

**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 «Уравнения математической физики»

**Тема: «АНАЛИТИЧЕСКОЕ РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ
МАТЕМАТИЧЕСКОЙ ФИЗИКИ»**

Вариант № 7

Выполнили: Логинов П.В.

Лаптев А.В.

Группа: 6409

Проверил: Дегтярёв А.А.

Оценка: _____

Дата: _____

Самара 2017

ЗАДАНИЕ

1. Осуществить математическую постановку краевой задачи для физического процесса, описанного в предложенном варианте курсовой работы.
2. Используя метод разделения переменных (метод Фурье) получить решение задачи в виде ряда Фурье.
3. Исследовать сходимость ряда, получить оценку остатка ряда.
4. Разработать компьютерную программу расчета решения задачи (суммирования ряда Фурье) с требуемой точностью. При расчете коэффициентов ряда использовать метод численного интегрирования, если это необходимо. Обеспечить контроль погрешности численного интегрирования. Если необходимо, то разработать специальный программный модуль для расчета используемых собственных чисел оператора Лапласа. Обеспечить контроль погрешности расчета собственных чисел.
5. Используя разработанную программу, провести экспериментальное исследование качества полученной аналитической оценки остатка ряда.
6. Оформить пояснительную записку к курсовой работе.

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 моделирования процесса теплопроводности в стержне использовать представление решения описанной задачи математической физики в виде ряда Фурье по собственным функциям оператора Лапласа, удовлетворяющим краевым условиям задачи.

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

Расчетно-пояснительная записка: 34 с., 4 рисунка, 4 таблицы, 5 источников, 2 приложения.

УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ, КРАЕВАЯ ЗАДАЧА, УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ, РЯД ФУРЬЕ, ЗАДАЧА ШТУРМА-ЛИУВИЛЛЯ, ИССЛЕДОВАНИЕ СХОДИМОСТИ, ОЦЕНКА ОСТАТКА РЯДА.

Целью данной работы является создание компьютерной программы моделирования теплового процесса в тонком однородном стержне длиной l , сечением s на временном промежутке $0 < t \leq T$. Для достижения цели строится математическая модель процесса, в результате чего получается дифференциальное уравнение в частных производных параболического типа. Для решения краевой задачи используется метод Фурье разделения переменных. Решение получено в виде разложения в ряд Фурье по косинусам. В работе приведены результаты исследования сходимости ряда и исследования качества полученной аналитической оценки остатка ряда.

Программа написана на языке C# 6 в среде разработки Visual Studio 2017, операционная система Windows 10 Professional.

СОДЕРЖАНИЕ

Введение	6
1 Математическая постановка задачи	7
2 Решение задачи методом разделения переменных	12
3 Исследование сходимости и оценка остатка ряда	18
4 Исследование качества аналитической оценки остатка ряда	20
Заключение	22
Список использованных источников	23
Приложение А Исходный код программы.....	24
Приложение Б Получение собственных значений задачи Штурма- Лиувилля	33

ВВЕДЕНИЕ

Задачи математической физики решаются разными методами: аналитическими, приближенно–аналитическими, численными, вероятностными. В данной работе применяется метод разделения переменных Фурье, суть которого состоит в представлении решения бесконечным функциональным рядом Фурье.

В первой главе идет изложение процесса получения математической модели рассматриваемого теплового процесса.

Во второй главе приведено решение задачи методом разделения переменных.

В третьей главе рассмотрен вопрос получения оценки остатка ряда.

В четвертой главе изложены результаты исследования качества полученной оценки остатка ряда.

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:

$$q = -k \frac{\partial u}{\partial n}.$$

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].

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

Найдем среднее распределение тепла по сечению стержня:

$$v = \frac{1}{S} \iint_S u dy dz.$$

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

$$b = \sqrt{S}.$$

Схематичное представление стержня изображено на рисунке 1.

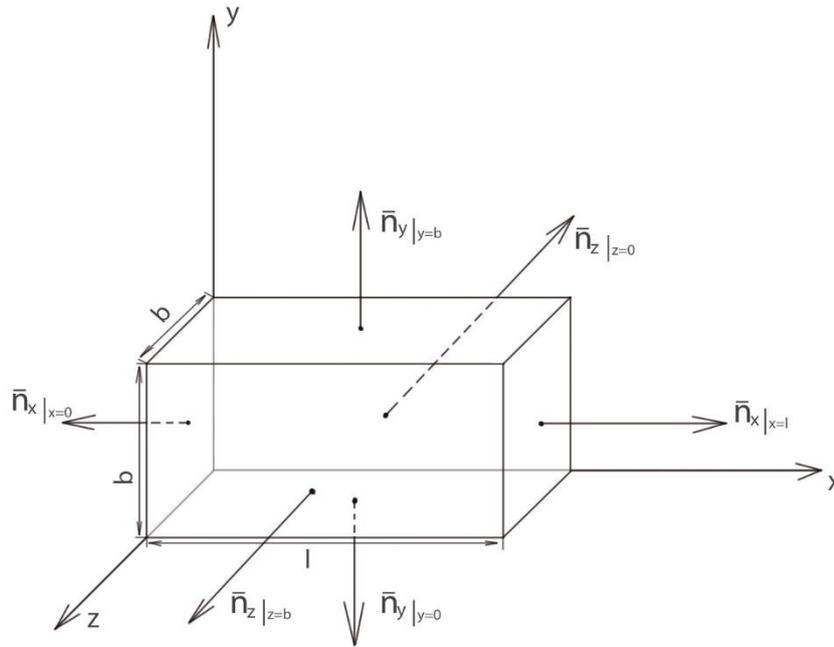


Рисунок 1 – Графическое представление тонкого стержня

Выразим частные производные для подстановки в исходное уравнение:

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial t} \left[\frac{1}{S} \iint_S u dy dz \right] = \frac{1}{S} \iint_S \frac{\partial u}{\partial t} dy dz; \quad \frac{\partial^2 v}{\partial x^2} = \frac{1}{S} \iint_S \frac{\partial^2 u}{\partial x^2} dy dz.$$

Поскольку теплообмен на боковой поверхности осуществляется по закону Ньютона, то тепловой поток q будет определяться на границах следующим образом:

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

Продифференцируем исходную функцию по нормали на грани $y = 0$:

$$\begin{aligned} \left. \frac{\partial u}{\partial \bar{n}_y} \right|_{y=0} &= \left(\frac{\partial u}{\partial x} \cos(\bar{n}_y, \bar{e}_x) + \frac{\partial u}{\partial y} \cos(\bar{n}_y, \bar{e}_y) + \frac{\partial u}{\partial z} \cos(\bar{n}_y, \bar{e}_z) \right) \Big|_{y=0} = \\ &= -\frac{\partial v}{\partial y}. \end{aligned}$$

Выразим значение теплового потока на нуле по оси y :

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

Из этого следует, что частные производные будут такими:

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

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

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

Аналогично выразим вторую частную производную для z :

$$\begin{aligned} \frac{\partial^2 v}{\partial z^2} &= \frac{2\alpha}{kS} \left[u_0 - \frac{1}{b} \int_0^b u dy \right], \\ v &= \frac{1}{b} \int_0^b u dy = \frac{1}{b} \int_0^b u dz. \end{aligned}$$

Подставим результат в формулы вторых частных производных:

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

В итоге получена математическая модель:

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

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

выглядит так:

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

Для упрощения вычислений проведем замену $w = \omega e^{\frac{4\alpha}{bc}t}$. Выведем

частные производные для подстановки в уравнение:

$$\frac{\partial w}{\partial t} = \frac{\partial \omega}{\partial t} e^{\frac{4\alpha}{bc}t} + \frac{4\alpha}{bc} \omega e^{\frac{4\alpha}{bc}t} = \frac{\partial \omega}{\partial t} e^{\frac{4\alpha}{bc}t} + \frac{4\alpha}{bc} w;$$

$$\frac{\partial^2 w}{\partial x^2} = \frac{\partial^2 \omega}{\partial x^2} e^{\frac{4\alpha}{bc}t}.$$

Умножим уравнение на $e^{\frac{4\alpha}{bc}t}$. После проведения замены итоговая математическая модель задачи выглядит так:

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

2 РЕШЕНИЕ ЗАДАЧИ МЕТОДОМ РАЗДЕЛЕНИЯ ПЕРЕМЕННЫХ

Разложим функцию $\varphi(x)$ в ряд по собственным функциям:

$$\varphi(x) = \sum_{n=0}^{\infty} \varphi_n X_n(x),$$

где $X_n(x)$ – собственные векторы системы Штурма-Лиувилля, возникающей при решении соответствующей исходной однородной задачи методом разделения переменных.

Аналогично разложим в ряд функцию $w(x, t)$:

$$w(x, t) = \sum_{n=0}^{\infty} A_n(t) X_n(x).$$

Решим методом Фурье однородную задачу:

$$\left\{ \begin{array}{l} \frac{\partial w}{\partial t} = \frac{k}{c} \frac{\partial^2 w}{\partial x^2}, \quad x \in [0, l], \quad t \in [0, T]; \\ w|_{t=0} = u_0, \quad x \in [0, l]; \\ \frac{\partial w}{\partial x}|_{x=0} = 0, \quad t \in [0, T]; \\ \frac{\partial w}{\partial x}|_{x=l} = -\frac{\alpha}{k} w|_{x=l}, \quad t \in [0, T]. \end{array} \right.$$

Для этого представим функцию $W(x, t)$ в виде произведения двух функций:

$$W(x, t) = X(x)A(t).$$

Тогда уравнение системы примет вид:

$$XA' = \frac{k}{c} X'' AS.$$

Разделяя переменные, получим:

$$\frac{c}{k} \frac{A'}{A} = \frac{X''}{X} = -\lambda^2.$$

где $\lambda^2 = const$, так как левая часть зависит только от t , а правая – от x . Из этого следует, что:

$$X'' + \lambda^2 X = 0.$$

Получена задача Штурма-Лиувилля [2]:

$$\begin{cases} X'' + \lambda^2 X = 0, \\ X'(0) = 0, \\ X'(l) = -\frac{\alpha}{k} X(l). \end{cases}$$

Общий вид решения задачи выглядит так [3]:

$$X(x) = C_1 \sin(\lambda x) + C_2 \cos(\lambda x), \text{ где } C_1 \text{ и } C_2 - \text{ константы.}$$

Подставим краевые условия в решение и получим:

$$X'(0) = \lambda C_1 \cos 0 - \lambda C_2 \sin 0 = 0.$$

Из этого условия выходит, что $C_1 = 0$. Подставим это во второе условие и получим:

$$X'(l) = -\frac{\alpha}{k} X = \lambda C_2 \sin(\lambda l) = \frac{\alpha}{k} C_2 \cos(\lambda l),$$

где $C_2 \neq 0$, так как нас не интересуют тривиальные решения $X = 0$.

Продолжим решение с учетом этого факта:

$$X'(l) = \frac{\lambda k \sin(\lambda l)}{\alpha \cos(\lambda l)} = 1.$$

Очевидно, что $\frac{\sin(\lambda l)}{\cos(\lambda l)} = \operatorname{tg}(\lambda l)$. Подставим в выражение и получим:

$$\operatorname{tg}(\lambda l) = \frac{\alpha}{\lambda k}.$$

Для удобства сделаем замены $\mu = \lambda l$ и $\eta = \frac{\alpha l}{k}$: $\operatorname{tg}(\mu) = \frac{\eta}{\mu}$.

Решим это уравнение с помощью метода половинного деления [4].

Результаты численного решения представлены в приложении Б.

Выразим ряд $X_n(x)$:

$$X_n(x) = \cos(\lambda_n x) = \cos\left(\frac{\mu_n x}{l}\right).$$

Подставим $X_n(x)$ в ряды $w(x, t)$ и $\varphi(x)$:

$$w(x, t) = \sum_{n=0}^{\infty} A_n(t) \cos\left(\frac{\mu_n x}{l}\right),$$

$$\varphi(x) = \sum_{n=0}^{\infty} \varphi_n \cos\left(\frac{\mu_n x}{l}\right).$$

Для нахождения коэффициентов A_n подставим выраженные выше ряды в исходную задачу:

$$\begin{aligned} \sum_{n=0}^{\infty} \dot{A}_n(t) \cos\left(\frac{\mu_n x}{l}\right) &= \\ &= -\frac{k}{c} \sum_{n=0}^{\infty} A_n(t) \left(\frac{\mu_n}{l}\right)^2 \cos\left(\frac{\mu_n x}{l}\right) + \sum_{n=0}^{\infty} \varphi_n \cos\left(\frac{\mu_n x}{l}\right) e^{\frac{4\alpha}{bc}} \Rightarrow \\ \Rightarrow \sum_{n=0}^{\infty} \cos\left(\frac{\mu_n x}{l}\right) \left[\dot{A}_n(t) \cos + A_n(t) \left(\frac{\mu_n}{l}\right)^2 - \varphi_n e^{\frac{4\alpha}{bc}} \right] &= 0 \Rightarrow \\ \Rightarrow \begin{cases} \dot{A}_n(t) + \frac{k\mu_n^2}{cl^2} A_n(t) = \varphi_n e^{\frac{4\alpha}{bc}}, \\ A_n(0) = 0. \end{cases} \end{aligned}$$

Запишем общий вид решения задачи:

$$A_n(t) = \gamma_n e^{-\frac{k\mu_n^2}{cl^2} t}.$$

Подставим выражение в уравнение:

$$\dot{\gamma}_n e^{-\frac{k\mu_n^2}{cl^2} t} - \frac{k\mu_n^2}{cl^2} \gamma_n e^{-\frac{k\mu_n^2}{cl^2} t} + \frac{k\mu_n^2}{cl^2} \gamma_n e^{-\frac{k\mu_n^2}{cl^2} t} = \varphi_n e^{\frac{4\alpha}{bc}},$$

$$\dot{\gamma}_n e^{-\frac{k\mu_n^2}{cl^2} t} = \varphi_n e^{\frac{4\alpha}{bc}},$$

$$\dot{\gamma}_n = \varphi_n e^{\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right) t}.$$

Выразим γ_n :

$$\begin{aligned} \gamma_n &= \varphi_n \int_0^t e^{\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right) \tau} d\tau + B_n = \frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \left[e^{\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right) \tau} \Big|_0^t \right] + B_n = \\ &= \frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \left[e^{\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right) t} - 1 \right] + B_n. \end{aligned}$$

Подставим в начальное условие:

$$A_n(0) = \left(\frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \left[e^{\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right)0} - 1 \right] + B_n \right) e^{-\frac{k\mu_n^2}{cl^2}} = B_n.$$

Из этого следует, что $B_n = 0$. Тогда подставим в γ_n :

$$\gamma_n = \frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \left[e^{\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right)t} - 1 \right].$$

Подставим γ_n в ряд A_n :

$$\begin{aligned} A_n &= \frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \left[e^{\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right)t} - 1 \right] e^{-\frac{k\mu_n^2}{cl^2}} = \\ &= \frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \left[e^{\left(\frac{4\alpha}{bc}\right)t} - e^{-\left(\frac{k\mu_n^2}{cl^2}\right)t} \right]. \end{aligned}$$

Выразим ряд $\varphi(x)$:

$$\varphi(x) = \sum_{n=0}^{\infty} \varphi_n \cos\left(\frac{\mu_n x}{l}\right).$$

Для нахождения коэффициентов φ_n воспользуемся свойством ортогональности: умножим обе части уравнения на $\cos\left(\frac{\mu_m x}{l}\right)$ и проинтегрируем:

$$\int_0^l \varphi(x) \cos\left(\frac{\mu_m x}{l}\right) dx = \int_0^l \sum_{n=0}^{\infty} \varphi_n \cos\left(\frac{\mu_n x}{l}\right) \cos\left(\frac{\mu_m x}{l}\right) dx, \text{ где}$$

$$\int_0^l \cos\left(\frac{\mu_n x}{l}\right) \cos\left(\frac{\mu_m x}{l}\right) dx = \begin{cases} = 0, & m \neq n, \\ \neq 0, & m = n. \end{cases}$$

Решим интеграл и получим коэффициент разложения φ_n :

$$\int_0^l \cos^2\left(\frac{\mu_n x}{l}\right) dx = \left(\frac{x}{2} + \frac{l \sin \frac{2\mu_n x}{l}}{4\mu_n} \right) \Big|_0^l =$$

$$= \frac{l}{2} + \frac{l \sin \frac{2\mu_n x}{l}}{4\mu_n} = \frac{2\mu_n l + l \sin(2\mu_n)}{4\mu_n}.$$

Коэффициент разложения φ_n на данном этапе выглядит так:

$$\varphi_n = \frac{4\mu_n}{2\mu_n l + l \sin(2\mu_n)} \int_0^l \varphi(x) \cos\left(\frac{\mu_n x}{l}\right) dx.$$

Вычислим φ_n при $\varphi(x) = \sin\left(\frac{\pi x}{l}\right)$:

$$\int_0^l \sin\left(\frac{\pi x}{l}\right) \cos\left(\frac{\mu_n x}{l}\right) dx =$$

$$= \frac{1}{2} \int_0^l \left(\sin\left(\frac{(\pi - \mu_n)x}{l}\right) + \sin\left(\frac{(\pi + \mu_n)x}{l}\right) \right) dx =$$

$$= \frac{1}{2} \int_0^l \left(\sin\left(\frac{x}{l}(\pi - \mu_n)\right) + \sin\left(\frac{x}{l}(\pi + \mu_n)\right) \right) dx =$$

$$= \frac{1}{2} \left[\frac{-l}{\pi - \mu_n} \cos\left(\frac{\pi - \mu_n}{l}x\right) \Big|_0^l - \frac{l}{\pi + \mu_n} \cos\left(\frac{\pi + \mu_n}{l}x\right) \Big|_0^l \right] =$$

$$= \frac{1}{2} \left[\frac{-l}{\pi - \mu_n} \cos\left(\frac{\pi - \mu_n}{l}l\right) + \frac{l}{\pi - \mu_n} - \frac{l}{\pi + \mu_n} \cos\left(\frac{\pi + \mu_n}{l}l\right) + \frac{l}{\pi + \mu_n} \right] =$$

$$= \frac{1}{2} \left[\frac{l}{\pi - \mu_n} \left(1 - \cos\left(\frac{\pi - \mu_n}{l}l\right)\right) + \frac{l}{\pi + \mu_n} \left(1 - \cos\left(\frac{\pi + \mu_n}{l}l\right)\right) \right] =$$

$$= \frac{1}{2} \left[\frac{l}{\pi - \mu_n} (1 - \cos \pi \cos \mu_n - \sin \pi \sin \mu_n) \right.$$

$$\left. + \frac{l}{\pi + \mu_n} (1 - \cos \pi \cos \mu_n + \sin \pi \sin \mu_n) \right] =$$

$$= \frac{1}{2} \left[\frac{l}{\pi - \mu_n} (1 + \cos \mu_n) + \frac{l}{\pi + \mu_n} (1 + \cos \mu_n) \right] =$$

$$= \frac{(1 + \cos \mu_m)}{2} \left[\frac{2\pi l}{(\pi^2 n^2)} \right] = \frac{\pi l (1 + \cos \mu_n)}{\pi^2 - \mu_n^2}.$$

В итоге коэффициент разложения φ_n представим в виде:

$$\varphi_n = \frac{4\mu_n \pi (1 + \cos \mu_n)}{(2\mu_n + \sin(2\mu_n))(\pi^2 - \mu_n^2)}.$$

Сделаем обратную замену переменных:

$$w(x, t) = \frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \left[e^{\left(\frac{4\alpha}{bc}\right)t} - e^{-\left(\frac{k\mu_n^2}{cl^2}\right)t} \right] \cos\left(\frac{\mu_n x}{l}\right);$$

$$\omega(x, t) = w(x, t) e^{-\left(\frac{4\alpha}{bc}\right)t};$$

$$u(x, t) = u_0 + \omega(x, t) = u_0 + w(x, t) e^{-\left(\frac{4\alpha}{bc}\right)t}.$$

Итоговая формула расчёта температуры в тонком стержне:

$$u(x, t) = u_0 + \sum_{n=1}^{\infty} \frac{\varphi_n}{\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}} \cos\left(\frac{\mu_n x}{l}\right) \left[1 - e^{-\left(\frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}\right)t} \right].$$

3 ИССЛЕДОВАНИЕ СХОДИМОСТИ И ОЦЕНКА ОСТАТКА РЯДА

Для оценки остатка ряда воспользуемся формулой

$$|R_n(x, t)| < \Psi(n) \leq \varepsilon.$$

Введём замену $\theta_n = \frac{4\alpha}{bc} + \frac{k\mu_n^2}{cl^2}$. Учтём, что $|\cos\mu_m| \leq 1, |\sin(2\mu_n)| \leq 1, \left|\cos\left(\frac{\mu_n x}{l}\right)\right| \leq 1, |e^{-\theta_n t}| < 1$:

$$\begin{aligned} |R_n(x, t)| &= \\ &= \left| \sum_{n=N+1}^{\infty} \frac{4\mu_n \pi (1 + \cos\mu_n) \cos\left(\frac{\mu_n x}{l}\right)}{\theta_n (2\mu_n + \sin(2\mu_n)) (\pi^2 - \mu_n^2)} [1 - e^{-\theta_n t}] \right| \leq \\ &\leq \left| \sum_{n=N+1}^{\infty} \frac{4\pi}{\theta_n (\pi^2 - \mu_n^2)} \right| \end{aligned}$$

Поскольку собственные числа системы Штурма-Лиувилля повторяются с периодом π , то $\lambda_n \leq \pi n$, а значит, можно заменить μ_n на πn :

$$\begin{aligned} \left| \sum_{n=N+1}^{\infty} \frac{4\pi}{\theta_n (\pi + \mu_n) (\pi - \mu_n)} \right| &\leq \left| \sum_{n=N+1}^{\infty} \frac{4}{\theta_n \pi (1 - n^2)} \right| \leq \\ &\leq \frac{4}{\pi} \left| \sum_{n=N+1}^{\infty} \frac{1}{\left(\frac{4\alpha}{bc} + \frac{k\pi^2}{cl^2} n^2\right) (1 - n^2)} \right| = \\ &= \frac{4bcl^2}{\pi} \left| \sum_{n=N+1}^{\infty} \frac{1}{(4\alpha l^2 + bk\pi^2 n^2) (1 - n^2)} \right|. \end{aligned}$$

Легко заметить, что данная сумма легко оценивается сверху интегралом соответствующей функции [5]:

$$\begin{aligned}
 & \frac{4bcl^2}{\pi} \left| \sum_{n=N+1}^{\infty} \frac{1}{(4\alpha l^2 + bk\pi^2 n^2)(1 - n^2)} \right| \leq \\
 & \leq \frac{4bcl^2}{\pi} \left| \int_N^{\infty} \frac{dn}{(4\alpha l^2 + bk\pi^2 n^2)(1 - n^2)} \right| = \\
 & = \frac{2cl\sqrt{b}}{\pi^2 \sqrt{\alpha k} (N^2 - 1)} \operatorname{arctg} \left(\frac{\pi}{2l} \sqrt{\frac{kb}{\alpha}} N \right) < \varepsilon.
 \end{aligned}$$

4 ИССЛЕДОВАНИЕ КАЧЕСТВА АНАЛИТИЧЕСКОЙ ОЦЕНКИ ОСТАТКА РЯДА

Исследуем качество аналитической оценки остатка ряда, сравнив между собой количество членов, рассчитанное по оценке N_T , а также экспериментально полученное количество членов N_{II} , удовлетворяющее заданной точности ε . Чтобы исследовать качество оценки ряда, проведем измерение необходимого количества членов и отобразим в таблицах 1, 2, 3.

Таблица 1 – Количество членов при оценке остатка ряда для $t = 0.3$; $x = l/2$

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
N_T	17	64	209	669	2124	6724	21270
N_{II}	1	5	13	33	65	117	205

Таблица 2 – Количество членов при оценке остатка ряда для $t = 2.5$; $x = l/2$

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
N_T	17	64	209	669	2124	6724	21270
N_{II}	3	9	19	37	65	117	205

Таблица 3 – Количество членов при оценке остатка ряда для $t = 30$; $x = l/2$

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
N_T	17	64	209	669	2124	6724	21270
N_{II}	3	9	19	37	65	117	205

Видно, что ряд является равномерно сходящимся. Значения экспериментально полученной суммы для достижения заданной точности не зависят от времени.

Также приведем график полученного аналитического решения при фиксированном значении времени t . На рисунке 2 изображен график распределения температуры при $t = 2.5$.

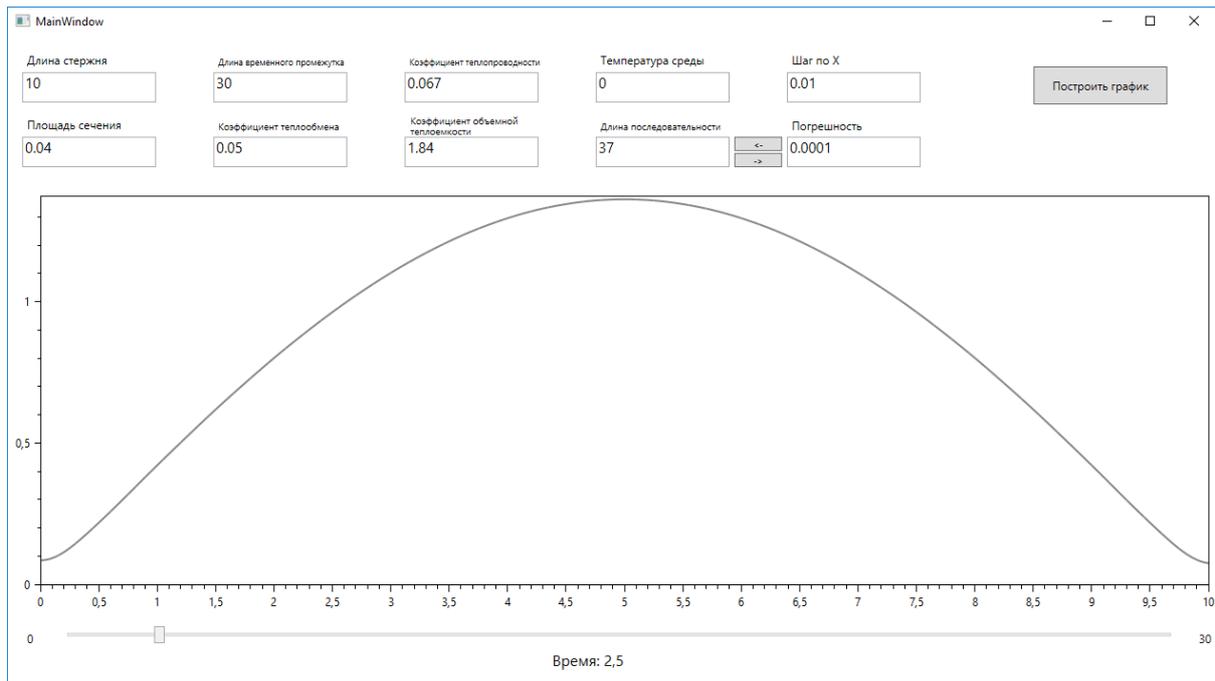


Рисунок 2 – График распределения температуры в тонком стержне при $t = 2.5$

На рисунке 3 изображен график распределения температуры при $t = 30$.

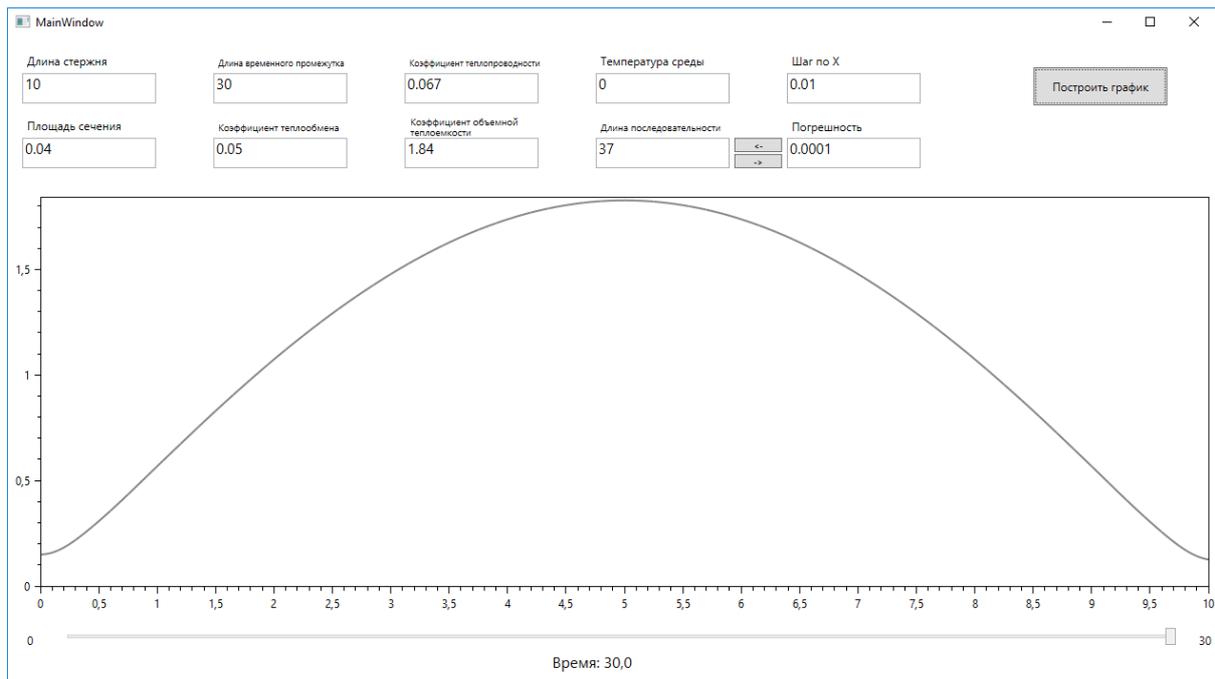


Рисунок 3 – График распределения температуры в тонком стержне при $t = 30$

ЗАКЛЮЧЕНИЕ

В работе было проведено нахождение и исследование аналитического решения задачи моделирования теплового процесса в тонком однородном стержне. Итоговое решение было получено в виде ряда Фурье. Было проведено исследование сходимости этого ряда, найдена оценка остаточного члена ряда.

Было проведено экспериментальное исследование качества оценки остатка ряда и установлено, что полученная оценка остатка ряда является грубой.

Для нашей модели была написана программа, которая отображает график распределения температуры по длине тонкого однородного стержня.

Данную программу можно использовать для моделирования теплового процесса в тонком однородном стержне, однако это потребует большой вычислительной мощности ЭВМ.

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.Diagnostics;
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; }
        //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[] 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));
    }
}

```

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;
}
}
}

```

TemperatureCalculator.cs

```

using System;
using System.Net.Sockets;
using System.Threading.Tasks;
using System.Windows.Data;
using OxyPlot;
using OxyPlot.Series;

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 +
                b * _data.ThermalConductivity *
Math.Pow(_muValues[n], 2)) /
                (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;
    }

//x -> u
public Series Calculate(CalculatorData data)
{
    _data = data;
    _data.Init(this);

    return DoCalculate();
}

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

```

using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using OxyPlot;
using OxyPlot.Axes;

namespace MathPhysics
{
    public partial class MainWindow : Window
    {
        private TemperatureCalculator _calculator;
        private CalculatorData _calculatorData;
        private string _currentTime => $"Время:
{_calculatorData.Time:F1}";

        public MainWindow()
        {
            InitValues();
            InitializeComponent();

            DataContext = _calculatorData;

            CreateSeries();
        }

        private void InitValues()
        {
            _calculator = new TemperatureCalculator();

            _calculatorData = new CalculatorData()
            {

```

```

        Length = 10,
        Square = 0.04,
        StartTemperature = 0,
        ThermalCapacity = 1.84,
        ThermalConductivity = 0.067,
        ThermalExchange = 0.05,
        TimeIntervalLength = 30,
        Step = 0.1,
        Time = 2.5,
        Number = 10
    };
    _calculatorData.CalculateAccuracy();
}

private void InitPlot()
{
    PlotView.Model = new PlotModel() { PlotType = PlotType.XY
};
}

private async void CreateSeries()
{
    InitPlot();
    await UpdateSeries();
}

private async Task UpdateSeries()
{
    if (PlotView.Model == null)
        return;

    PlotView.Model.Series.Clear();
    PlotView.Model.Axes.Clear();
    PlotView.Model.Axes.Add(new LinearAxis
    {
        Position = AxisPosition.Bottom,
        AbsoluteMinimum = 0,
        AbsoluteMaximum = _calculatorData.Length,
        Minimum = 0,
        Maximum = _calculatorData.Length
    });

    PlotView.Model.Axes.Add(new LinearAxis
    {
        Position = AxisPosition.Left,
        Minimum = 0,
        //Maximum = 2,
        AbsoluteMinimum = 0
    });

    ShowLoading();
    var series = await
_calculator.CalculateAsync(_calculatorData);
    HideLoading();
}

```

```

        PlotView.Model.Series.Add(series);
        PlotView.InvalidatePlot();

        CurrentTimeLabel.Content = _currentTime;
    }

    private async void ButtonBase_OnClick(object sender,
RoutedEventArgs e)
    {
        await UpdateSeries();
    }

    private void RangeBase_OnValueChanged(object sender,
RoutedPropertyChangedEventArgs<double> e)
    {
        CurrentTimeLabel.Content = _currentTime;
    }

    private async void FromAccuracyToNumber_Click(object sender,
RoutedEventArgs e)
    {
        ShowLoading();
        var number = await _calculatorData.CalculateNumber();
        HideLoading();
    }

    private void FromNumberToAccuracy_Click(object sender,
RoutedEventArgs e)
    {
        _calculatorData.CalculateAccuracy();
    }

    private void ShowLoading()
    {
        Spinner.Visibility = Visibility.Visible;
        MainGrid.IsEnabled = false;
    }

    private void HideLoading()
    {
        MainGrid.IsEnabled = true;
        Spinner.Visibility = Visibility.Hidden;
    }
}
}

```

ПРИЛОЖЕНИЕ Б Получение собственных значений задачи
Штурма-Лиувилля

Таблица 2 – Перечисление собственных значений для $n = \overline{1, 50}$

n	μ_n	n	μ_n	n	μ_n
1	1,38719	18	53,54553	35	106,8836
2	4,199723	19	56,67945	36	110,0236
3	7,093562	20	59,81414	37	113,1629
4	10,06257	21	62,94959	38	116,303
5	13,08451	22	66,08582	39	119,443
6	16,14096	23	69,22281	40	122,5831
7	19,21965	24	72,35903	41	125,7232
8	22,31369	25	75,49679	42	128,8632
9	25,41846	26	78,63455	43	132,0033
10	28,53014	27	81,7723	44	135,1433
11	31,64719	28	84,91083	45	138,2842
12	34,76884	29	88,04935	46	141,4242
13	37,89356	30	91,18788	47	144,565
14	41,02058	31	94,3264	48	147,7051
15	44,1499	32	97,46569	49	150,8459
16	47,28075	33	100,605	50	153,9867
17	50,41237	34	103,7443		

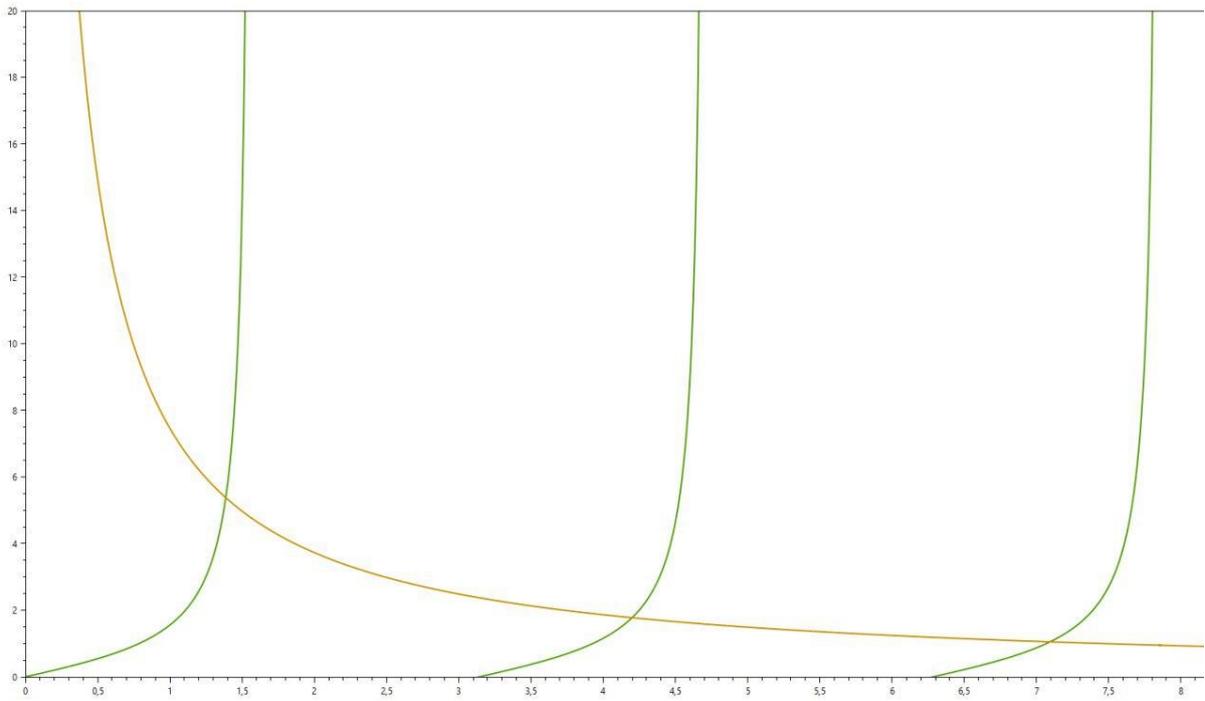


Рисунок 4 – Графическая локализация первых трех корней уравнения

$$\operatorname{tg}(\mu) = \frac{\eta}{\mu}$$