

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное
учреждение высшего образования «Самарский национальный
исследовательский университет имени академика С.П.Королёва»
ИНСТИТУТ ИНФОРМАТИКИ, МАТЕМАТИКИ И ЭЛЕКТРОНИКИ
ФАКУЛЬТЕТ ИНФОРМАТИКИ
*КАФЕДРА ТЕХНИЧЕСКОЙ КИБЕРНЕТИКИ***

Расчётно-пояснительная записка к курсовой работе
по дисциплине «Численные методы математической физики»

**Тема: «РЕШЕНИЕ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ
МЕТОДОМ КОНЕЧНЫХ РАЗНОСТЕЙ»**

Вариант № 7

Выполнили: Логинов П.В.

Лаптев А.В.

Группа: 6409

Проверил: Дегтярёв А.А.

Оценка: _____

Дата: _____

Самара 2018

ЗАДАНИЕ

1. Осуществить математическую постановку краевой задачи для физического процесса, описанного в предложенном варианте курсовой работы.
2. Осуществить построение разностной схемы, приближающей полученную краевую задачу. При этом следует согласовать с преподавателем тип разностной схемы.
3. Провести теоретическое исследование схемы: показать, что схема аппроксимирует исходную краевую задачу, и найти порядки аппроксимации относительно шагов дискретизации; исследовать устойчивость схемы и сходимость сеточного решения к решению исходной задачи математической физики.
4. Разработать алгоритм численного решения разностной краевой задачи.
5. Разработать компьютерную программу, реализующую созданный алгоритм, с интерфейсом, обеспечивающим следующие возможности: диалоговый режим ввода физических, геометрических и сеточных параметров задачи; графическую визуализацию численного решения задачи.
6. Используя разработанную программу и тестовый пример, согласованный с преподавателем, провести экспериментальное исследование фактической сходимости сеточного решения к точному (вычисленному с помощью ряда Фурье). Проводя измельчение сетки, сравнить экспериментальную скорость убывания погрешности сеточного решения со скоростью, полученной при теоретическом исследовании схемы.
7. Оформить отчет о проделанной работе в соответствии с требованиями, изложенными в подразделе 3.2 настоящих методических указаний.

ВАРИАНТ 7

Разработать программу численного моделирования теплового процесса в тонком однородном стержне длиной l , сечением S на временном промежутке $0 < t \leq T$. На боковой поверхности стержня и на одном из его концов происходит теплообмен с окружающей средой, описываемый законом Ньютона с коэффициентом теплообмена. Другой конец стержня теплоизолирован. Начальная температура стержня во всех его точках одинакова и равна температуре окружающей среды u_0 . Коэффициенты теплопроводности и объемной теплоемкости материала стержня равны k и c соответственно. Интенсивность распределенных внутренних источников тепла характеризуется функцией $\varphi(x)$, $0 \leq x \leq l$.

Для численного решения задачи использовать неявные разностные схемы, различающиеся расчётом граничных условий.

Разработать программу расчёта теплового процесса в тонком однородном стержне используя полученное решение.

При проведении расчетов использовать значения параметров а также выражение функции $\varphi(x)$, указанные преподавателем.

$$l = 10 \text{ см},$$

$$S = 4 \text{ мм}^2,$$

$$T = 30 \text{ с},$$

$$\alpha = 0.05,$$

$$k = 0.067,$$

$$c = 1.84,$$

$$\varphi(x) = \sin\left(\frac{\pi x}{l}\right).$$

РЕФЕРАТ

Отчет по курсовой работе: 45 страниц, 4 рисунка, 2 таблицы, 5 источников, 1 приложение.

УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ, КРАЕВАЯ ЗАДАЧА, УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ, МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ, МЕТОД ПРОГОНКИ, МЕТОДИЧЕСКАЯ ПОГРЕШНОСТЬ.

Целью курсовой работы является построение и исследование разностной схемы для решения краевой задачи теплопроводности для тонкого однородного стержня, теплоизолированного с одного из концов.

Для решения задачи использованы две неявные конечно-разностные схемы. Проведено теоретическое исследование аппроксимации и устойчивости разностных схем. Сделан вывод о сходимости сеточных решений к точному решению исходной задачи.

Разработана компьютерная программа, реализующая алгоритм скалярной прогонки и позволяющая рассчитывать решения сеточной задачи.

Приведены графические результаты численного решения задачи теплопроводности.

Программа написана на языке C# 7.0 в среде разработки Microsoft Visual Studio Community 2017, операционная система Windows 10.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ	7
2 ПРОСТЕЙШАЯ НЕЯВНАЯ РАЗНОСТНАЯ СХЕМА	11
2.1 Построение неявной разностной схемы.....	11
2.2 Аппроксимация простейшей разностной схемы.....	13
2.3 Устойчивость разностной схемы.....	15
2.4 Метод прогонки для простейшей неявной разностной схемы	17
2.5 Исследование сходимости к решению исходной задачи	20
3 НЕЯВНАЯ РАЗНОСТНАЯ СХЕМА ПОВЫШЕННОГО ПОРЯДКА.....	22
3.1 Построение неявной разностной схемы.....	22
3.2 Аппроксимация неявной разностной схемы	23
3.3 Устойчивость разностной схемы.....	25
3.4 Метод прогонки для неявной разностной схемы.....	28
3.5 Исследование сходимости к решению исходной задачи	30
4 ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ РЕШЕНИЯ.....	31
5 ИССЛЕДОВАНИЕ ПОГРЕШНОСТИ С ПОМОЩЬЮ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА	34
ЗАКЛЮЧЕНИЕ	36
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	37
<i>ПРИЛОЖЕНИЕ А</i> Исходный код программы	38

ВВЕДЕНИЕ

Характеризуя метод конечных разностей, необходимо выделить его достоинства и недостатки в сравнении с другими методами.

К достоинствам метода конечных разностей следует отнести его высокую универсальность, например, значительно более высокую, чем у аналитических методов. Применение этого метода нередко характеризуется относительной простотой построения решающего алгоритма и его программной реализации. Зачастую удается осуществить распараллеливание решающего алгоритма.

К числу недостатков метода следует отнести: проблематичность его использования на нерегулярных сетках; очень быстрый рост вычислительной трудоемкости при увеличении размерности задачи (увеличении числа неизвестных переменных); сложность аналитического исследования свойств разностной схемы.

Суть метода конечных разностей состоит в замене исходной (непрерывной) задачи математической физики ее дискретным аналогом (разностной схемой), а также последующим применением специальных алгоритмов решения дискретной задачи.

В настоящей работе метод конечных разностей применен для численного решения задачи теплопроводности. Вычислительный алгоритм основан на использовании неявной разностной схемы. Проведено теоретическое исследование аппроксимации и устойчивости разностной схемы. Сделан вывод о сходимости сеточного решения к точному решению исходной задачи. Разработана компьютерная программа, реализующая алгоритм скалярной прогонки. Приведены графические результаты численного решения задачи теплопроводности.

1 МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

Рассмотрим твердое тело, температура которого в точке пространства (x,y,z) в момент времени t определяется функцией $u(x,y,z,t)$.

Будем считать, что коэффициент внутренней теплопроводности k зависит только от точки (x,y,z) тела и не зависит от направления нормали поверхности S в этой точке.

Обозначим через q тепловой поток - количество тепла, проходящего через единицу площади поверхности за единицу времени:

$$q = -k \frac{\partial u}{\partial \vec{n}} \quad (1.1)$$

Для вывода уравнения распространения тепла выделим внутри тела произвольный объем V и рассмотрим изменение количество тепла в этом объеме за промежуток времени $[0, T]$ [1].

$$\frac{\partial u}{\partial t} = \frac{k}{c} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + \varphi(x). \quad (1.2)$$

Найдем среднее распределение тепла по каждой координате:

$$v = \frac{1}{S} \iint_S u dy dz. \quad (1.3)$$

Будем считать, что сечение стержня представляет собой квадрат, поэтому обозначим длину грани следующим образом:

$$b = \sqrt{S}. \quad (1.4)$$

Схематичное представление стержня изображено на рисунке 1.

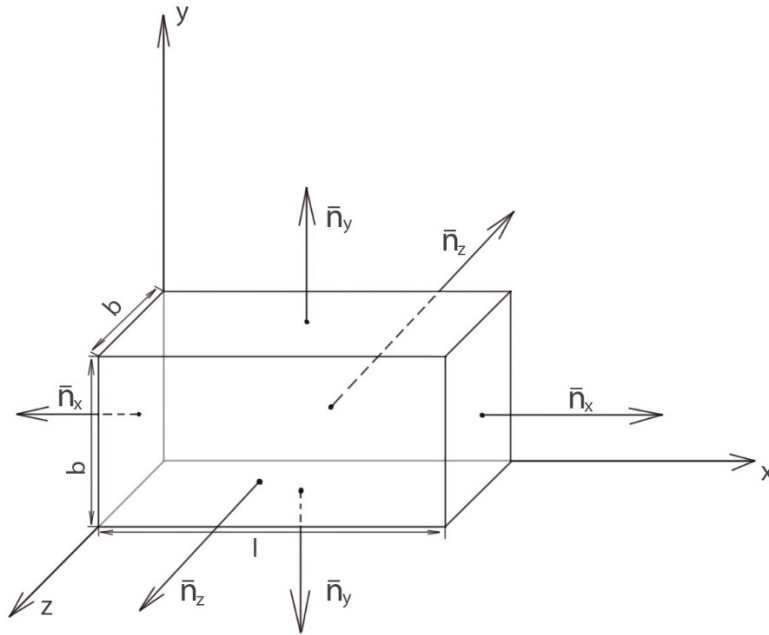


Рисунок 1 – Графическое представление тонкого стержня

Выразим частные производные для подстановки в исходное уравнение:

$$\frac{\partial v}{\partial t} = \frac{1}{S} \iint_S \frac{\partial u}{\partial t} dydz = \frac{\partial}{\partial t} \left[\frac{1}{S} \iint_S u dydz \right]; \quad (1.5)$$

$$\frac{\partial^2 v}{\partial x^2} = \frac{1}{S} \iint_S \frac{\partial^2 u}{\partial x^2} dydz. \quad (1.6)$$

Поскольку теплообмен на боковой поверхности осуществляется по закону Ньютона, то тепловой поток q будет определяться на границах следующим образом [2].

$$\begin{cases} q|_{y=0} = \alpha(u|_{y=0} - u_0); \\ q|_{y=b} = \alpha(u|_{y=b} - u_0). \end{cases} \quad (1.7)$$

Продифференцируем исходную функцию по нормали:

$$\begin{aligned} \frac{\partial u}{\partial \bar{n}} &= \frac{\partial u}{\partial x} \cos(\bar{n}, \bar{e}_x) + \\ &+ \frac{\partial u}{\partial y} \cos(\bar{n}, \bar{e}_y) + \frac{\partial u}{\partial z} \cos(\bar{n}, \bar{e}_z) = -\frac{\partial v}{\partial y}. \end{aligned} \quad (1.8)$$

Выразим значение теплового потока на нуле по оси y :

$$q|_{y=0} = -k \left(-\frac{\partial u}{\partial y} \Big|_{y=0} \right) = k \frac{\partial u}{\partial y} \Big|_{y=0} = \alpha(u|_{y=0} - u_0). \quad (1.9)$$

Из этого следует, что частные производные будут такими:

$$\begin{cases} \frac{\partial u}{\partial y} \Big|_{y=0} = \frac{\alpha}{k}(u|_{y=0} - u_0), \\ \frac{\partial u}{\partial y} \Big|_{y=b} = -\frac{\alpha}{k}(u|_{y=b} - u_0). \end{cases} \quad (1.10)$$

Выразим вторую частную производную по y для подстановки в исходное уравнение:

$$\begin{aligned} \frac{\partial^2 v}{\partial y^2} &= \frac{1}{S} \int_0^b \left[\frac{\partial u}{\partial y} \Big|_{y=b} - \frac{\partial u}{\partial y} \Big|_{y=0} \right] dz = \\ &= \frac{1}{S} \int_0^b \left[-\frac{\alpha}{k}((u|_{y=b} - u_0) - (u|_{y=0} - u_0)) \right] dz = \\ &= -\frac{\alpha}{kS} \int_0^b (u|_{y=b} - u|_{y=0} - 2u_0) dz = \\ &= -\frac{\alpha}{kS} \int_0^b (u|_{y=b} - u|_{y=0}) dz + \frac{2\alpha}{kS} u_0 = \\ &= \frac{2\alpha}{kS} u_0 - \frac{2\alpha}{kS} \int_0^b u dz = \frac{2\alpha}{kS} \left[u_0 - \frac{1}{b} \int_0^b u dz \right]. \end{aligned} \quad (1.11)$$

Аналогично выразим вторую частную производную для z :

$$\frac{\partial^2 v}{\partial z^2} = \frac{2\alpha}{kS} \left[u_0 - \frac{1}{b} \int_0^b u dy \right]. \quad (1.12)$$

$$v = \frac{1}{b} \int_0^b u dy = \frac{1}{b} \int_0^b u dz. \quad (1.13)$$

Подставим результат в формулы вторых частных производных [3].

$$\frac{\partial^2 v}{\partial y^2} = \frac{\partial^2 v}{\partial z^2} = \frac{2\alpha}{kb} [u_0 - v]. \quad (1.14)$$

В итоге получена математическая модель задачи:

$$\left\{ \begin{array}{l} \frac{\partial v}{\partial t} = \frac{k}{c} \left(\frac{\partial^2 v}{\partial x^2} + \frac{4\alpha}{kb} [u_0 - v] \right) + \varphi(x), \quad x \in [0, l], \quad t \in [0, T]; \\ v|_{t=0} = u_0, \quad x \in [0, l]; \\ \frac{\partial v}{\partial x}|_{x=0} = 0, \quad t \in [0, T]; \\ \frac{\partial v}{\partial x}|_{x=l} = -\frac{\alpha}{k} [v|_{x=l} - u_0], \quad t \in [0, T]. \end{array} \right. \quad (1.15)$$

Введём замену $\omega = v - u_0$. Математическая модель после замены выглядит так:

$$\left\{ \begin{array}{l} \frac{\partial \omega}{\partial t} = \frac{k}{c} \left(\frac{\partial^2 \omega}{\partial x^2} - \frac{4\alpha\omega}{kb} \right) + \varphi(x), \quad x \in [0, l], \quad t \in [0, T]; \\ \omega|_{t=0} = 0, \quad x \in [0, l]; \\ \frac{\partial \omega}{\partial x}|_{x=0} = 0, \quad t \in [0, T]; \\ \frac{\partial \omega}{\partial x}|_{x=l} = -\frac{\alpha}{k} \omega|_{x=l}, \quad t \in [0, T]. \end{array} \right. \quad (1.16)$$

Для простоты записи разностных схем, используемых в дальнейшем для численного решения задачи (1.16) примем $u = \omega$.

2 ПРОСТЕЙШАЯ НЕЯВНАЯ РАЗНОСТНАЯ СХЕМА

2.1 Построение неявной разностной схемы

Сетка задается узловыми точками (x_i, t_k) , где

$$x_i = ih_x, \quad i = \overline{0, I}, \quad h_x = \frac{l}{I},$$
$$t_k = kh_t, \quad k = \overline{0, K}, \quad h_t = \frac{T}{K}.$$

Аппроксимируем на этой сетке производные разностными соотношениями:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2}; \quad (2.1)$$

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1}^k - u_i^k}{h_x}; \quad (2.2)$$

$$\frac{\partial u}{\partial t} \approx \frac{u_i^k - u_i^{k-1}}{h_t}. \quad (2.3)$$

Начальное условие аппроксимируется следующим образом:

$$u_i^0 = u_0, \quad i \in [0, I]; \quad (2.4)$$

Граничные условия аппроксимируются выражениями:

$$\frac{u_1^k - u_0^k}{h_x} = 0, \quad k \in [0, K]; \quad u_i^0 = u_0, \quad i \in [0, I]; \quad (2.5)$$

$$\frac{u_I^k - u_{I-1}^k}{h_x} = -\frac{\alpha}{m} u_I^k. \quad k \in [0, K]. \quad (2.6)$$

Для удобства записи разностной схемы проведем замену $m = k$.

Получим неявную разностную схему:

$$\left\{ \begin{array}{l} \frac{u_i^k - u_i^{k-1}}{h_t} = \frac{m}{c} \left[\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2} - \frac{4\alpha}{mb} u_i^k \right] + \varphi_i, \quad i = \overline{1, I-1}, k = \overline{1, K}; \\ u_i^0 = u_0, \quad i = \overline{0, I}; \\ \frac{u_1^k - u_0^k}{h_x} = 0, \quad k = \overline{0, K}; \\ \frac{u_I^k - u_{I-1}^k}{h_x} = -\frac{\alpha}{m} u_I^k, \quad k = \overline{0, K}. \end{array} \right. \quad (2.7)$$

2.2 Аппроксимация простейшей разностной схемы

Краевая задача (1.16) аппроксимируется на решении u разностной схемой, если для любой последовательности сгущающихся сеток выполняются условия:

$$1) f_h \xrightarrow{h \rightarrow 0} f \Leftrightarrow \|f_h - [f]_h\|_{F_h} \xrightarrow{h \rightarrow 0} 0; \quad (2.8)$$

$$2) \delta f_h \xrightarrow{h \rightarrow 0} 0 \Leftrightarrow \|\delta f_h\|_{F_h} = \|L_h[u]_h - f_h\|_{F_h} \xrightarrow{h \rightarrow 0} 0; \quad (2.9)$$

$$\text{где } f = \begin{pmatrix} f^1 \\ f^2 \\ f^3 \\ f^4 \end{pmatrix} = \begin{pmatrix} \varphi(x) \\ u_0 \\ 0 \\ -\frac{\alpha}{m}u|_{x=l} \end{pmatrix}, \quad f_h = \begin{pmatrix} f_h^1 \\ f_h^2 \\ f_h^3 \\ f_h^4 \end{pmatrix} = \begin{pmatrix} \varphi_i \\ u_0 \\ 0 \\ -\frac{\alpha}{m}u_l^k \end{pmatrix},$$

$[\cdot]_h: U \rightarrow U_h$ - оператор дискретизации,

Введем вектор невязок:

$$\delta f_h = \begin{pmatrix} \delta f_h^1 \\ \delta f_h^2 \\ \delta f_h^3 \\ \delta f_h^4 \end{pmatrix} = \begin{pmatrix} L_h^1[u]_h - f_h^1 \\ L_h^2[u]_h - f_h^2 \\ L_h^3[u]_h - f_h^3 \\ L_h^4[u]_h - f_h^4 \end{pmatrix}, \quad (2.10)$$

где L_h^i – операторы для каждой из i -го уравнения системы, $i = \overline{1,4}$,

$[u]_h$ – дискретный аналог точного решения, δf_h^i – невязки уравнений системы.

Очевидно, условие (2.8) выполняется.

Проверим выполнение условия (2.9) и, если оно верно, то найдем порядок аппроксимации для разностной схемы:

$$\begin{aligned} \delta f_h^1 &= \{L_h^1[u]_h - f_h^1\}_{x_i}^{t_k} = \\ &= \left\{ \frac{1}{h_t} \left(u - u + u'_t h_t - u''_{tt} \frac{h_t^2}{2} + u'''_{ttt} \frac{h_t^3}{6} - u^{IV}_{tttt} \frac{h_t^4}{24} + O(h_t^5) \right) - \right. \end{aligned}$$

$$\begin{aligned}
& -\frac{m}{ch_x^2} \left(u - u + u'_x h_x - u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} - u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) - \right. \\
& \left. - 2u + u - u'_x h_x + u''_{xx} \frac{h_x^2}{2} - u'''_{xxx} \frac{h_x^3}{6} + u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) \right) + \\
& + \frac{4\alpha}{cb} u - \varphi(x) \Bigg\}_{x_i}^{t_k} = \\
& = \left\{ u'_t - u''_{tt} \frac{h_t}{2} + u'''_{ttt} \frac{h_t^2}{6} - u^{IV}_{tttt} \frac{h_t^3}{24} + O(h_t^4) - \right. \\
& \left. - \frac{2m}{c} \left(\frac{u''_{xx}}{2} + u^{IV}_{xxxx} \frac{h_x^2}{24} + O(h_x^3) \right) + \frac{4\alpha}{cb} u - \varphi(x) \right\}_{x_i}^{t_k} = \\
& = \left\{ u'_t - u''_{tt} \frac{h_t}{2} + u'''_{ttt} \frac{h_t^2}{6} - u^{IV}_{tttt} \frac{h_t^3}{24} + O(h_t^4) - \right. \\
& \left. - \frac{2m}{c} \left(\frac{u''_{xx}}{2} + u^{IV}_{xxxx} \frac{h_x^2}{24} + O(h_x^3) \right) + \right. \\
& \left. + \frac{4\alpha}{cb} u - u'_t + \frac{\alpha}{c} u''_{xx} - \frac{4\alpha}{cb} u \right\}_{x_i}^{t_k} = O(h_x^2, h_t). \\
& \delta f_h^2 = \{L_h^2[u]_h - f_h^2\}_{x_i}^{t_0} = u(x_i, t_0) - u_0 = 0. \\
& \delta f_h^3 = \{L_h^3[u]_h - f_h^3\}_{x_0}^{t_k} = \\
& = \left\{ \frac{1}{h_x} \left(u + u'_x h_x - u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} - u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) - u \right) \right\}_{x_i}^{t_k} = \\
& = \left\{ 0 - u''_{xx} \frac{h_x}{2} + u'''_{xxx} \frac{h_x^2}{6} - u^{IV}_{xxxx} \frac{h_x^3}{24} + O(h_x^4) \right\}_{x_i}^{t_k} = O(h_x).
\end{aligned}$$

$$\begin{aligned}
\delta f_h^4 &= \{L_h^4[u]_h - f_h^4\}_{x_i}^{t_k} = \\
&= \left\{ \frac{1}{h_x} \left(u + u'_x h_x - u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} - u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) \right) + \frac{\alpha}{m} u \right\}_{x_i}^{t_k} = \\
&= \left\{ u'_x - u''_{xx} \frac{h_x}{2} + u'''_{xxx} \frac{h_x^2}{6} - u^{IV}_{xxxx} \frac{h_x^3}{24} + O(h_x^4) - u'_x \right\}_{x_i}^{t_k} = O(h_x).
\end{aligned}$$

Норма вектора невязок: $\|\delta f_h\|_{F_h} = \max_{i=1,4} \{\delta f_h^i\}$.

Следовательно, $\|\delta f_h\|_{F_h} = O(h_t, h_x)$. Условие (2.9) выполняется, разностная схема (2.7) аппроксимирует на решении u краевую задачу (1.16) и имеет первый порядок аппроксимации и относительно h_t , и относительно h_x .

2.3 Устойчивость разностной схемы

Разностная схема (2.7) устойчива, если существует $\delta > 0, h_0 > 0$ и $c > 0$ такие, что для любых возмущений с $h < h_0$ и $\delta f_h \in F_h$:

$\|\delta - f_h\| \leq \delta$ выполняются следующие условия:

- 1) Решение (2.7) существует и единственно,
- 2) $\|\delta u_h\|_{U_h} \leq c \|\delta f_h\|_{F_h}$, где $c - const$.

Существование и единственность решения схемы вытекает из существования и единственности решения, полученного с помощью метода прогонки.

Проверим выполнение второго условия. Пусть правая часть разностной схемы (2.7) получила возмущение. Тогда она примет вид:

$$\left\{ \begin{array}{l} \frac{u_i^k - u_i^{k-1}}{h_t} = \frac{m}{c} \left[\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2} - \frac{4\alpha}{mb} u_i^k \right] + \varphi_i^k, \\ \quad i = \overline{1, I-1}, \quad k = \overline{1, K}; \\ \quad u_i^0 = \psi_i, \quad i = \overline{0, I}; \\ \quad \frac{u_1^k - u_0^k}{h_x} = 0, \quad k = \overline{0, K}; \\ \quad \frac{u_I^k - u_{I-1}^k}{h_x} = -\frac{\alpha}{m} u_I^k, \quad k = \overline{0, K}. \end{array} \right. \quad (2.11)$$

Определим нормы:

$$\|u_h\|_{U_h} = \max_{\substack{i=\overline{0, I} \\ k=\overline{0, K}}} |u_i^k|; \quad (2.12)$$

$$\|f_h\|_{F_h} = \max \left\{ \max_{i=\overline{0, I}} |\psi_i|, \max_{\substack{i=\overline{1, I-1} \\ k=\overline{1, K}}} |\varphi_i^k| \right\}. \quad (2.13)$$

Рассмотрим первое уравнение системы,

где $\gamma_1 = \frac{2mh_t}{ch_x^2} + \frac{4\alpha h_t}{cb}$, $\gamma_2 = \frac{mh_t}{ch_x^2}$:

$$\begin{aligned} (1 + \gamma_1)u_i^k &= u_i^{k-1} + \gamma_2(u_{i-1}^k + u_{i+1}^k) + \varphi_i^k h_t, \\ i &= \overline{1, I-1}, \quad k = \overline{1, K}. \end{aligned} \quad (2.14)$$

Воспользуемся оценками:

$$|\varphi_i^k| \leq \|f_h\|_{F_h}, \quad i = \overline{0, I}, k = \overline{0, K}; \quad (2.15)$$

$$|u_i^k| \leq \max_{j=\overline{1, I-1}} |u_j^k|, \quad i = \overline{0, I}, k = \overline{0, K}. \quad (2.16)$$

Воспользуемся неравенством треугольника и преобразуем уравнение (2.14) в неравенство:

$$\begin{aligned} (1 + \gamma_1)|u_i^k| &\leq |u_i^{k-1}| + \gamma_2(|u_{i-1}^k| + |u_{i+1}^k|) + \varphi_i^k h_t, \\ i &= \overline{1, I-1}, \quad k = \overline{1, K}. \end{aligned} \quad (2.17)$$

Воспользуемся оценками (2.15) и (2.16) в неравенстве (2.17):

$$\begin{aligned} (1 + \gamma_1)|u_i^k| &\leq \max_{j=\overline{1, I-1}} |u_j^{k-1}| + 2\gamma_2 \max_{j=\overline{1, I-1}} |u_j^k| + h_t \|f_h\|_{F_h}, \\ i &= \overline{1, I-1}, \quad k = \overline{1, K}. \end{aligned} \quad (2.18)$$

Т.к. правая часть (2.18) не зависит от i , то данные неравенства справедливы и для максимальных левых частей по i .

В итоге получим:

$$(1 + \gamma_1) \max_{j=1, I-1} |u_i^k| \leq |u_j^k| \max_{j=1, I-1} |u_j^{k-1}| + 2\gamma_2 \max_{j=1, I-1} |u_j^k| + h_t \|f_h\|_{F_h}, \quad k = \overline{1, K} \quad (2.19)$$

Введем $\gamma = \gamma_1 - 2\gamma_2 = \frac{4\alpha h_t}{cb}$:

$$(1 + \gamma) \max_{j=1, I-1} |u_j^k| \leq \max_{j=1, I-1} |u_j^{k-1}| + h_t \|f_h\|_{F_h}, \quad k = \overline{1, K} \quad (2.20)$$

Усилим неравенство (2.20), отбросив слева γ , так как $1 \leq 1 + \gamma$ и получим итоговое неравенство для проверки устойчивости схемы (2.7):

$$\max_{j=1, I-1} |u_j^k| \leq \max_{j=1, I-1} |u_j^{k-1}| + h_t \|f_h\|_{F_h}, \quad k = \overline{1, K}. \quad (2.21)$$

Теперь рассмотрим все $k = \overline{1, K}$:

$$k = 1: \max_{j=1, I-1} |u_j^1| \leq \max_{j=1, I-1} |u_j^0| + h_t \|f_h\|_{F_h}. \quad (2.22)$$

Очевидно, что:

$$\max_{j=1, I-1} |u_j^0| \leq \max_{j=0, I} |u_j^0| \leq \|f_h\|_{F_h}. \quad (2.23)$$

Подставим (2.23) в неравенство (2.22):

$$\max_{j=1, I-1} |u_j^1| \leq \max_{j=0, I} |u_j^0| + h_t \|f_h\|_{F_h} \leq (1 + h_t) \|f_h\|_{F_h};$$

Аналогично для других $k = \overline{2, K}$

$$k = 2: \max_{j=1, I-1} |u_j^2| \leq \max_{j=1, I-1} |u_j^1| + h_t \|f_h\|_{F_h} \leq (1 + 2h_t) \|f_h\|_{F_h};$$

...

$$\begin{aligned} k = K: \max_{j=0, I} |u_j^K| &\leq \max_{j=1, I-1} |u_j^1| + h_t \|f_h\|_{F_h} \leq (1 + Kh_t) \|f_h\|_{F_h} \leq \\ &\leq (1 + T) \|f_h\|_{F_h} \leq c \|f_h\|_{F_h}. \end{aligned}$$

Отсюда, $\|u_h\|_{U_h} \leq c \|f_h\|_{F_h}$. Доказано, что схема безусловно устойчива.

2.4 Метод прогонки для простейшей неявной разностной схемы

В пункте 2.1 курсовой работы была получена разностная схема, аппроксимирующая задачу (1.16):

$$\left\{ \begin{array}{l} \frac{u_i^k - u_i^{k-1}}{h_t} = \frac{m}{c} \left[\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2} - \frac{4\alpha}{mb} u_i^k \right] + \varphi_i^k, i = \overline{1, I-1}, k = \overline{1, K}; \\ u_i^0 = u_0, i = \overline{0, I}; \\ \frac{u_1^k - u_0^k}{h_x} = 0, k = \overline{0, K}; \\ \frac{u_I^k - u_{I-1}^k}{h_x} = -\frac{\alpha}{m} u_I^k, k = \overline{0, K}. \end{array} \right.$$

Ее можно привести к наиболее общему виду:

$$(1 + \gamma_1)u_i^k = u_i^{k-1} + \gamma_2(u_{i-1}^k + u_{i+1}^k) + \varphi_i h_t, \quad (2.21)$$

$$\text{где } \gamma_1 = \frac{2mh_t}{ch_x^2} + \frac{4\alpha}{cb}, \quad \gamma_2 = \frac{ah_t}{ch_x^2}.$$

Приведем систему к общему виду с трехдиагональной матрицей и вектором значений правой части:

$$A_i u_{i+1}^k + B_i u_i^k + C_i u_{i-1}^k = F_i^{k-1}; \quad (2.22)$$

$$\text{где } A_i = C_i = -\gamma_2, \quad B_i = 1 + \gamma_1, \quad F_i^{k-1} = u_i^{k-1} + \varphi_i h_t; \quad i = \overline{1, I-2}, k = \overline{1, K}$$

Рассчитаем матричные коэффициенты на границах:

$$i = 1: (1 + \gamma_1)u_1^k = u_1^{k-1} + \gamma_2(u_0^k + u_2^k) + \varphi_1 h_t.$$

Так как по граничному условию $u_1^k = u_0^k$, то:

$$(1 + \gamma_1 - \gamma_2)u_1^k = u_1^{k-1} + \gamma_2 u_2^k + \varphi_1 h_t.$$

Коэффициенты матрицы в этом случае выглядят вот так:

$$A_1 = -\gamma_2, \quad B_1 = 1 + \gamma_1 - \gamma_2, \quad C_1 = 0.$$

Вектор значений правой части в этом случае выглядит следующим образом:

$$F_1^{k-1} = u_1^{k-1} + \varphi_1 h_t.$$

Рассчитаем коэффициенты на другом конце:

$$i = I - 1: (1 + \gamma_1)u_{I-1}^k = u_{I-1}^{k-1} + \gamma_2(u_{I-2}^k + u_I^k) + \varphi_{I-1} h_t.$$

Так как по граничному условию $u_I^k - u_{I-1}^k = -\frac{\alpha h_t}{m} u_I^k$, то:

$$\left(1 + \frac{\alpha h_t}{m}\right) u_I^k = u_{I-1}^k;$$

Подставим это в основное уравнение схемы:

$$(1 + \gamma_1) \left(1 + \frac{\alpha h_t}{m}\right) u_I^k = u_{I-1}^{k-1} + \gamma_2 u_I^k + \gamma_2 u_{I-1}^k + \varphi_{I-1} h_t;$$

$$\left[(1 + \gamma_1) \left(1 + \frac{\alpha h_t}{m}\right) - \gamma_2 \right] u_I^k = u_{I-1}^{k-1} + \gamma_2 u_{I-1}^k + \varphi_{I-1} h_t.$$

Коэффициенты матрицы в этом случае выглядят вот так:

$$A_{I-1} = 0, \quad B_{I-1} = (1 + \gamma_1) \left(1 + \frac{\alpha h_t}{m}\right) - \gamma_2, \quad C_{I-1} = -\gamma_2.$$

Вектор значений правой части в этом случае выглядит следующим образом:

$$F_{I-1}^{k-1} = u_{I-1}^{k-1} + \varphi_{I-1} h_t.$$

Система имеет трехдиагональную структуру. Предположим, что существуют такие наборы чисел α_i, β_i , при которых:

$$u_i^k = \alpha_{i+1} u_{i+1}^k + \beta_{i+1}. \quad (2.23)$$

Подставим в нулевой слой по i :

$$\begin{cases} B_0 u_0^k + C_0 u_1^k = F_0^{k-1}, \\ u_0^k = \alpha_1 u_1^k + \beta_1; \end{cases}$$

и получим $B_0(\alpha_1 u_1^k + \beta_1) + C_0 u_1^k = F_0^{k-1}$.

Выразим коэффициенты α_1, β_1 :

$$\begin{cases} \alpha_1 B_0 = -C_0; \\ \beta_1 = \frac{F_0^{k-1}}{B_0}; \end{cases}$$

$$\begin{cases} \alpha_1 = -\frac{C_0}{B_0}; \\ \beta_1 = \frac{F_0^{k-1}}{B_0}. \end{cases}$$

Рассчитаем коэффициенты α_i, β_i для остальных $i = \overline{1, I-1}$:

$$A_i u_{i+1}^k + B_i u_i^k + C_i u_{i-1}^k = F_i^{k-1};$$

$$A_i(\alpha_i u_i^k + \beta_i) + B_i u_i^k + C_i u_{i-1}^k = F_i^{k-1}.$$

Итоговые значения коэффициентов α_i, β_i :

$$\begin{cases} \alpha_1 = -\frac{C_i}{\alpha_i A_i + B_i}; \\ \beta_1 = \frac{F_i^{k-1} - A_i \beta_i}{\alpha_i A_i + B_i}. \end{cases}$$

Рассчитаем значения u_I^k, u_{I-1}^k для обратного хода метода прогонки:

$$\begin{aligned} u_I^k: \frac{u_I^k - u_{I-1}^k}{h_x} &= -\frac{\alpha}{m} u_I^k; \\ u_I^k &= \frac{u_{I-1}^k}{1 + \frac{\alpha h_x}{m}}. \end{aligned} \quad (2.24)$$

С помощью рекуррентной формулы выразим $u_{I-2}^k = \alpha_{I-1} u_{I-1} + \beta_{I-1}$:

$$\begin{aligned} (1 + \gamma_1) u_{I-1}^k &= \gamma_2 (u_{I-2}^k + u_I^k) + F_{I-1}^{k-1}; \\ (1 + \gamma_1) u_{I-1}^k &= \gamma_2 \left(\alpha_{I-1} u_{I-1} + \beta_{I-1} + \frac{u_{I-1}^k}{1 + \frac{\alpha h_x}{m}} \right) + F_{I-1}^{k-1}; \\ \left(1 + \gamma_1 - \gamma_2 \alpha_{I-1} - \gamma_2 \frac{u_{I-1}^k}{1 + \frac{\alpha h_x}{m}} \right) u_{I-1}^k &= F_{I-1}^{k-1} + \gamma_2 \beta_{I-1}. \end{aligned}$$

Получим итоговую формулу u_{I-1}^k для расчета в начале обратного хода метода прогонки:

$$u_{I-1}^k = \frac{F_{I-1}^{k-1} + \gamma_2 \beta_{I-1}}{\left(1 + \gamma_1 - \gamma_2 \alpha_{I-1} - \gamma_2 \frac{u_{I-1}^k}{1 + \frac{\alpha h_x}{m}} \right)}. \quad (2.25)$$

2.5 Исследование сходимости к решению исходной задачи

Так как разностная схема (2.9) аппроксимирует задачу (1.16) на решении u с первым порядком по временному шагу h_t и с первым порядком по пространственному шагу h_x и является устойчивой, то решение u_h

разностной задачи сходится к решению исходной задачи $[u]_h$, причем скорость сходимости линейная и по h_t , и по h_x .

3 НЕЯВНАЯ РАЗНОСТНАЯ СХЕМА ПОВЫШЕННОГО ПОРЯДКА

3.1 Построение неявной разностной схемы

Сетка задается узловыми точками (x_i, t_k) , где

$$x_i = ih_x, \quad i = \overline{0, I}, \quad h_x = \frac{l}{I},$$

$$t_k = kh_t, \quad k = \overline{0, K}, \quad h_t = \frac{T}{K}.$$

Аппроксимируем на этой сетке производные разностными соотношениями:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2};$$

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1}^k - u_{i-1}^k}{2h_x};$$

$$\frac{\partial u}{\partial t} \approx \frac{u_i^k - u_i^{k-1}}{h_t}.$$

Начальное условие аппроксимируется следующим образом:

$$u_i^0 = u_0, \quad i = \overline{0, I};$$

Граничные условия аппроксимируются выражениями:

$$\frac{u_1^k - u_{-1}^k}{2h_x} = 0, \quad k = \overline{0, K};$$

$$\frac{u_{I+1}^k - u_{I-1}^k}{2h_x} = -\frac{\alpha}{m} u_I^k, \quad k = \overline{0, K}.$$

Для удобства записи разностной схемы в исходной записи проведем замену $m = k$.

Получим неявную разностную схему:

$$\left\{ \begin{array}{l} \frac{u_i^k - u_i^{k-1}}{h_t} = \frac{m}{c} \left[\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2} - \frac{4\alpha}{mb} u_i^k \right] + \varphi_i, \\ i = \overline{1, I-1}, \quad k = \overline{1, K}; \\ u_i^0 = u_0, \quad i = \overline{0, I}; \\ \frac{u_1^k - u_{-1}^k}{2h_x} = 0, \quad k = \overline{0, K}; \\ \frac{u_{I+1}^k - u_{I-1}^k}{2h_x} = -\frac{\alpha}{m} u_I^k, \quad k = \overline{0, K}. \end{array} \right. \quad (3.1)$$

3.2 Аппроксимация неявной разностной схемы

Краевая задача (1.16) аппроксимируется на решении u разностной схемой, если для любой последовательности сгущающихся сеток выполняются условия:

$$1) f_h \xrightarrow{h \rightarrow 0} f \Leftrightarrow \|f_h - [f]_h\|_{F_h} \xrightarrow{h \rightarrow 0} 0; \quad (3.2)$$

$$2) \delta f_h \xrightarrow{h \rightarrow 0} 0 \Leftrightarrow \|\delta f_h\|_{F_h} = \|L_h[u]_h - f_h\|_{F_h} \xrightarrow{h \rightarrow 0} 0; \quad (3.3)$$

$$\text{где } f = \begin{pmatrix} f^1 \\ f^2 \\ f^3 \\ f^4 \end{pmatrix} = \begin{pmatrix} \varphi(x) \\ u_0 \\ 0 \\ -\frac{\alpha}{m} u|_{x=l} \end{pmatrix}, \quad f_h = \begin{pmatrix} f_h^1 \\ f_h^2 \\ f_h^3 \\ f_h^4 \end{pmatrix} = \begin{pmatrix} \varphi_i \\ u_0 \\ 0 \\ -\frac{\alpha}{m} u_I^k \end{pmatrix},$$

$[\cdot]_h: U \rightarrow U_h$ - оператор дискретизации,

Введем вектор невязок:

$$\delta f_h = \begin{pmatrix} \delta f_h^1 \\ \delta f_h^2 \\ \delta f_h^3 \\ \delta f_h^4 \end{pmatrix} = \begin{pmatrix} L_h^1[u]_h - f_h^1 \\ L_h^2[u]_h - f_h^2 \\ L_h^3[u]_h - f_h^3 \\ L_h^4[u]_h - f_h^4 \end{pmatrix}, \quad (3.4)$$

где L_h^i – операторы для каждой из i -го уравнения системы, $i = \overline{1, 4}$,

$[u]_h$ – дискретный аналог точного решения, δf_h^i – правые части уравнений системы.

Очевидно, условие (3.2) выполняется.

Проверим выполнение условия (3.3) и, если оно верно, то найдем порядок аппроксимации для разностной схемы:

$$\begin{aligned}
\delta f_h^1 &= \{L_h^1[u]_h - f_h^1\}_{x_i}^{t_k} = \\
&= \left\{ \frac{1}{h_t} \left(u - u + u'_t h_t - u''_{tt} \frac{h_t^2}{2} + u'''_{ttt} \frac{h_t^3}{6} - u^{IV}_{tttt} \frac{h_t^4}{24} + O(h_t^5) \right) - \right. \\
&\quad - \frac{m}{ch_x^2} \left(u - u + u'_x h_x - u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} - u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) - \right. \\
&\quad \left. \left. - 2u + u - u'_x h_x + u''_{xx} \frac{h_x^2}{2} - u'''_{xxx} \frac{h_x^3}{6} + u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) \right) + \right. \\
&\quad \left. + \frac{4\alpha}{cb} u - \varphi(x) \right\}_{x_i}^{t_k} = \\
&= \left\{ u'_t - u''_{tt} \frac{h_t}{2} + u'''_{ttt} \frac{h_t^2}{6} - u^{IV}_{tttt} \frac{h_t^3}{24} + O(h_t^4) - \right. \\
&\quad \left. - \frac{2m}{c} \left(\frac{u''_{xx}}{2} + u^{IV}_{xxxx} \frac{h_x^2}{24} + O(h_x^3) \right) + \frac{4\alpha}{cb} u - \varphi(x) \right\}_{x_i}^{t_k} = \\
&= \left\{ u'_t - u''_{tt} \frac{h_t}{2} + u'''_{ttt} \frac{h_t^2}{6} - u^{IV}_{tttt} \frac{h_t^3}{24} + O(h_t^4) - \right. \\
&\quad \left. - \frac{2m}{c} \left(\frac{u''_{xx}}{2} + u^{IV}_{xxxx} \frac{h_x^2}{24} + O(h_x^3) \right) + \right. \\
&\quad \left. + \frac{4\alpha}{cb} u - u'_t + \frac{\alpha}{c} u''_{xx} - \frac{4\alpha}{cb} u \right\}_{x_i}^{t_k} = O(h_x^2, h_t). \\
\delta f_h^2 &= \{L_h^2[u]_h - f_h^2\}_{x_i}^{t_0} = u(x_i, t_0) - u_0 = 0.
\end{aligned}$$

$$\begin{aligned}
\delta f_h^3 &= \{L_h^3[u]_h - f_h^3\}_{x_0}^{t_k} = \\
&= \left\{ \frac{1}{2h_t} \left(u + u'_x h_x + u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} + u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) - u + \right. \right. \\
&\quad \left. \left. + u'_x h_x - u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} - u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) \right) \right\}_{x_i}^{t_k} = \\
&= \left\{ u'_x h_x + u'''_{xxx} \frac{h_x^2}{6} + O(h_x^3) \right\}_{x_i}^{t_k} = O(h_x^2). \\
\delta f_h^4 &= \{L_h^4[u]_h - f_h^4\}_{x_i}^{t_k} = \\
&= \left\{ \frac{1}{2h_x} \left(u + u'_x h_x + u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} + u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) + u'_x h_x - \right. \right. \\
&\quad \left. \left. - u''_{xx} \frac{h_x^2}{2} + u'''_{xxx} \frac{h_x^3}{6} - u^{IV}_{xxxx} \frac{h_x^4}{24} + O(h_x^5) \right) + \frac{\alpha}{m} u \right\}_{x_i}^{t_k} = \\
&= \left\{ u'_x + u'''_{xxx} \frac{h_x^2}{6} - u'_x + O(h_x^3) \right\}_{x_i}^{t_k} = O(h_x^2).
\end{aligned}$$

Норма вектора невязок: $\|\delta f_h\|_{F_h} = \max_{i=1,4} \{\delta f_h^i\}$.

Следовательно, $\|\delta f_h\|_{F_h} = O(h_t, h_x^2)$. Условие выполняется, разностная схема (3.1) аппроксимирует на решении u краевую задачу (1.16) и имеет первый порядок аппроксимации относительно h_t , и второй относительно h_x .

3.3 Устойчивость разностной схемы

Разностная схема (3.1) устойчива, если существует $\delta > 0, h_0 > 0$ и $c > 0$ такие, что для любых возмущений с $h < h_0$ и $\delta f_h \in F_h$:

$\|\delta - f_h\| \leq \delta$ выполняются следующие условия:

- 3) Решение (3.1) существует и единственно,
- 4) $\|\delta u_h\|_{U_h} \leq c \|f_h\|_{F_h}$, где $c - const$.

Существование и единственность решения схемы вытекает из существования и единственности решения, полученного с помощью метода прогонки[4].

Проверим выполнение второго условия. Пусть правая часть разностной схемы (3.1) получила возмущение. Тогда она примет вид:

$$\left\{ \begin{array}{l} \frac{u_i^k - u_i^{k-1}}{h_t} = \frac{m}{c} \left[\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2} - \frac{4\alpha}{mb} u_i^k \right] + \varphi_i^k, \quad i = \overline{1, I-1}, \quad k = \overline{1, K}; \\ u_i^0 = \psi_i, \quad i = \overline{0, I}; \\ \frac{u_1^k - u_{-1}^k}{2h_x} = 0, \quad k = \overline{0, K}; \\ \frac{u_{I+1}^k - u_{I-1}^k}{2h_x} = -\frac{\alpha}{m} u_I^k, \quad k = \overline{0, K}. \end{array} \right. \quad (3.5)$$

Определим нормы:

$$\|u_h\|_{U_h} = \max_{\substack{i=\overline{0, I} \\ k=\overline{0, K}}} |u_i^k|; \quad (3.6)$$

$$\|f_h\|_{F_h} = \max \left\{ \max_{i=\overline{0, I}} |\psi_i|, \max_{\substack{i=\overline{1, I-1} \\ k=\overline{1, K}}} |\varphi_i^k| \right\}. \quad (3.7)$$

Рассмотрим первое уравнение системы,

$$\text{где } \gamma_1 = \frac{2mh_t}{ch_x^2} + \frac{4\alpha h_t}{cb}, \quad \gamma_2 = \frac{mh_t}{ch_x^2} :$$

$$\begin{aligned} (1 + \gamma_1)u_i^k &= u_i^{k-1} + \gamma_2(u_{i-1}^k + u_{i+1}^k) + \varphi_i^k h_t, \\ i &= \overline{1, I-1}, \quad k = \overline{1, K}. \end{aligned} \quad (3.8)$$

Воспользуемся оценками:

$$|\varphi_i^k| \leq \|f_h\|_{F_h}, \quad i = \overline{0, I}, k = \overline{0, K}; \quad (3.9)$$

$$|u_i^k| \leq \max_{j=\overline{1, I-1}} |u_j^k|, \quad i = \overline{0, I}, k = \overline{0, K}. \quad (3.10)$$

Воспользуемся неравенством треугольника и преобразуем уравнение (3.8) в неравенство:

$$\begin{aligned} (1 + \gamma_1)|u_i^k| &\leq |u_i^{k-1}| + \gamma_2(|u_{i-1}^k| + |u_{i+1}^k|) + \varphi_i^k h_t, \\ i &= \overline{1, I-1}, \quad k = \overline{1, K}. \end{aligned} \quad (3.11)$$

Воспользуемся оценками (3.9) и (3.10) в неравенстве (3.11):

$$(1 + \gamma_1)|u_i^k| \leq \max_{j=\overline{1, I-1}} |u_j^{k-1}| + 2\gamma_2 \max_{j=\overline{1, I-1}} |u_j^k| + h_t \|f_h\|_{F_h}, \quad (3.12)$$

$$i = \overline{1, I-1}, \quad k = \overline{1, K}.$$

Т.к. правая часть (3.12) не зависит от i , то данные неравенства справедливы и для максимальных левых частей по i .

В итоге получим:

$$(1 + \gamma_1) \max_{j=\overline{1, I-1}} |u_i^k| \leq |u_j^k| \max_{j=\overline{1, I-1}} |u_j^{k-1}| +$$

$$+ 2\gamma_2 \max_{j=\overline{1, I-1}} |u_j^k| + h_t \|f_h\|_{F_h}, \quad k = \overline{1, K} \quad (3.13)$$

Введем $\gamma = \gamma_1 - 2\gamma_2 = \frac{4\alpha h_t}{cb}$:

$$(1 + \gamma) \max_{j=\overline{1, I-1}} |u_j^k| \leq \max_{j=\overline{1, I-1}} |u_j^{k-1}| + h_t \|f_h\|_{F_h}, \quad k = \overline{1, K} \quad (3.14)$$

Усилим неравенство (3.14), отбросив слева γ , так как $1 < 1 + \gamma$ и получим итоговое неравенство для проверки устойчивости схемы (3.1):

$$\max_{j=\overline{1, I-1}} |u_j^k| \leq \max_{j=\overline{1, I-1}} |u_j^{k-1}| + h_t \|f_h\|_{F_h}, \quad k = \overline{1, K}. \quad (3.15)$$

Теперь рассмотрим все $k = \overline{1, K}$:

$$k = 1: \max_{j=\overline{1, I-1}} |u_j^1| \leq \max_{j=\overline{1, I-1}} |u_j^0| + h_t \|f_h\|_{F_h}. \quad (3.16)$$

Очевидно, что:

$$\max_{j=\overline{1, I-1}} |u_j^0| \leq \max_{j=\overline{0, I}} |u_j^0| \leq \|f_h\|_{F_h}. \quad (3.17)$$

Подставим (3.17) в неравенство (3.16):

$$\max_{j=\overline{1, I-1}} |u_j^1| \leq \max_{j=\overline{0, I}} |u_j^0| + h_t \|f_h\|_{F_h} \leq (1 + h_t) \|f_h\|_{F_h};$$

Аналогично для других $k = \overline{2, K}$

$$k = 2: \max_{j=\overline{1, I-1}} |u_j^2| \leq \max_{j=\overline{1, I-1}} |u_j^1| + h_t \|f_h\|_{F_h} \leq (1 + 2h_t) \|f_h\|_{F_h};$$

...

$$k = K: \max_{j=\overline{0, I}} |u_j^K| \leq \max_{j=\overline{1, I-1}} |u_j^1| + h_t \|f_h\|_{F_h} \leq (1 + Kh_t) \|f_h\|_{F_h} \leq$$

$$\leq (1 + T) \|f_h\|_{F_h} \leq c \|f_h\|_{F_h}.$$

Отсюда, $\|u_h\|_{U_h} \leq c\|f_h\|_{F_h}$. Доказано, что схема безусловно устойчива.

3.4 Метод прогонки для неявной разностной схемы

В пункте 3.1 курсовой работы была получена разностная схема, аппроксимирующая задачу (1.16):

$$\left\{ \begin{array}{l} \frac{u_i^k - u_i^{k-1}}{h_t} = \frac{m}{c} \left[\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h_x^2} - \frac{4\alpha}{mb} u_i^k \right] + \varphi_i^k, \quad i = \overline{1, I-1}, \quad k = \overline{1, K}; \\ u_i^0 = u_0, \quad i = \overline{0, I}; \\ \frac{u_1^k - u_{-1}^k}{h_x} = 0, \quad k = \overline{0, K}; \\ \frac{u_{I+1}^k - u_{I-1}^k}{2h_x} = -\frac{\alpha}{m} u_I^k, \quad k = \overline{0, K}. \end{array} \right.$$

Ее можно привести к наиболее общему виду:

$$(1 + \gamma_1)u_i^k = u_i^{k-1} + \gamma_2(u_{i-1}^k + u_{i+1}^k) + \varphi_i h_t, \quad (3.18)$$

$$\text{где } \gamma_1 = \frac{2mh_t}{ch_x^2} + \frac{4\alpha}{cb}, \quad \gamma_2 = \frac{ah_t}{ch_x^2}.$$

Приведем систему к общему виду с трехдиагональной матрицей и вектором значений правой части:

$$A_i u_{i+1}^k + B_i u_i^k + C_i u_{i-1}^k = F_i^{k-1}; \quad (3.19)$$

$$\text{где } A_i = C_i = -\gamma_2, \quad B_i = 1 + \gamma_1, \quad F_i^{k-1} = u_i^{k-1} + \varphi_i h_t.$$

Рассчитаем матричные коэффициенты на границах:

$$i = 0: (1 + \gamma_1)u_0^k = u_0^{k-1} + \gamma_2(u_1^k + u_{-1}^k) + \varphi_0 h_t.$$

Так как по граничному условию $u_1^k = u_{-1}^k$, то:

$$(1 + \gamma_1)u_0^k = u_0^{k-1} + 2\gamma_2 u_1^k + \varphi_0 h_t.$$

Коэффициенты матрицы в этом случае выглядят вот так:

$$A_1 = -2\gamma_2, \quad B_1 = 1 + \gamma_1, \quad C_1 = 0.$$

Вектор значений правой части в этом случае выглядит вот так:

$$F_0^{k-1} = u_0^{k-1} + \varphi_0 h_t.$$

Рассчитаем коэффициенты на другом конце:

$$i = I: (1 + \gamma_1)u_I^k = u_I^{k-1} + \gamma_2(u_{I+1}^k - u_{I-1}^k) + \varphi_I h_t.$$

Так как по граничному условию $u_{I+1}^k - u_{I-1}^k = -\frac{\alpha h_t}{m} u_I^k$, то:

$$u_I^k = -\frac{m}{2h_t\alpha}(u_{I+1}^k - u_{I-1}^k);$$

Подставим это в основное уравнение схемы:

$$-\frac{m}{2h_t\alpha}(1 + \gamma_1)(u_{I+1}^k - u_{I-1}^k) = u_I^{k-1} + \gamma_2 u_{I+1}^k + \gamma_2 u_{I-1}^k + \varphi_I h_t;$$

$$-\left(\frac{m}{2h_t\alpha}(1 + \gamma_1) + \gamma_2\right)(u_{I+1}^k - u_{I-1}^k) = u_I^{k-1} + \varphi_I h_t.$$

Коэффициенты матрицы в этом случае выглядят вот так:

$$A_I = 0, \quad B_I = 0, \quad C_I = -\left(\frac{m}{2h_t\alpha}(1 + \gamma_1) + \gamma_2\right).$$

Вектор значений правой части в этом случае выглядит вот так:

$$F_I^{k-1} = u_I^{k-1} + \varphi_I h_t.$$

Система имеет трехдиагональную структуру. Предположим, что существуют такие наборы чисел α_i, β_i , при которых:

$$u_i^k = \alpha_{i+1}u_{i+1}^k + \beta_{i+1}. \quad (3.20)$$

Подставим в нулевой слой по i :

$$\begin{cases} B_0 u_1^k + C_0 u_1^k = F_0^{k-1}, \\ u_0^k = \alpha_1 u_1^k + \beta_1; \end{cases}$$

и получим $(1 + \gamma_1)(\alpha_1 u_1^k + \beta_1) - 2\gamma_2 u_1^k = F_0^{k-1}$.

Выразим коэффициенты α_1, β_1 :

$$\begin{cases} \alpha_1 = -\frac{2\gamma_2}{1 + \gamma_1}; \\ \beta_1 = \frac{F_0^{k-1}}{1 + \gamma_1}. \end{cases}$$

Рассчитаем коэффициенты α_i, β_i для остальных $i = \overline{2, I}$:

Итоговые значения коэффициентов α_i, β_i :

$$\begin{cases} \alpha_i = -\frac{C_i}{\alpha_i A_i + B_i}; \\ \beta_i = \frac{F_i^{k-1} - A_i \beta_i}{\alpha_i A_i + B_i}. \end{cases}$$

Рассчитаем значения u_I^k для расчета обратного хода метода прогонки:

С помощью рекуррентной формулы выразим $u_{I-1}^k = \alpha_I u_I + \beta_I$.

$$(1 + \gamma_1)u_I^k = \gamma_2(u_{I-1}^k + u_{I+1}^k) + F_I^{k-1};$$

$$u_{I+1}^k = u_{I-1}^k - 2\frac{\alpha h_x}{m}u_I^k;$$

$$(1 + \gamma_1)u_I^k - \gamma_2 u_{I-1}^k - \gamma_2 \left(u_{I-1}^k + 2\frac{\alpha h_x}{m}u_I^k \right) = F_I^{k-1};$$

$$u_I^k \left(1 + \gamma_1 - 2\gamma_2 \frac{\alpha h_x}{m} \right) - 2\gamma_2 u_{I-1}^k = F_I^{k-1};$$

$$u_I^k \left(1 + \gamma_1 - 2\gamma_2 \frac{\alpha h_x}{m} \right) - 2\gamma_2 (\alpha_I u_I + \beta_I) = F_I^{k-1};$$

$$u_I^k \left(1 + \gamma_1 - 2\gamma_2 \frac{\alpha h_x}{m} - 2\gamma_2 \alpha_I \right) = F_I^{k-1} + 2\gamma_2 \beta_I.$$

$$u_I^k = \frac{F_I^{k-1} + 2\gamma_2 \beta_I}{1 + \gamma_1 - 2\gamma_2 \frac{\alpha h_x}{m} - 2\gamma_2 \alpha_I}. \quad (3.21)$$

3.5 Исследование сходимости к решению исходной задачи

Так как разностная схема (3.1) аппроксимирует задачу (1.16) на решении u с первым порядком по временному шагу h_t и со вторым порядком по пространственному шагу h_x и является устойчивой, то решение u_h разностной задачи сходится к решению исходной задачи $[u]_h$, причем скорость сходимости линейная по h_t , и квадратичная по h_x .

4 ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ РЕШЕНИЯ

Приведём примеры решений, полученных с помощью неявных разностных схем, и сравним их с аналитическим решением.

На синем графике изображена простейшая неявная разностная схема, а на зеленом – неявная схема повышенного порядка. Серый график показывает аналитическое решение задачи.

Для значений $I = 100$ и $K = 100$ хорошо заметно расхождение двух сеточных и точного решений, продемонстрированных на рисунке 1. Особенно это заметно на середине интервала $[0; l]$.

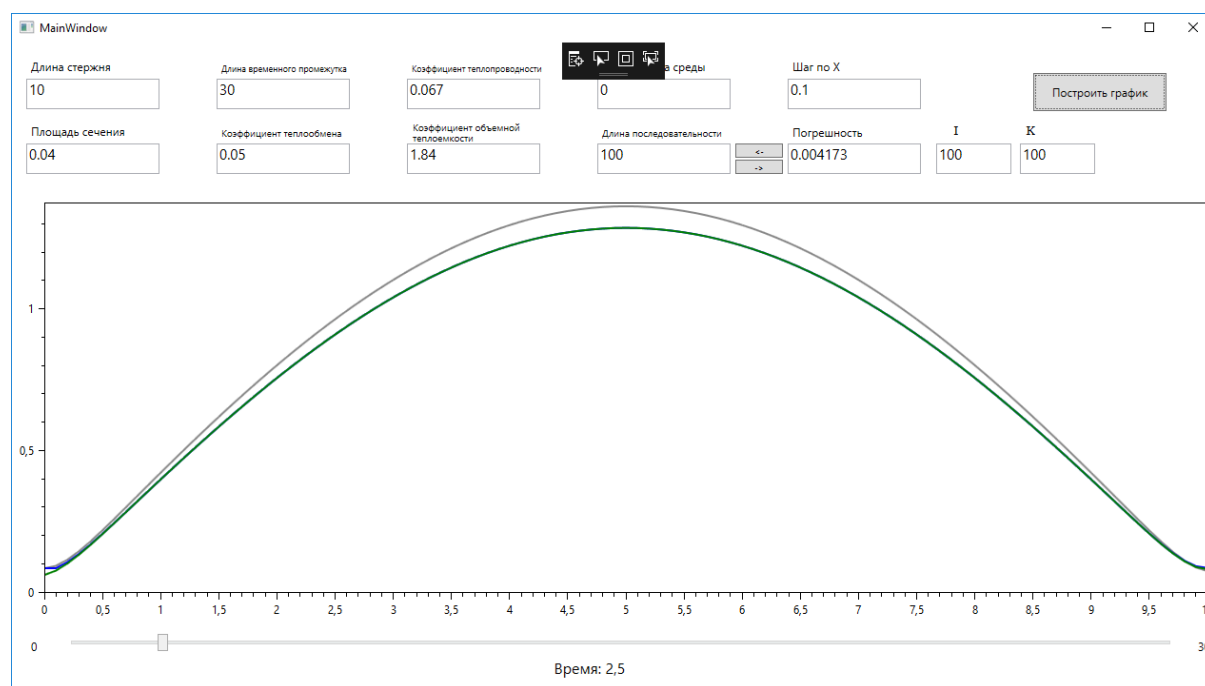


Рисунок 1 - График распределения температуры при фиксированном $t = 2.5, I = 100, K = 100$.

При увеличении I до 500 и K до 500 расхождение становится практически незаметным (рисунок 2):

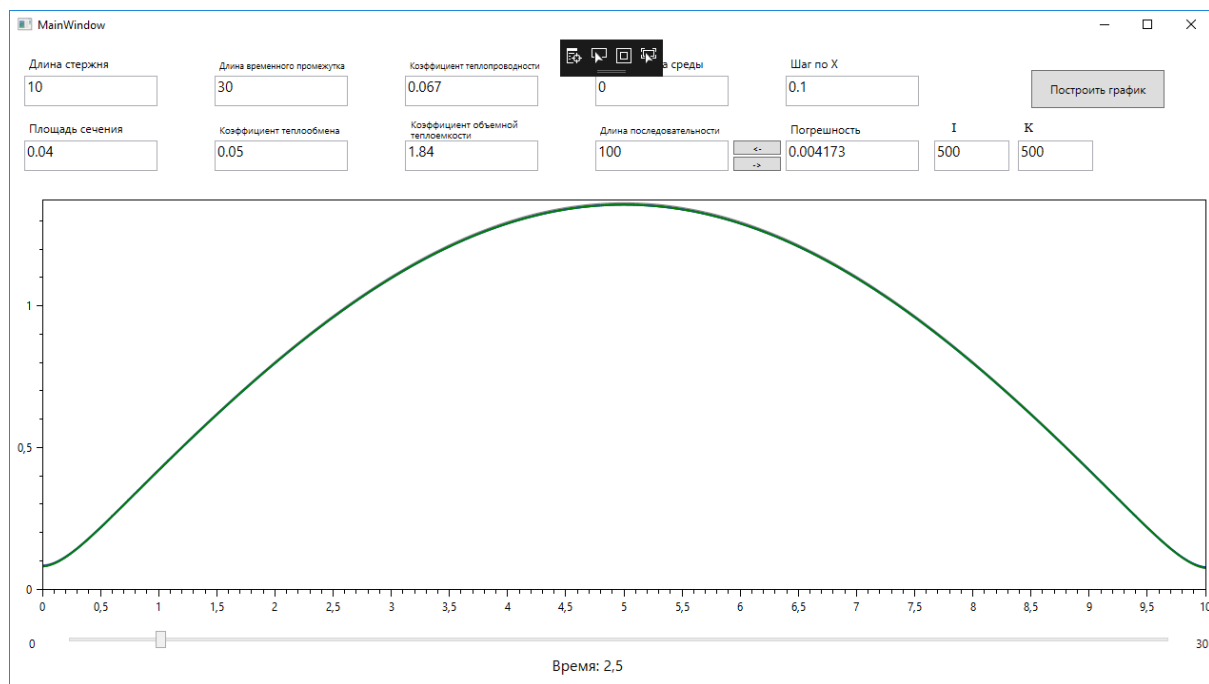


Рисунок 2 - График распределения температуры при фиксированном $t = 2,5$, $I = 500$, $K = 500$.

Для значений $I = 100$ и $K = 100$ при $t = 17.8$ хорошо заметно расхождение двух сеточных и точного решений, продемонстрированных на рисунке 3 на концах интервала x .

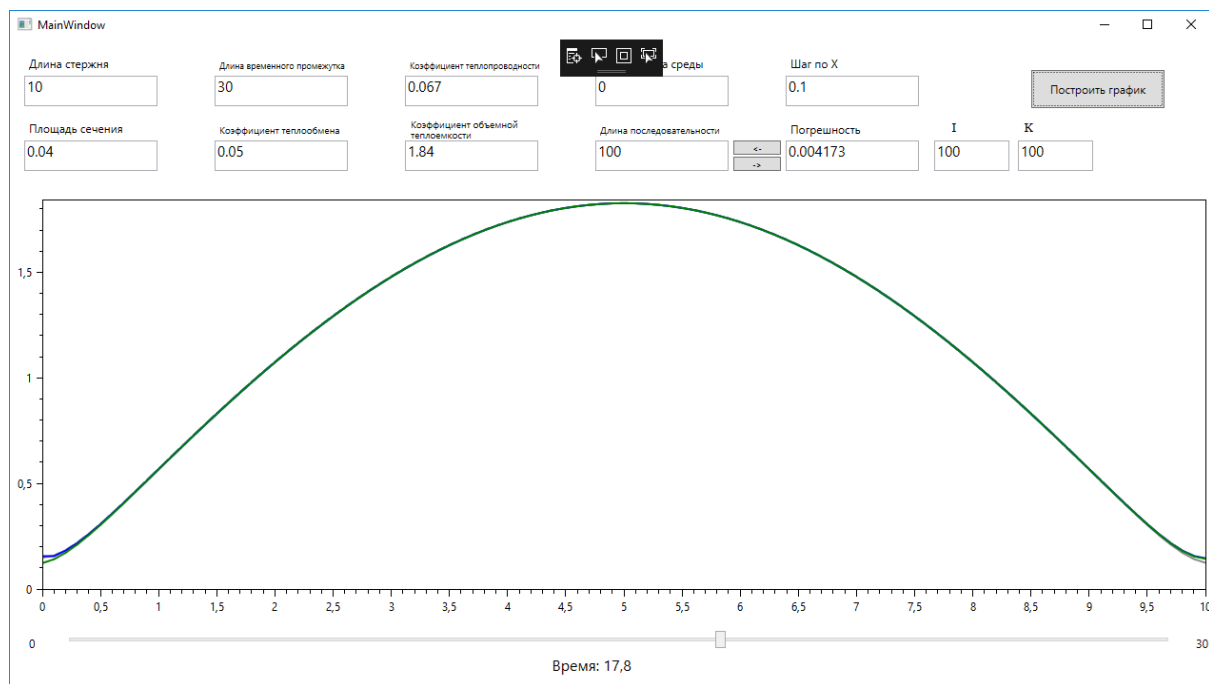


Рисунок 3 - График распределения температуры при фиксированном $t = 17.8$, $I = 100$, $K = 100$.

При увеличении I до 500 и K до 500 расхождение становится практически незаметным (рисунок 4):

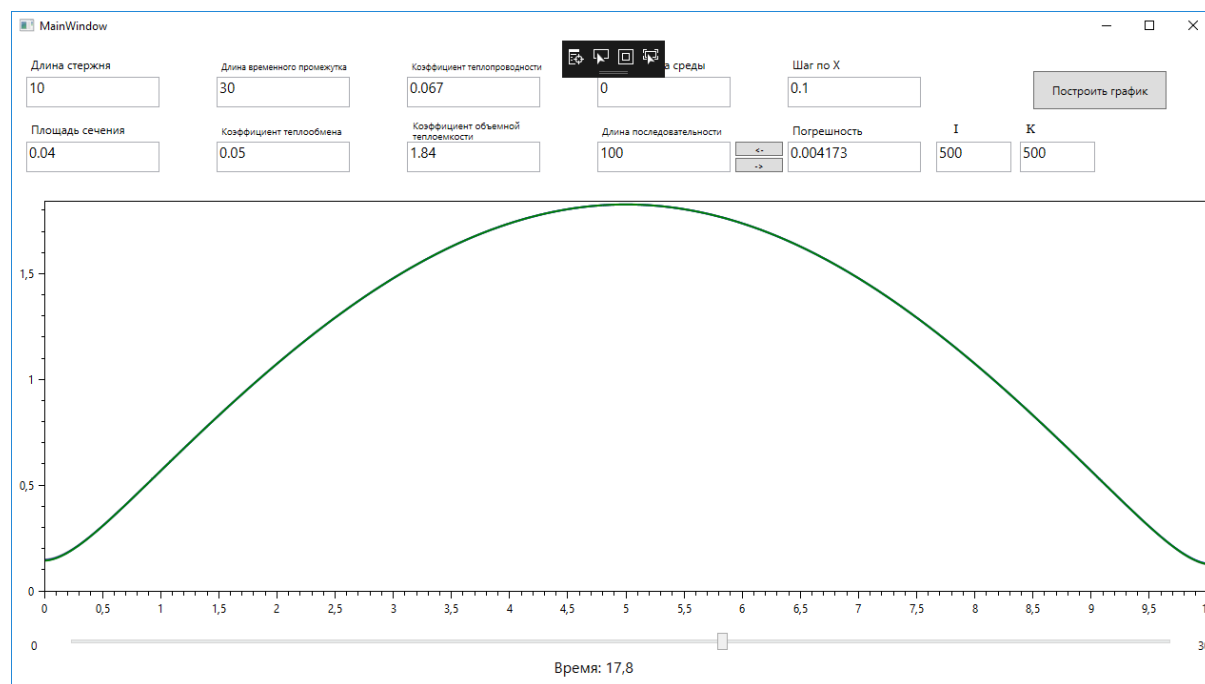


Рисунок 4 - График распределения температуры при фиксированном $t = 17.8$, $I = 500$, $K = 500$.

Видим, что при увеличении количества шагов разбиения сеточное решение приближается к точному решению.

5 ИССЛЕДОВАНИЕ ПОГРЕШНОСТИ С ПОМОЩЬЮ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

Простейшая неявная схема аппроксимирует исходную задачу с первым порядком по временному шагу h_t и с первым порядком по пространственному шагу h_x .

Неявная схема повышенного порядка аппроксимирует исходную задачу с первым порядком по временному шагу h_t и со вторым порядком по пространственному шагу h_x . Таким образом, погрешность сеточного решения можно определить в виде:

$$\varepsilon(h_x, h_t) = \max_{\substack{i=\overline{0,I} \\ k=\overline{0,K}}} |u_i^k - u(x_i, t_k)| \quad (3.22)$$

где $u(x_i, t_k)$ – значение аналитического решения в узле сетки x_i, t_k .

Для исследования простейшей неявной схемы будем использовать следующий алгоритм: рассчитаем сеточную функцию, рассчитаем функцию с помощью рядов, получим погрешность, затем уменьшим шаг вдвое и по пространственной координате, и по временной [5].

Для исследования неявной схемы повышенного порядка будем использовать следующий алгоритм: рассчитаем сеточную функцию, рассчитаем функцию с помощью рядов, получим погрешность, затем уменьшим шаг вдвое по пространственной координате и вчетверо по временной.

Рассмотрим полученные результаты, представленные в таблицах 1 и 2.

Таблица 1 - Результаты вычислительного эксперимента для простейшей неявной схемы

I	K	h_x	h_t	$\varepsilon(h_x, h_t)$	$\varepsilon(h_x, h_t) / \varepsilon\left(\frac{h_x}{2}, \frac{h_t}{2}\right)$
8	4	1,25	7,5	1,375616	-
16	8	0,625	3,75	0,138838	9,908041
32	16	0,3125	1,875	0,436929	0,31776
64	32	0,15625	0,9375	0,062331	7,009767
128	64	0,078125	0,46875	0,11821	0,527292
256	128	0,039063	0,234375	0,019302	6,124402
512	256	0,019531	0,117188	0,030156	0,640065

Таблица 2 - Результаты вычислительного эксперимента для неявной схемы с уточнениями на концах

I	K	h_x	h_t	$\varepsilon(h_x, h_t)$	$\varepsilon(h_x, h_t) / \varepsilon\left(\frac{h_x}{2}, \frac{h_t}{4}\right)$
4	4	0,15625	7,5	1,36235692	-
128	16	0,078125	1,875	0,43676817	3,119176288
256	64	0,039063	0,46875	0,118209932	3,694851708
512	256	0,019531	0,117188	0,030155609	3,919998145
1024	1024	0,009766	0,029297	0,007574915	3,980982939
2048	4096	0,004883	0,007324	0,001894246	3,998907504
4096	16384	0,002441	0,001831	0,000471869	4,014347534

ЗАКЛЮЧЕНИЕ

В результате выполнения работы осуществлена постановка краевой задачи теплопроводности для тонкого однородного стержня, теплоизолированного с одного из концов. Для численного решения краевой задачи построены две неявные разностные схемы: простейшая и повышенного порядка.

Проведено теоретическое исследование разностных схем, в результате которого установлено, что простейшая схема аппроксимирует исходную краевую задачи с первым порядком по временному шагу h_t и с первым порядком по пространственному шагу h_x , а схема повышенного порядка аппроксимирует исходную краевую задачи с первым порядком по временному шагу h_t и со вторым порядком по пространственному шагу h_x . Также в результате исследования установлено, что обе схемы являются безусловно устойчивыми.

Для проверки фактической сходимости разностной схемы был использован тестовый пример. Высокоточное решение тестовой задачи получено методом разделения переменных в виде ряда Фурье. Для контроля погрешности аналитического решения использована оценка остатка ряда.

В результате серии вычислительных экспериментов установлено, что фактическая погрешность разностного решения, вычисляемого с помощью схемы повышенного порядка, убывает с измельчением шагов сетки. Простейшая неявная схема является непрогнозируемой, так как вычислительная погрешность не обладает свойством сходимости, что можно наблюдать в результатах исследования погрешности с помощью вычислительного эксперимента в таблице 2. При этом порядки убывания погрешности численного решения отличались от порядков, полученных при теоретическом исследовании.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тихонов А.Н., Самарский А.А. Уравнения математической физики: Учебное пособие для университетов. - М.: Наука, 1972. – 736 с.
2. Владимиров В.С. Уравнения математической физики: Учеб. Для вузов / В. С. Владимиров, В. В. Жаринов. - М.: Физико-математическая литература, 2000. - 399 с.
3. Степанов В.В. Курс дифференциальных уравнений – 8-е изд. - М.: ГИФМЛ, 1959.
4. Бахвалов Н. С. Численные методы: [учеб. пособие для физ.-мат. специальностей вузов] / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков; Моск. гос. ун-т им. М. В. Ломоносова. - 5-е изд. - М.: Бином. Лаб. знаний, 2007. - 636 с.
5. Ильин В.А., Садовничий В.А., Сендов Бл.Х. Математический анализ. – 2-е изд. – М.: Проспект, 2006. – 368 с.

ПРИЛОЖЕНИЕ А Исходный код программы

```
CalculatorData.cs
using System;
using System.ComponentModel;
using System.Runtime.CompilerServices;
using System.Threading.Tasks;
using MathPhysics.Annotations;
namespace MathPhysics
{
    public class CalculatorData : INotifyPropertyChanged
    {
        //l
        public double Length { get; set; }
        //S
        public double Square { get; set; }
        //T
        public double TimeIntervalLength
        {
            get => _timeIntervalLength;
            set
            {
                _timeIntervalLength = value;
                if (Time > _timeIntervalLength)
                    Time = _timeIntervalLength;
                OnPropertyChanged(nameof(TimeIntervalLength));
            }
        }
        //alpha
        public double ThermalExchange { get; set; }
        //k
        public double ThermalConductivity { get; set; }
        //c
        public double ThermalCapacity { get; set; }
        //u_0
        public double StartTemperature { get; set; }
        public long I { get; set; } = 1000;
        public long K { get; set; } = 1000;

        //mu_n(t), NOT MU(X)
        public Func<double, int, double> ThermalSourcesIntensity => (t, n) =>
        {
            double result = 0;
            double mu = _muValues[n];
            double numerator = 4 * mu * Math.PI * (1 + Math.Cos(mu));
            double denominator = (2 * mu + Math.Sin(2*mu)) * (Math.Pow(Math.PI, 2) -
Math.Pow(mu, 2));
            result = numerator / denominator;
            return result;
        };

        public double[] MuValues => _equationSolver.GetRoots(0, Number, Math.PI/2 +
0.0001, Math.PI, 0.001 * _accuracy, x => Math.Tan(x) -
(ThermalExchange * Length/ThermalConductivity)/x);

        public long Number
        {
            get => _number;
            set => _number = value;
        }

        public double Accuracy
        {
            get => _accuracy;
            set => _accuracy = value;
        }

        public double LeftBorder => 0;
        public double RightBorder => Length;
        public double Step { get; set; }

        public double Time { get; set; }

        public double Eps1 { get; set; }
        public double Eps2 { get; set; }
    }
}
```

```

public double[] PrepareEigenvalues()
{
    _muValues = MuValues;
    return _muValues;
}

public void Init(TemperatureCalculator _calculator)
{
    _parent = _calculator;
}

private readonly EquationSolver _equationSolver = new EquationSolver();
private double[] _muValues;
private long _number, _maxNumber = long.MaxValue;
private double _accuracy, _timeIntervalLength;
private TemperatureCalculator _parent;
private double GetEps(long n)
{
    var b = Math.Sqrt(Square);
    var multiplierA = 2 * ThermalCapacity * Length / Math.Pow(Math.PI, 2) *
        Math.Sqrt(b / (ThermalExchange * ThermalConductivity));
    var multiplierB = Math.Sqrt(b * ThermalConductivity / ThermalExchange) *
        Math.PI / (2 * Length);

    var numerator = multiplierA * Math.Atan(multiplierB * n);
    var denominator = Math.Pow(n, 2) - 1;
    return numerator / denominator;
}

public void CalculateAccuracy()
{
    _accuracy = GetEps(Number);
    OnPropertyChanged(nameof(Accuracy));
}

public async Task<long> CalculateNumber()
{
    if (_accuracy < 0.00001)
        CalculateNumberByEquationRoots();
    else
        CalculateNumberBruteForce();

    _number = await _parent.GetNumberByAccuracyAsync(_accuracy);

    OnPropertyChanged(nameof(Number));
    return _number;
}

private void CalculateNumberBruteForce()
{
    for (int n = 1; n <= _maxNumber; n++)
    {
        var eps = GetEps(n);
        if (eps < _accuracy)
        {
            _number = n;
            break;
        }
    }
}

private void CalculateNumberByEquationRoots()
{
    var b = Math.Sqrt(Square);
    Func<double, double> func = x =>
    {
        var multiplierA = 2 * ThermalCapacity * Length / Math.Pow(Math.PI, 2) *
            Math.Sqrt(b / (ThermalExchange * ThermalConductivity));
        var multiplierB = Math.Sqrt(b * ThermalConductivity / ThermalExchange) *
            Math.PI / (2 * Length);

        return multiplierA * Math.Atan(multiplierB * x) - _accuracy *
            (Math.Pow(x, 2) - 1);
    };
}

```

```

        _number = (long)_equationSolver.GetRoots(1, 1, _maxNumber, _maxNumber, 0.5,
func)[0] + 1;
    }

    public event PropertyChangedEventHandler PropertyChanged;
    [NotifyPropertyChangedInvocator]
    protected virtual void OnPropertyChanged([CallerMemberName] string propertyName
= null)
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }
}

TemperatureCalculator.cs
using System;
using System.IO;
using System.Text;
using System.Threading.Tasks;
using OxyPlot;
using OxyPlot.Series;
using Color = System.Drawing.Color;

namespace MathPhysics
{
    public class TemperatureCalculator
    {
        private static CalculatorData _data;
        private static double[] _muValues;

        //x,t -> u
        private Func<double, double, double> temperatureFunc = (x, t) =>
        {
            double sum = 0;
            double b = Math.Sqrt(_data.Square);
            double l2 = Math.Pow(_data.Length, 2);
            for (int n = 0; n < _data.Number; n++)
            {
                var denominator = (4 * _data.ThermalExchange * l2 +
Math.Pow(_muValues[n], 2)) /
                                b * _data.ThermalConductivity *
                                (b * _data.ThermalCapacity * l2);
                var numerator = _data.ThermalSourcesIntensity(t, n) *
Math.Cos(_muValues[n] * x / _data.Length) *
                                (1 - Math.Exp(-denominator * t));
                sum += numerator / denominator;
            }
            return sum + _data.StartTemperature;
        };

        private LineSeries DoCalculate()
        {
            _muValues = _data.PrepareEigenvalues();

            if (_data.Number > _muValues.Length)
                _data.Number = _muValues.Length - 1;

            var series = new LineSeries() { Color = OxyColor.FromRgb(136, 136, 136) };

            for (double x = _data.LeftBorder; x <= _data.RightBorder; x += _data.Step)
            {
                var u = temperatureFunc(x, _data.Time);
                series.Points.Add(new DataPoint(x, u));
            }

            return series;
        }

        private LineSeries GetSeries(double[,] matrix, Color color, long I = 1000, long
K = 1000)
        {
            double hX = _data.Length / I, hT = _data.TimeIntervalLength / K;
            int k = (int)Math.Round(_data.Time / hT);
            var series = new LineSeries() { Color = OxyColor.FromRgb(color.R, color.G,
color.B) };

```



```

        for (int i = 0; i <= I; i++)
        {
            double x = i * hX;
            double u = _data.StartTemperature + matrix[i, k];
            series.Points.Add(new DataPoint(x, u));
        }
        return series;
    }

private double[,] DoFirstCalculate3(long I = 1000, long K = 1000)
{
    double b = Math.Sqrt(_data.Square);
    double[,] result = new double[I + 1, K + 1];
    double hX = _data.Length / I, hT = _data.TimeIntervalLength / K;
    double hX2 = Math.Pow(hX, 2);

    double firstGamma = 4 * _data.ThermalExchange * hT / (_data.ThermalCapacity
* b) +
    2 * _data.ThermalConductivity * hT /
(_data.ThermalCapacity * hX2);
    double secondGamma = _data.ThermalConductivity * hT / (_data.ThermalCapacity
* hX2);

    for (int i = 0; i <= I; i++)
    {
        result[i, 0] = 0;
    }

    for (int k = 1; k <= K; k++)
    {
        double[] A = new double[I + 1];
        double[] B = new double[I + 1];
        double[] C = new double[I + 1];
        double[] F = new double[I + 1];

        B[0] = 1 + firstGamma - secondGamma;
        F[0] = result[0, k - 1];

        for (int i = 1; i <= I; i++)
        {
            A[i] = -secondGamma;
            B[i] = 1 + firstGamma;
            C[i - 1] = -secondGamma;
            F[i] = Math.Sin(Math.PI * (hX * i) / _data.Length) * hT + result[i,
k - 1];
        }

        B[I] = 1 + firstGamma - (1 + _data.ThermalExchange * hX /
_data.ThermalConductivity) * secondGamma;
        double[] alphas = new double[I + 1], betas = new double[I + 1];
        alphas[1] = -C[0] / B[0];
        betas[1] = F[0] / B[0];
        for (int i = 2; i <= I; i++)
        {
            alphas[i] = -C[i - 1] / (alphas[i - 1] * A[i - 1] + B[i - 1]);
            betas[i] = (F[i - 1] - A[i - 1] * betas[i - 1]) / (alphas[i - 1] *
A[i - 1] + B[i - 1]);
        }

        result[I - 1, k] = (F[I - 1] + secondGamma * betas[I - 1]) /
(1 + firstGamma -
secondGamma / (1 + _data.ThermalExchange * hX /
_data.ThermalConductivity) -
secondGamma * alphas[I - 1]);
        result[I, k] = result[I - 1, k] / (1 + _data.ThermalExchange * hX /
_data.ThermalConductivity);

        for (long i = I - 2; i >= 0; i--)
        {
            result[i, k] = alphas[i + 1] * result[i + 1, k] + betas[i + 1];
        }
        result[0, k] = result[1, k];
    }
    return result;
}

private double[,] DoSecondCalculate3(long I = 1000, long K = 1000)

```

```

    {
        double b = Math.Sqrt(_data.Square);
        double[,] result = new double[I + 1, K + 1];
        double hX = _data.Length / I, hT = _data.TimeIntervalLength / K;
        double hX2 = Math.Pow(hX, 2);

        double firstGamma = 4 * _data.ThermalExchange * hT / (_data.ThermalCapacity
* b) +
                                2 * _data.ThermalConductivity * hT /
(_data.ThermalCapacity * hX2);
        double secondGamma = _data.ThermalConductivity * hT / (_data.ThermalCapacity
* hX2);

        //double A = -secondGamma, B = 1 + firstGamma, C = -secondGamma;

        for (int i = 0; i <= I; i++)
        {
            result[i, 0] = 0;
        }

        for (int k = 1; k <= K; k++)
        {
            double[] A = new double[I + 1];
            double[] B = new double[I + 1];
            double[] C = new double[I + 1];
            double[] F = new double[I + 1];
            B[0] = 1 + firstGamma - secondGamma;
            F[0] = result[0, k - 1];
            for (int i = 1; i <= I; i++)
            {
                A[i] = -secondGamma;
                B[i] = 1 + firstGamma;
                C[i - 1] = -secondGamma;
                F[i] = Math.Sin(Math.PI * (hX * i) / _data.Length) * hT + result[i,
k - 1];
            }

            A[I] = -(secondGamma + (1 + firstGamma) *
_data.ThermalConductivity / (2 * hX * _data.ThermalExchange));
            B[I] = 0;

            double[] alphas = new double[I + 1], betas = new double[I + 1];
            alphas[1] = -C[0] / B[0];
            betas[1] = F[0] / B[0];
            for (int i = 2; i <= I; i++)
            {
                alphas[i] = -C[i - 1] / (alphas[i - 1] * A[i - 1] + B[i - 1]);
                betas[i] = (F[i - 1] - A[i - 1] * betas[i - 1]) / (alphas[i - 1] *
A[i - 1] + B[i - 1]);
            }

            result[I, k] = (F[I - 1] - betas[I] * (A[I - 1] + C[I - 1])) /
(alphas[I] * (A[I - 1] + C[I - 1]) + B[I - 1] - 2 * C[I
- 1] * _data.ThermalExchange *
                                hX / _data.ThermalConductivity);

            for (long i = I - 1; i >= 0; i--)
            {
                result[i, k] = alphas[i + 1] * result[i + 1, k] + betas[i + 1];
            }
        }
        return result;
    }

//x -> u
public Series[] Calculate(CalculatorData data)
{
    _data = data;
    _data.Init(this);
    long I = _data.I, K = _data.K;
    double[,] matrixB = DoFirstCalculate3(I, K);
    double[,] matrixC = DoSecondCalculate3(I, K);
    var result = new[]
    {
        DoCalculate(),
        GetSeries(matrixB, Color.Blue, I, K),
    }
}

```

```

        GetSeries(matrixC, Color.Green, I, K)
    };
    data.Eps1 = GetFault(result[0], result[1]);
    data.Eps2 = GetFault(result[0], result[2]);

    return result;
}

public double GetFault(LineSeries s1, LineSeries s2)
{
    double eps = double.MinValue;
    int shift = s1.Points.Count / 4;
    for (int i = shift; i < s1.Points.Count - shift; i++)
    {
        double diff = Math.Abs(s1.Points[i].Y - s2.Points[i].Y);
        if (diff > eps)
            eps = diff;
    }
    return eps;
}

public void WriteMatrixToCsv(double[,] matrix, int I, int K)
{
    var stringBuilder = new StringBuilder();
    for (int k = 0; k <= K; k++)
    {
        for (int i = 0; i < I; i++)
        {
            stringBuilder.Append($"{matrix[i, k]};");
        }
        stringBuilder.Append($"{matrix[I, k]}\n");
    }
    File.WriteAllText($"{matrix}_{I}_{K}_{new Random(DateTime.Now.Millisecond + I
* K + stringBuilder.Length).Next()}.csv", stringBuilder.ToString());
}

public async Task<Series[]> CalculateAsync(CalculatorData data)
{
    return await Task.Run(() => Calculate(data));
}

public long GetNumberByAccuracy(double eps)
{
    double value, prevValue = double.NaN;

    _muValues = _data.PrepareEigenvalues();

    double x = _data.Length / 2;
    while (true)
    {
        value = temperatureFunc(x, _data.Time);
        if (double.IsNaN(prevValue))
        {
            prevValue = value;
            _data.Number--;
            continue;
        }
        double accuracy = Math.Abs(value - prevValue);
        if (accuracy > eps)
            break;
        _data.Number--;
    }
    return ++_data.Number;
}

public async Task<long> GetNumberByAccuracyAsync(double eps)
{
    return await Task.Run(() => GetNumberByAccuracy(eps));
}
}

}
MainWindow.cs
<Window x:Class="MathPhysics.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:oxy="http://oxyplot.org/wpf"

```

```

        xmlns:gif="http://wpfanimatedgif.codeplex.com"
        xmlns:mathPhysics="clr-namespace:MathPhysics"
        mc:Ignorable="d"
        Title="MainWindow" Height="720" Width="1280">
<Window.DataContext>
    <mathPhysics:CalculatorData />
</Window.DataContext>
<Grid Name="MainGrid" Margin="0,160,0,0">
    <oxy:PlotView x:Name="PlotView" Margin="0,0,0,70"/>
    <TextBox Name="Length" HorizontalAlignment="Left" Height="32" Margin="15,-
122,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15" Text="{Binding
Length}"/>
    <TextBox Name="Square" HorizontalAlignment="Left" Height="32" Margin="15,-
54,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15" Text="{Binding
Square}"/>
    <TextBox Name="TimeInterval" HorizontalAlignment="Left" Height="32"
Margin="215,-122,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15"
Text="{Binding TimeIntervalLength, Mode=TwoWay}"/>
    <TextBox Name="ThermalExchange" HorizontalAlignment="Left" Height="32"
Margin="215,-54,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15"
Text="{Binding ThermalExchange}"/>
    <TextBox Name="ThermalConductivity" HorizontalAlignment="Left" Height="32"
Margin="415,-122,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15"
Text="{Binding ThermalConductivity}"/>
    <TextBox Name="ThermalCapacity" HorizontalAlignment="Left" Height="32"
Margin="415,-54,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15"
Text="{Binding ThermalCapacity}"/>
    <TextBox Name="StartTemperature" HorizontalAlignment="Left" Height="32"
Margin="615,-122,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15"
Text="{Binding StartTemperature}"/>
    <TextBox x:Name="XStep" HorizontalAlignment="Left" Height="32" Margin="815,-
122,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15" Text="{Binding
Step}"/>
    <TextBox x:Name="Number" HorizontalAlignment="Left" Height="32" Margin="615,-
54,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15" Text="{Binding
Number, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"/>
    <TextBox x:Name="Accuracy" HorizontalAlignment="Left" Height="32" Margin="815,-
54,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="15" Text="{Binding
Accuracy, Mode=TwoWay, StringFormat=G4}"/>
    <TextBox x:Name="I" HorizontalAlignment="Left" Height="32" Margin="971,-54,0,0"
TextWrapping="Wrap" VerticalAlignment="Top" Width="79" FontSize="15" Text="{Binding I,
Mode=TwoWay, StringFormat=G4}"/>
    <TextBox x:Name="K" HorizontalAlignment="Left" Height="32" Margin="1059,-
54,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="79" FontSize="15" Text="{Binding K,
Mode=TwoWay, StringFormat=G4}"/>
    <Button Content="Построить график" HorizontalAlignment="Left" Margin="1073,-
128,0,0" VerticalAlignment="Top" Width="140" Height="40" Click="ButtonBase_OnClick"/>

    <Grid Margin="15,0,0,0" Height="65" VerticalAlignment="Bottom">
        <Slider Name="TimeSlider" Margin="42,0,42,25" Height="35"
VerticalAlignment="Bottom" Value="{Binding Time}" Maximum="{Binding TimeIntervalLength}"
ValueChanged="RangeBase_OnValueChanged"/>
        <Label Content="0" HorizontalAlignment="Left" Margin="0,0,0,35" Height="25"
VerticalAlignment="Bottom"/>
        <Label Content="{Binding TimeIntervalLength}" Margin="0,0,0,35" Height="25"
VerticalAlignment="Bottom" HorizontalAlignment="Right"/>
        <Label Name="CurrentTimeLabel" Content="" Margin="0,25,0,0" FontSize="16"
HorizontalAlignment="Center" Width="150" Height="36" VerticalAlignment="Top"/>
    </Grid>
    <Label Content="Длина стержня" HorizontalAlignment="Left" Margin="15,-148,0,0"
VerticalAlignment="Top"/>
    <Label Content="Площадь сечения" HorizontalAlignment="Left" Margin="15,-80,0,0"
VerticalAlignment="Top"/>
    <Label Content="Длина временного промежутка" HorizontalAlignment="Left"
Margin="215,-144,0,0" VerticalAlignment="Top" FontSize="9"/>
    <Label Content="Коэффициент теплообмена" HorizontalAlignment="Left"
Margin="215,-77,0,0" VerticalAlignment="Top" FontSize="10"/>
    <Label Content="Коэффициент теплопроводности" HorizontalAlignment="Left"
Margin="415,-144,0,0" VerticalAlignment="Top" FontSize="9"/>
    <Label Content="Коэффициент объемной" HorizontalAlignment="Left"
VerticalAlignment="Top" FontSize="10" Margin="416,-83,0,0" Width="130"/>
    <Label Content="теплоемкости" HorizontalAlignment="Left" Margin="416,-72,0,0"
VerticalAlignment="Top" FontSize="10"/>
    <Label Content="Температура среды" HorizontalAlignment="Left" Margin="615,-
148,0,0" VerticalAlignment="Top"/>

```

```

        <Label Content="Длина последовательности" HorizontalAlignment="Left"
Margin="615,-77,0,0" VerticalAlignment="Top" FontSize="10"/>
        <Label Content="Шаг по X" HorizontalAlignment="Left" Margin="815,-148,0,0"
VerticalAlignment="Top"/>
        <Label Content="Порешность" HorizontalAlignment="Left" Margin="815,-79,0,0"
VerticalAlignment="Top"/>
        <Image gif:ImageBehavior.AnimatedSource="Images/Rolling.gif" Name="Spinner"
HorizontalAlignment="Left" Height="40" Margin="1163,-79,0,0" VerticalAlignment="Top" Width="40"
Visibility="Hidden"/>
        <Button Name="FromNumberToAccuracy" Content="-&gt;" HorizontalAlignment="Left"
Margin="760,-37,0,0" VerticalAlignment="Top" Width="50" Height="15" FontSize="9"
Click="FromNumberToAccuracy_Click"/>
        <Button Name="FromAccuracyToNumber" Content="&lt;-" HorizontalAlignment="Left"
Margin="760,-54,0,0" VerticalAlignment="Top" Width="50" Height="15" FontSize="9"
Click="FromAccuracyToNumber_Click"/>
        <Label Content="I" HorizontalAlignment="Left" Margin="984,-80,0,0"
VerticalAlignment="Top" FontFamily="Sitka Small" FontSize="14" Height="25"/>
        <Label Content="K" HorizontalAlignment="Left" Margin="1060,-80,0,0"
VerticalAlignment="Top" FontFamily="Sitka Small" FontSize="14" Height="25"/>
        <Label Name="Eps1" Content="ε1=" HorizontalAlignment="Left" Margin="1143,-
78,0,0" VerticalAlignment="Top"/>
        <Label Name="Eps2" Content="ε2=" HorizontalAlignment="Left" Margin="1143,-
43,0,0" VerticalAlignment="Top"/>
    </Grid>
</Window>
EquationSolver.cs
using System;
namespace MathPhysics
{
    public class EquationSolver
    {
        private double _eps;
        private Func<double, double> _func;
        public double[] GetRoots(double leftX, long count, double intervalLength, double
iterationStep,
            double eps, Func<double, double> f)
        {
            _eps = eps;
            _func = f;
            double[] roots = new double[count];
            for (int n = 0; n < count; n++)
            {
                double x = leftX + n * iterationStep;
                double root = GetRoot(x, x + intervalLength);
                roots[n] = root;
            }
            return roots;
        }

        //bisection method
        private double GetRoot(double leftX, double rightX)
        {
            double center = 0;
            while (rightX - leftX > 2 * _eps)
            {
                center = (leftX + rightX) / 2;
                if (_func(leftX) * _func(center) < 0)
                    rightX = center;
                else
                    leftX = center;
            }
            return center;
        }
    }
}

```