

# Метод сеток для решения уравнения параболического типа.

## Вариант 11.

### Содержание

Введение.....	1
Описание алгоритмов.....	1
Подготовительная часть.....	1
Явная схема.....	2
Неявная схема.....	2
Тестирование алгоритма на решении, для которого разностная схема точно аппроксимирует дифференциальную задачу.....	3
Решение.....	3
Явная схема.....	4
Неявная схема.....	5
Тестирование на решении .....	8
Явная схема.....	9
Неявная схема.....	10
Вспомогательные функции.....	13
Разностные схемы.....	13
PrintMethodTable.....	14
Прочее.....	15

### Введение

Необходимо решить задачу

$$\frac{\partial u}{\partial t} = \cos(x) \frac{\partial^2 u}{\partial x^2} + f(x, t)$$

$$u(x, 0) = \phi(x), \quad 0 \leq x \leq 1$$

$$u(0, t) = \alpha(t), \quad \left. \frac{\partial u}{\partial x} \right|_{x=1} = \beta(t), \quad 0 \leq t \leq 0.1$$

используя явную разностную схему и схему с весами.

### Описание алгоритмов

#### Подготовительная часть

Обе схемы подразумевают поиск решения на сетке  $(x_i, t_k)$ , где  $x_i = ih$ ,  $h = \frac{1}{N}$ ,  $t_k = k\tau$ ,  $\tau = \frac{T}{M}$ . В нашем случае,  $T = 0.1$ .

Также заменим наш дифференциальный оператор  $Lu = \cos(x) \frac{\partial^2 u}{\partial x^2}$  на разностный оператор

$$L_h u_i^k = \cos(x_i) \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{h^2}$$

в точке  $(x_i, t_k)$ .

## Явная схема

Аппроксимируем исходное уравнение в узле  $(x_i, t_{k-1})$ , получим

$$\frac{u_i^k - u_i^{k-1}}{\tau} = \cos(x_i) \frac{u_{i+1}^{k-1} - 2u_i^{k-1} + u_{i-1}^{k-1}}{h^2} + f(x_i, t_{k-1})$$

Из начального условия имеем  $u_i^0 = \phi(x_i), i = 0 \dots N$ .

Аппроксимируем граничные условия, получим:

$$u_0^k = \alpha(t_k)$$

$$\frac{3u_N^k - 4u_{N-1}^k + u_{N-2}^k}{2h} = \beta(t_k)$$

Теперь порядок вычисления решения очевиден:

1. Находим  $u_i^0$  по формуле  $u_i^0 = \phi(x_i) \quad i = 0 \dots N$
2. Полагаем  $k = 1$
3. Находим  $u_i^k$  из  $u_i^k = u_i^{k-1} + \tau(L_h u_i^{k-1} + f(x_i, t_{k-1}))$ ,  $i = 1 \dots N - 1$
4. Находим  $u_0^k = \alpha(t_k)$
5. Находим  $u_N^k$  по формуле  $u_N^k = \frac{2h\beta(t_k) + 4u_{N-1}^k - u_{N-2}^k}{3}$
6. Повторяем шаги 3-5 для всех оставшихся значений  $k$ .

Реализация данного алгоритма описана в функции [explicitGridMethod](#).

## Неявная схема

Добавим параметр  $\sigma$  и рассмотрим семейство разностных систем

$$\frac{u_i^k - u_i^{k-1}}{\tau} = L_h(\sigma u_i^k + (1 - \sigma)u_i^{k-1}) + f(x_i, t_k), \quad i = 1 \dots N, \quad k = 1, \dots, M$$

Из начального условия имеем  $u_i^0 = \phi(x_i), i = 0 \dots N$ .

Учитывая тот факт, что к моменту определения решения на  $k$ -м слое решение на  $(k-1)$ -м слое уже известно, систему можно переписать следующим образом:

$$\sigma L_h u_i^k - \frac{1}{\tau} u_i^k = G_i^k$$

где  $G_i^k = -\frac{1}{\tau} u_i^{k-1} - (1 - \sigma)L_h u_i^{k-1} - f(x_i, t_k)$ .

Граничные условия приведем к виду

$$-B_0 u_0^k + C_0 u_1^k = G_0^k$$

$$A_N u_{N-1}^k - B_N u_N^k = G_N^k$$

Таким образом, мы получаем систему с трехдиагональной матрицей

$$A_i = \frac{\sigma \cos x_i}{h^2}$$

$$B_i = \frac{2\sigma \cos x_i}{h^2}$$

$$C_i = \frac{\sigma \cos x_i}{h^2}$$

которую можно решить методом прогонки.

Получаем следующий алгоритм решения:

1. Находим  $u_0$  из  $u_i^0 = \phi(x_i), i = 0 \dots N$
2. Полагаем  $k = 1$
3. Решаем систему при текущем значении  $k$
4. Повторяем шаг 3 для всех оставшихся значений  $k$

Реализация данного алгоритма описана в функции [paramGridMethod](#).

**Примечание:** для решения системы методом прогонки используется написанная ранее функция *TMA-algorithm.m*

## Тестирование алгоритма на решении, для которого разностная схема точно аппроксимирует дифференциальную задачу

### Решение

Рассмотрим функцию  $u(x, t) \equiv 1$ . Сделаем следующие наблюдения:

- $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \equiv 0 \Rightarrow f(x, t) \equiv 0$ .
- $u(x, 0) = \phi(x) \Rightarrow \phi(x) \equiv 1$ .
- $u(0, t) = \alpha(t) \Rightarrow \alpha(t) \equiv 1$
- $\frac{\partial u}{\partial x} = \beta(t) \Rightarrow \beta(t) \equiv 0$

Введем все необходимые переменные

```
syms x t u_ex alpha_ex beta_ex phi_ex f_ex;
u_ex = 1;
f_ex = 0;
phi_ex = 1;
```

```
alpha_ex = 1;
beta_ex = 0;
```

Построим таблицу значений нашей функции  $u(x, t)$  на крупной сетке

```
exSol = generateSolutionTable(u_ex);
printBigGridTable(exSol);
```

ans = 6×7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1
6	0.1000	1	1	1	1	1	1

## Явная схема

Проверим, что явная схема точно аппроксимирует выбранную нами задачу.

**Зам:** описание метода [printMethodTable](#)

```
printMethodTable(exSol, alpha_ex, beta_ex, phi_ex, f_ex, -1)
```

```
=====
Таблицы решений на крупной сетке при различных h
h = 0.2
```

ans = 6×7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1
6	0.1000	1	1	1	1	1	1

h = 0.1

ans = 6×7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1

	tlx	0	0.2	0.4	0.6	0.8	1
6	0.1000	1	1	1	1	1	1

h = 0.05

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1
6	0.1000	1	1	1	1	1	1

Таблица, характеризующая точность решения

ans = 3x4 table

	h	tau	J_ex - u	u^h - u^(2h)
1	0.2000	0.0200	0	0
2	0.1000	0.0050	0	0
3	0.0500	0.0013	0	0

=====

Как и ожидалось, задача была аппроксимирована точно.

## Неявная схема

Теперь проверим работу неявной схемы при трех различных значениях параметра: 0, 1 и 0.5

```
printMethodTable(exSol,alpha_ex,beta_ex,phi_ex,f_ex,[0, 1, 0.5])
```

Текущее значение sigma = 0

=====

Таблицы решений на крупной сетке при различных h

h = 0.2

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1
6	0.1000	1	1	1	1	1	1

h = 0.1

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1

	$\tau \backslash x$	0	0.2	0.4	0.6	0.8	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1
6	0.1000	1	1	1	1	1	1

$h = 0.05$

ans = 6×7 table

	$\tau \backslash x$	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1
6	0.1000	1	1	1	1	1	1

-----  
Таблица, характеризующая точность решения

ans = 3×4 table

	$h$	$\tau$	$ J_{ex} - u $	$ u^h - u^{(2h)} $
1	0.2000	0.0200	0	0
2	0.1000	0.0050	0	0
3	0.0500	0.0013	0	0

=====

Текущее значение  $\sigma = 1$

=====

Таблицы решений на крупной сетке при различных  $h$

$h = 0.2$

ans = 6×7 table

	$\tau \backslash x$	0	0.2	0.4	0.6	0.8	1
1	0	1	1.0000	1.0000	1	1	1
2	0.0200	1	1.0000	1.0000	1	1	1
3	0.0400	1	1.0000	1.0000	1	1	1
4	0.0600	1	1.0000	1.0000	1	1	1
5	0.0800	1	1.0000	1.0000	1	1	1
6	0.1000	1	1.0000	1.0000	1	1	1

$h = 0.1$

ans = 6×7 table

	$\tau \backslash x$	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1.0000	1	1.0000
2	0.0200	1	1	1	1.0000	1	1.0000
3	0.0400	1	1	1	1.0000	1	1.0000

	tlx	0	0.2	0.4	0.6	0.8	1
4	0.0600	1	1	1	1.0000	1	1.0000
5	0.0800	1	1	1	1.0000	1	1.0000
6	0.1000	1	1	1	1.0000	1	1.0000

h = 0.05

ans = 6×7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	1	1.0000	1	1	1	1
2	0.0200	1	1.0000	1	1	1	1
3	0.0400	1	1.0000	1	1	1	1
4	0.0600	1	1.0000	1	1	1	1
5	0.0800	1	1.0000	1	1	1	1
6	0.1000	1	1.0000	1	1	1	1

-----  
Таблица, характеризующая точность решения

ans = 3×4 table

	h	tau	J_ex - u	u^h - u^(2h)
1	0.2000	0.0200	2.2204e-16	0
2	0.1000	0.0050	2.2204e-16	2.2204e-16
3	0.0500	0.0013	1.1102e-16	2.2204e-16

=====

Текущее значение sigma = 0.5

=====

Таблицы решений на крупной сетке при различных h

h = 0.2

ans = 6×7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	1	1.0000	1	1.0000	1.0000	1.0000
2	0.0200	1	1.0000	1	1.0000	1.0000	1.0000
3	0.0400	1	1.0000	1	1.0000	1.0000	1.0000
4	0.0600	1	1.0000	1	1.0000	1.0000	1.0000
5	0.0800	1	1.0000	1	1.0000	1.0000	1.0000
6	0.1000	1	1.0000	1	1.0000	1.0000	1.0000

h = 0.1

ans = 6×7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	1	1.0000	1	1	1	1.0000
2	0.0200	1	1.0000	1	1	1	1.0000
3	0.0400	1	1.0000	1	1	1	1.0000
4	0.0600	1	1.0000	1	1	1	1.0000
5	0.0800	1	1.0000	1	1	1	1.0000

	tlx	0	0.2	0.4	0.6	0.8	1
6	0.1000	1	1.0000	1	1	1	1.0000

h = 0.05

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	1	1	1	1	1	1
2	0.0200	1	1	1	1	1	1
3	0.0400	1	1	1	1	1	1
4	0.0600	1	1	1	1	1	1
5	0.0800	1	1	1	1	1	1
6	0.1000	1	1	1	1	1	1

Таблица, характеризующая точность решения

ans = 3x4 table

	h	tau	J_ex - u	u^h - u^(2h)
1	0.2000	0.0200	2.2204e-16	0
2	0.1000	0.0050	2.2204e-16	4.4409e-16
3	0.0500	0.0013	0	2.2204e-16

Легко заметить, что при правильном подборе параметра и шага неявная схема точно аппроксимирует задачу.

## Тестирование на решении $u(x, t) = x^3 t^3$

Теперь рассмотрим функцию  $u(x, t) = x^3 t^3$ . Сделаем следующие наблюдения:

- $\frac{\partial u}{\partial t} = 3x^3 t^2$ ,  $\frac{\partial^2 u}{\partial x^2} = 6xt^3 \Rightarrow f(x, t) = 3x^3 t^2 - 6xt^3 \cos x$ .
- $u(x, 0) = \phi(x) \Rightarrow \phi(x) \equiv 0$ .
- $u(0, t) = \alpha(t) \Rightarrow \alpha(t) \equiv 0$
- $\left. \frac{\partial u}{\partial x} \right|_{x=1} = \beta(t) \Rightarrow \beta(t) = 3t^3$

Введем все необходимые переменные

```
syms x t u_appr alpha_appr beta_appr phi_appr f_appr;
u_appr = x^3 * t^3;
f_appr = 3*t^2*x^3 - 6*t^3*x*cos(x);
phi_appr = 0;
alpha_appr = 0;
beta_appr = 3*t^3;
```

Построим таблицу значений функции на крупной сетке



```
apprSol = generateSolutionTable(u_appr);
printBigGridTable(apprSol);
```

ans = 6×7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	6.4000e-08	5.1200e-07	1.7280e-06	4.0960e-06	8.0000e-06
3	0.0400	0	5.1200e-07	4.0960e-06	1.3824e-05	3.2768e-05	6.4000e-05
4	0.0600	0	1.7280e-06	1.3824e-05	4.6656e-05	1.1059e-04	2.1600e-04
5	0.0800	0	4.0960e-06	3.2768e-05	1.1059e-04	2.6214e-04	5.1200e-04
6	0.1000	0	8.0000e-06	6.4000e-05	2.1600e-04	5.1200e-04	1.0000e-03

## Явная схема

Посмотрим, насколько точно явная схема аппроксимирует данную задачу

**Зам:** описание метода [printMethodTable](#)

```
printMethodTable(apprSol,alpha_appr,beta_appr,phi_appr,f_appr,-1)
```

=====

Таблицы решений на крупной сетке при различных h

h = 0.2

ans = 6×7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	0	0	0	0	3.2000e-06
3	0.0400	0	3.8272e-09	1.1823e-06	4.7086e-06	1.2868e-05	4.1187e-05
4	0.0600	0	-1.5793e-07	5.5781e-06	2.3553e-05	6.4762e-05	1.6490e-04
5	0.0800	0	-6.2238e-07	1.5489e-05	6.6961e-05	1.8143e-04	4.2439e-04
6	0.1000	0	-1.3933e-06	3.3714e-05	1.4548e-04	3.8856e-04	8.6958e-04

h = 0.1

ans = 6×7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	2.8845e-08	3.1127e-07	1.1007e-06	2.6671e-06	5.9375e-06
3	0.0400	0	2.7877e-07	3.0950e-06	1.1005e-05	2.6850e-05	5.6018e-05
4	0.0600	0	1.0036e-06	1.1195e-05	3.9899e-05	9.7355e-05	1.9894e-04
5	0.0800	0	2.4767e-06	2.7525e-05	9.8099e-05	2.3901e-04	4.8324e-04
6	0.1000	0	5.0033e-06	5.5065e-05	1.9599e-04	4.7664e-04	9.5740e-04

h = 0.05

ans = 6×7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	5.3020e-08	4.5540e-07	1.5565e-06	3.7105e-06	7.4114e-06
3	0.0400	0	4.4528e-07	3.8255e-06	1.3081e-05	3.1215e-05	6.1802e-05
4	0.0600	0	1.5289e-06	1.3127e-05	4.4894e-05	1.0714e-04	2.1135e-04
5	0.0800	0	3.6609e-06	3.1391e-05	1.0735e-04	2.5611e-04	5.0420e-04
6	0.1000	0	7.2055e-06	6.1667e-05	2.1081e-04	5.0278e-04	9.8845e-04

-----  
Таблица, характеризующая точность решения

ans = 3x4 table

	h	tau	J_ex - u	u^h - u^(2h)
1	0.2000	0.0200	1.3042e-04	0
2	0.1000	0.0050	4.2603e-05	8.8079e-05
3	0.0500	0.0013	1.1545e-05	3.1058e-05

=====

## Неявная схема

Теперь оценим результаты применения неявной схемы

```
printMethodTable(apprSol,alpha_appr,beta_appr,phi_appr,f_appr,[0,1,0.5])
```

Текущее значение sigma = 0

=====

Таблицы решений на крупной сетке при различных h

h = 0.2

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	0	0	0	0	0.0000
3	0.0400	0	3.8272e-09	1.1823e-06	4.7086e-06	1.3425e-05	0.0001
4	0.0600	0	-1.5793e-07	5.5781e-06	2.3783e-05	6.8637e-05	0.0002
5	0.0800	0	-6.2238e-07	1.5595e-05	6.8601e-05	1.9430e-04	0.0005
6	0.1000	0	-1.3414e-06	3.4477e-05	1.5112e-04	4.1989e-04	0.0010

h = 0.1

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	2.8845e-08	3.1127e-07	1.1007e-06	2.6761e-06	0.0000
3	0.0400	0	2.7877e-07	3.0953e-06	1.1019e-05	2.7109e-05	0.0001
4	0.0600	0	1.0044e-06	1.1209e-05	4.0065e-05	9.8820e-05	0.0002
5	0.0800	0	2.4895e-06	2.7636e-05	9.8883e-05	2.4381e-04	0.0005
6	0.1000	0	5.0783e-06	5.5526e-05	1.9844e-04	4.8851e-04	0.0010

h = 0.05

ans = 6x7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	5.3020e-08	4.5540e-07	1.5566e-06	3.7138e-06	0.0000
3	0.0400	0	4.4529e-07	3.8258e-06	1.3087e-05	3.1303e-05	0.0001
4	0.0600	0	1.5294e-06	1.3133e-05	4.4958e-05	1.0767e-04	0.0002
5	0.0800	0	3.6671e-06	3.1438e-05	1.0766e-04	2.5797e-04	0.0005
6	0.1000	0	7.2391e-06	6.1859e-05	2.1180e-04	5.0756e-04	0.0010

Таблица, характеризующая точность решения

ans = 3x4 table

	h	tau	J_ex - u	u^h - u^(2h)
1	0.2000	0.0200	9.2108e-05	0
2	0.1000	0.0050	2.3487e-05	6.8621e-05
3	0.0500	0.0013	1.1246e-05	1.9047e-05

Текущее значение sigma = 1

Таблицы решений на крупной сетке при различных h

h = 0.2

ans = 6x7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	5.1919e-07	2.0901e-06	5.6320e-06	1.1412e-05	0.0000
3	0.0400	0	2.3862e-06	1.0087e-05	2.7955e-05	5.8885e-05	0.0001
4	0.0600	0	6.3835e-06	2.7766e-05	7.8255e-05	1.6868e-04	0.0003
5	0.0800	0	1.3291e-05	5.8985e-05	1.6825e-04	3.6835e-04	0.0007
6	0.1000	0	2.3932e-05	1.0778e-04	3.1021e-04	6.8657e-04	0.0013

h = 0.1

ans = 6x7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	1.2576e-07	7.8647e-07	2.5114e-06	5.7695e-06	0.0000
3	0.0400	0	8.3893e-07	5.3128e-06	1.7017e-05	3.9312e-05	0.0001
4	0.0600	0	2.6441e-06	1.6867e-05	5.4165e-05	1.2575e-04	0.0002
5	0.0800	0	6.0386e-06	3.8747e-05	1.2474e-04	2.9075e-04	0.0006
6	0.1000	0	1.1529e-05	7.4312e-05	2.3976e-04	5.6049e-04	0.0011

h = 0.05

ans = 6x7 table

	tx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0

	tlx	0	0.2	0.4	0.6	0.8	1
2	0.0200	0	7.6650e-08	5.7326e-07	1.9095e-06	4.4993e-06	0.0000
3	0.0400	0	5.8467e-07	4.3805e-06	1.4596e-05	3.4423e-05	0.0001
4	0.0600	0	1.9399e-06	1.4553e-05	4.8517e-05	1.1459e-04	0.0002
5	0.0800	0	4.5575e-06	3.4232e-05	1.1420e-04	2.7005e-04	0.0005
6	0.1000	0	8.8588e-06	6.6588e-05	2.2227e-04	5.2611e-04	0.0010

-----  
Таблица, характеризующая точность решения

ans = 3x4 table

	h	tau	J_ex - u	u^h - u^(2h)
1	0.2000	0.0200	2.8657e-04	0
2	0.1000	0.0050	9.7147e-05	1.8942e-04
3	0.0500	0.0013	3.4409e-05	6.2738e-05

=====

Текущее значение sigma = 0.5

=====

Таблицы решений на крупной сетке при различных h

h = 0.2

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	9.4846e-08	4.7689e-07	1.4543e-06	3.4871e-06	0.0000
3	0.0400	0	6.7267e-07	4.2453e-06	1.3763e-05	3.3127e-05	0.0001
4	0.0600	0	2.1479e-06	1.4519e-05	4.7870e-05	1.1565e-04	0.0002
5	0.0800	0	4.9937e-06	3.4708e-05	1.1526e-04	2.7899e-04	0.0006
6	0.1000	0	9.7818e-06	6.8501e-05	2.2803e-04	5.5211e-04	0.0012

h = 0.1

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	6.5884e-08	5.0947e-07	1.7092e-06	4.0579e-06	0.0000
3	0.0400	0	5.2154e-07	4.1039e-06	1.3827e-05	3.2958e-05	0.0001
4	0.0600	0	1.7538e-06	1.3877e-05	4.6859e-05	1.1200e-04	0.0002
5	0.0800	0	4.1633e-06	3.2986e-05	1.1153e-04	2.6701e-04	0.0005
6	0.1000	0	8.1828e-06	6.4691e-05	2.1882e-04	5.2427e-04	0.0011

h = 0.05

ans = 6x7 table

	tlx	0	0.2	0.4	0.6	0.8	1
1	0	0	0	0	0	0	0
2	0.0200	0	6.4118e-08	5.1184e-07	1.7268e-06	4.0955e-06	0.0000
3	0.0400	0	5.1260e-07	4.0967e-06	1.3828e-05	3.2846e-05	0.0001

	tlx	0	0.2	0.4	0.6	0.8	1
4	0.0600	0	1.7301e-06	1.3833e-05	4.6720e-05	1.1111e-04	0.0002
5	0.0800	0	4.1056e-06	3.2821e-05	1.1091e-04	2.6399e-04	0.0005
6	0.1000	0	8.0407e-06	6.4208e-05	2.1701e-04	5.1682e-04	0.0010

-----  
Таблица, характеризующая точность решения

ans = 3x4 table

	h	tau	J_ex - u	u^h - u^(2h)
1	0.2000	0.0200	1.5211e-04	0
2	0.1000	0.0050	5.4862e-05	9.7251e-05
3	0.0500	0.0013	2.2812e-05	3.2050e-05

=====

Легко заметить, что обе схемы аппроксимируют данную задачу с примерно одинаковой точностью.

## Вспомогательные функции

### Разностные схемы

Данная функция является реализацией явной разностной схемы.

```
function result = explicitGridMethod(f,phi,alpha,beta,N,M,T)
    syms x t;
    u = zeros(M + 1, N + 1);
    tau = T / M;
    h = 1 / N;
    for i = 1: N + 1
        u(1, i) = subs(phi,x,((i - 1) * h));
    end

    for k = 2: M + 1
        tk_1 = (k - 2) * tau;
        for i = 2: N
            xi = (i - 1) * h;
            u(k, i) = u(k - 1, i) + tau * (cos(xi)*(u(k-1,i+1) ...
                -2*u(k-1,i)+u(k-1,i-1))/(h^2) + subs(f,[x,t],[xi,tk_1]));
        end
        tk = tk_1 + tau;
        u(k, 1) = subs(alpha,t,tk);
        u(k, N + 1) = (2 * h * subs(beta,t,tk) + 4 * u(k, N) - u(k, N - 1)) / 3;
    end
    result = u(1:M/5:end,1:N/5:end);
end
```

Данная функция является реализацией неявной разностной схемы.

```
function result = paramGridMethod(f, phi, alpha, beta, N, M, T, sigma)
    syms x t;
    A = zeros(1, N + 1);
```

```

B = zeros(1, N + 1);
C = zeros(1, N + 1);
G = zeros(1, N + 1);
u = zeros(M + 1, N + 1);
tau = T / M;
h = 1 / N;
for i = 1: N + 1
    u(1, i) = subs(phi,x,((i - 1) * h));
end

cr = (sigma - 1) * tau;
for k = 2: M + 1
    tk = (k - 1) * tau;
    t_k = tk + cr;
    for i = 2: N
        xi = (i - 1) * h;
        A(i) = sigma * cos(xi) / (h ^ 2);
        B(i) = 1 / tau + 2 * sigma * cos(xi) / (h ^ 2);
        C(i) = sigma * cos(xi)/(h^2);
        G(i) = -u(k - 1, i) / tau - (1 - sigma) * (cos(xi)*(u(k - 1, i + 1) ...
            - 2 * u(k - 1, i) + u(k - 1, i - 1)) / (h ^ 2)) ...
            - subs(f,[x,t],[xi, t_k]);
    end
    A(1) = 0;
    B(1) = -1;
    C(1) = 0;
    G(1) = subs(alpha,t,tk);
    A(N + 1) = -1/h;
    B(N + 1) = -1/h;
    C(N + 1) = 0;
    G(N + 1) = subs(beta,t,tk);

    [Y, ~, ~] = TMA_algorithm(A, B, C, G);
    for i = 1: N + 1
        u(k, i) = Y(i);
    end
end
result = u(1:M/5:end,1:N/5:end);
end

```

## PrintMethodTable

Следующий метод - основной среди всех вспомогательных инструментов. В зависимости от значения последнего параметра (sigma), данный метод вызывает одну из описанных выше реализаций разностных схем и печатает всю необходимую информацию о полученной аппроксимации: таблицу значений на крупной сетке для каждого из заданных значений h и таблицу, характеризующую точность решения и внутреннюю сходимость.

**Примечание:** при  $\sigma = -1$  будем вызван метод [explicitGridMethod](#). При прочих значениях sigma вызывается [paramGridMethod](#).

```

function printMethodTable(u_ex,alpha,beta,phi,f,sigma)
    h = [0.2, 0.1, 0.05];
    tau = [0.02, 0.005, 0.00125];
    T = 0.1;
    errors = zeros(3,1);
    conv = zeros(3,1);
    u_prev = zeros(5,5);
    u_curr = zeros(5,5);

    for sigma=sigma
        if sigma > -1
            disp("Текущее значение sigma = "+sigma);
        end
        disp("=====");
        disp("Таблицы решений на крупной сетке при различных h");
        for i=1:3
            M = T/tau(i);
            N = 1/h(i);
            u_prev = u_curr;
            if sigma > -1
                u_curr = paramGridMethod(f,phi,alpha,beta,N,M,T,sigma);
            else
                u_curr = explicitGridMethod(f,phi,alpha,beta,N,M,T);
            end

            disp("h = "+h(i));
            printBigGridTable(u_curr);
            errors(i) = max(max(abs(u_ex-u_curr)));
            if i>1
                conv(i) = max(max(abs(u_curr-u_prev)));
            end
        end

        variableNames = {'h','tau','|J_ex - u|','|u^h - u^(2h)|'};
        disp("-----");
        disp("Таблица, характеризующая точность решения")
        table(transpose(h),transpose(tau),errors,conv, ...
            'VariableNames',variableNames)
        disp("=====");
    end
end

```

## Прочее

Данная функция генерирует таблицу значений функции-аргумента на крупной сетке

```

function result = generateSolutionTable(u)
    syms x t;
    x_vals = 0:0.2:1;
    t_vals = 0:0.02:0.1;
    values = zeros(size(x_vals,2),size(x_vals,2));
    for i=1:size(x_vals,2)
        for j = 1:size(t_vals,2)
            values(i,j) = subs(u,[x, t],[x_vals(1,i),t_vals(1,j)]);
        end
    end
end

```

```
        end
    end
    result = values;
end
```

Данная функция отвечает за печать таблицы значений функции на крупной сетке

```
function printBigGridTable(u)
    t_vals = 0:0.02:0.1;
    variableNames = {'t\x', '0', '0.2', '0.4', '0.6', '0.8', '1'};
    table(transpose(t_vals), u(:,1), u(:,2), u(:,3), ...
        u(:,4), u(:,5), u(:,6), 'VariableNames', variableNames)
end
```