-1.140822	-1.420000
$t_5 = 0.05$	0.500000	0.512017	0.473177	0.391584	0.258427	0.065922	-0.186183	-0.486220	-0.808086	-1.113801	-1.400000
$t_6 = 0.06$	0.500000	0.515779	0.479157	0.398993	0.267538	0.077314	-0.171712	-0.467742	-0.784764	-1.086693	-1.380000
$t_7 = 0.07$	0.500000	0.518962	0.484693	0.405891	0.276068	0.088127	-0.157700	-0.449465	-0.761307	-1.059606	-1.360000
$t_8 = 0.08$	0.500000	0.521564	0.489771	0.412282	0.284024	0.098371	-0.144135	-0.431380	-0.737735	-1.032643	-1.340000
$t_9 = 0.09$	0.500000	0.523592	0.494379	0.418168	0.291414	0.108059	-0.131004	-0.413475	-0.714075	-1.005904	-1.320000
$t_{10} = 0.10$	0.500000	0.525057	0.498503	0.423549	0.298247	0.117202	-0.118293	-0.395743	-0.690360	-0.979478	-1.300000
$t_{11} = 0.11$	0.500000	0.525977	0.502129	0.428429	0.304531	0.125814	-0.105986	-0.378173	-0.666629	-0.953449	-1.280000
$t_{12} = 0.12$	0.500000	0.526375	0.505242	0.432808	0.310277	0.133910	-0.094066	-0.360757	-0.642926	-0.927892	-1.260000
$t_{13} = 0.13$	0.500000	0.526280	0.507827	0.436687	0.315495	0.141506	-0.082516	-0.343491	-0.619301	-0.902871	-1.240000
$t_{14} = 0.14$	0.500000	0.525723	0.509874	0.440064	0.320196	0.148618	-0.071316	-0.326368	-0.595807	-0.878439	-1.220000
$t_{15} = 0.15$	0.500000	0.524738	0.511369	0.442941	0.324392	0.155264	-0.060449	-0.309386	-0.572504	-0.854639	-1.200000
$t_{16} = 0.16$	0.500000	0.523362	0.512305	0.445315	0.328093	0.161463	-0.049895	-0.292548	-0.549452	-0.831503	-1.180000
$t_{17} = 0.17$	0.500000	0.521633	0.512676	0.447186	0.331312	0.167234	-0.039635	-0.275856	-0.526715	-0.809054	-1.160000
$t_{18} = 0.18$	0.500000	0.519590	0.512479	0.448551	0.334061	0.172598	-0.029648	-0.259317	-0.504357	-0.787302	-1.140000
$t_{19} = 0.19$	0.500000	0.517272	0.511714	0.449408	0.336353	0.177574	-0.019917	-0.242943	-0.482442	-0.766250	-1.120000
$t_{20} = 0.20$	0.500000	0.514715	0.510387	0.449756	0.338201	0.182184	-0.010423	-0.226749	-0.461033	-0.745890	-1.100000
$t_{21} = 0.21$	0.500000	0.511957	0.508504	0.449593	0.339616	0.186450	-0.001150	-0.210753	-0.440190	-0.726208	-1.080000
$t_{22} = 0.22$	0.500000	0.509030	0.506080	0.448919	0.340613	0.190393	0.007920	-0.194978	-0.419968	-0.707182	-1.060000
$t_{23} = 0.23$	0.500000	0.505968	0.503130	0.447734	0.341203	0.194036	0.016799	-0.179450	-0.400421	-0.688784	-1.040000
$t_{24} = 0.24$	0.500000	0.502797	0.499675	0.446039	0.341399	0.197400	0.025501	-0.164200	-0.381594	-0.670983	-1.020000
$t_{25} = 0.25$	0.500000	0.499543	0.495739	0.443836	0.341215	0.200506	0.034034	-0.149259	-0.363526	-0.653742	-1.000000
$t_{26} = 0.26$	0.500000	0.496230	0.491349	0.441131	0.340662	0.203375	0.042406	-0.134663	-0.346250	-0.637023	-0.980000
$t_{27} = 0.27$	0.500000	0.492874	0.486537	0.437930	0.339754	0.206028	0.050622	-0.120449	-0.329791	-0.620787	-0.960000
$t_{28} = 0.28$	0.500000	0.489492	0.481335	0.434243	0.338504	0.208482	0.058682	-0.106658	-0.314165	-0.604992	-0.940000
$t_{29} = 0.29$	0.500000	0.486095	0.475780	0.430081	0.336923	0.210757	0.066585	-0.093328	-0.299380	-0.589599	-0.920000
$t_{30} = 0.30$	0.500000	0.482693	0.469908	0.425459	0.335024	0.212868	0.074325	-0.080499	-0.285437	-0.574569	-0.900000
$t_{31} = 0.31$	0.500000	0.479291	0.463758	0.420396	0.332822	0.214829	0.081893	-0.068210	-0.272329	-0.559865	-0.880000
$t_{32} = 0.32$	0.500000	0.475894	0.457368	0.414910	0.330329	0.216653	0.089277	-0.056499	-0.260039	-0.545452	-0.860000
$t_{33} = 0.33$	0.500000	0.472503	0.450778	0.409028	0.327558	0.218351	0.096462	-0.045401	-0.248546	-0.531299	-0.840000
$t_{34} = 0.34$	0.500000	0.469118	0.444027	0.402776	0.324524	0.219929	0.103428	-0.034947	-0.237819	-0.517376	-0.820000
$t_{35} = 0.35$	0.500000	0.465737	0.437151	0.396184	0.321241	0.221394	0.110153	-0.025167	-0.227824	-0.503659	-0.800000
$t_{36} = 0.36$	0.500000	0.462358	0.430189	0.389285	0.317723	0.222746	0.116612	-0.016085	-0.218521	-0.490124	-0.780000
$t_{37} = 0.37$	0.500000	0.458979	0.423174	0.382116	0.313987	0.223987	0.122779	-0.007718	-0.209865	-0.476753	-0.760000
$t_{38} = 0.38$	0.500000	0.455595	0.416139	0.374714	0.310049	0.225113	0.128623	-0.000083	-0.201809	-0.463529	-0.740000
$t_{39} = 0.39$	0.500000	0.452205	0.409115	0.367120	0.305924	0.226116	0.134114	0.006815	-0.194303	-0.450440	-0.720000
$t_{40} = 0.40$	0.500000	0.448806	0.402130	0.359377	0.301630	0.226988	0.139221	0.012973	-0.187296	-0.437474	-0.700000
$t_{41} = 0.41$	0.500000	0.445397	0.395211	0.351526	0.297185	0.227717	0.143910	0.018394	-0.180737	-0.424623	-0.680000
$t_{42} = 0.42$	0.500000	0.441977	0.388379	0.343614	0.292607	0.228287	0.148151	0.023089	-0.174574	-0.411880	-0.660000
$t_{43} = 0.43$	0.500000	0.438548	0.381657	0.335683	0.287914	0.228682	0.151910	0.027075	-0.168758	-0.399240	-0.640000
$t_{44} = 0.44$	0.500000	0.435111	0.375063	0.327779	0.283125	0.228883	0.155159	0.030372	-0.163240	-0.386698	-0.620000
$t_{45} = 0.45$	0.500000	0.431672	0.368612	0.319944	0.278259	0.228867	0.157869	0.033008	-0.157976	-0.374251	-0.600000
$t_{46} = 0.46$	0.500000	0.428236	0.362320	0.312222	0.273335	0.228613	0.160016	0.035014	-0.152921	-0.361896	-0.580000
$t_{47} = 0.47$	0.500000	0.424808	0.356198	0.304652	0.268371	0.228096	0.161576	0.036428	-0.148038	-0.349628	-0.560000
$t_{48} = 0.48$	0.500000	0.421399	0.350258	0.297273	0.263385	0.227292	0.162531	0.037288	-0.143290	-0.337444	-0.540000
$t_{49} = 0.49$	0.500000	0.418017	0.344509	0.290121	0.258394	0.226178	0.162867	0.037637	-0.138647	-0.325340	-0.520000
$t_{50} = 0.50$	0.500000	0.414674	0.338960	0.283228	0.253414	0.224728	0.162574	0.037521	-0.134080	-0.313310	-0.500000

## 1.6. Выводы

Был исследован метод конечных разностей. Метод достаточно прост в реализации. Была получена поверхность с распределением  $U(x, t)$  по явной и неявной схеме. Проведено графическое сравнение схем.