

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Самарский национальный исследовательский университет
имени академика С.П.Королёва»

Факультет информатики

Кафедра технической кибернетики

Отчет по курсовой работе

Дисциплина: «Уравнения математической физики»

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

Вариант № 2

Выполнили студенты: Катков П.И., Гайдар А.И.

Группа: 6407-010302D

Преподаватель: Дегтярев А.А.

Самара 2018

ЗАДАНИЕ К КУРСОВОЙ РАБОТЕ

- 1 Осуществить математическую постановку краевой задачи для физического процесса, описанного в предложенном варианте курсовой работы.
- 2 Используя метод разделения переменных (метод Фурье) получить решение краевой задачи в виде разложения в ряд Фурье по собственным функциям оператора Лапласа, соответствующим краевым условиям задачи.
- 3 Исследовать сходимость ряда Фурье, получить оценку остатка ряда.
- 4 Разработать компьютерную программу расчета функции-решения краевой 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 вычисления используемых собственных чисел оператора Лапласа. Обеспечить контроль погрешности вычисления собственных чисел. Компьютерная программа должна обеспечивать возможность диалогового режима ввода физических, геометрических параметров задачи, числа суммируемых элементов ряда, графическую визуализацию рассчитанного решения задачи.
- 5 Используя разработанную программу, провести экспериментальное исследование качества полученной аналитической оценки остатка ряда Фурье.
- 6 Оформить отчет о выполненной курсовой работе.

Разработать программу численного моделирования теплового процесса в тонком однородном стержне длиной l , сечением s на временном промежутке $0 < t \leq T$, если боковая поверхность стержня теплоизолирована, температура одного из концов стержня поддерживается равной u_0 , а на другом конце происходит теплообмен с окружающей средой, описываемый законом Ньютона с коэффициентом теплообмена α .

Начальная температура стержня распределена по закону

$$u|_{t=0} = \psi(x), \quad 0 \leq x \leq l.$$

Коэффициенты теплопроводности и объёмной теплоёмкости материала стержня равны k и c соответственно.

Предполагается отсутствие внутренних источников тепла. Температура окружающей среды равна u_0 .

Для расчёта функции-решения задачи использовать аналитическую форму её представления в виде разложения в ряд Фурье по собственным функциям оператора Лапласа, соответствующим краевым условиям задачи.

При проведении расчётов использовать значения параметров $l, s, T, u_0, k, c, \alpha$, а также выражение функции $\psi(x)$, указанные преподавателем.

Значения параметров, указанные преподавателем:

$$\begin{aligned} l &= 2 \text{ см}, \\ s &= 0,01 \text{ мм}^2, \\ T &= [0; 20] \text{ с}, \\ u_0 &= 0^\circ, \\ k &= 0,04 \frac{\text{Вт}}{\text{см} \cdot \text{К}}, \\ c &= 2 \frac{\text{Дж}}{\text{см}^3 \cdot \text{К}}, \\ \alpha &= 0,01 \frac{\text{Вт}}{\text{см}^2 \cdot \text{К}}, \\ \psi(x) &= \sin\left(\frac{\pi x}{l}\right). \end{aligned}$$

РЕФЕРАТ

Отчет по курсовой работе: 31 с., 7 рисунков, 4 таблицы, 4 источника, 1 приложение.

УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ, КРАЕВАЯ ЗАДАЧА, УРАВНЕНИЕ ТЕПЛОПРОВОДНОСТИ, МЕТОД РАЗДЕЛЕНИЯ ПЕРЕМЕННЫХ, ЗАДАЧА ШТУРМА-ЛИУВИЛЛЯ, РЯД ФУРЬЕ, ОЦЕНКА ОСТАТКА РЯДА.

Целью курсовой работы является получение решения краевой задачи теплопроводности для тонкого однородного стержня в виде разложения в ряд Фурье по собственным функциям оператора Лапласа и создание компьютерной программы для расчета функции-решения с заданной точностью.

Для получения аналитического решения краевой задачи использован метод разделения переменных (метод Фурье). Решение задачи получено в виде бесконечного ряда Фурье. Получена оценка остатка ряда.

Разработана компьютерная программа, обеспечивающая расчет и графическую визуализацию распределения температуры в стержне на временном промежутке $0 < t \leq T$. Для контроля погрешности усечения бесконечного ряда использована оценка остатка ряда.

Приведены графические результаты численного решения задачи теплопроводности, а также результаты экспериментального исследования практической пригодности полученной оценки остатка ряда.

Программа написана на языке Python 3.6, операционная система Ubuntu 18.04.1.

СОДЕРЖАНИЕ

Введение	6
1 Математическая постановка задачи	7
2 Аналитическое решение краевой задачи теплопроводности методом разделения переменных.....	10
3 Расчёт собственных чисел оператора Лапласа (Катков П.И.)	15
4 Исследование сходимости ряда Фурье. Оценка остатка ряда (Гайдар А. И.).....	18
5 Исследование практической пригодности оценки остатка ряда с помощью вычислительных экспериментов	21
6 Сценарий использования программы	22
Заключение	25
Список использованных источников	26
Приложение А Код программы на языке Python	26

ВВЕДЕНИЕ

Метод разделения переменных относится к классу аналитических методов решения краевых задач математической физики. Характеризуя этот метод необходимо выделить его достоинства и недостатки в сравнении с другими методами.

К достоинствам метода разделения переменных следует отнести возможность получения точного решения краевой задачи в виде ряда Фурье. Такая форма решения задачи часто и весьма успешно используется для теоретического исследования свойств этого решения. В случае достаточно быстрой сходимости ряда Фурье она может с успехом использоваться для численного моделирования физического процесса (явления).

К числу недостатков метода следует отнести его невысокую универсальность. Этот метод весьма проблематично использовать для решения нелинейных уравнений математической физики, уравнений с переменными операторными коэффициентами, а также для решения краевых задач в областях со сложными границами.

Суть метода разделения переменных состоит в факторизации по каждой независимой переменной функции, определяющей решение уравнения математической физики. Далее осуществляется переход к так называемой задаче Штурма-Лиувилля, решение которой приводит к получению собственных функций и соответствующих им собственных чисел оператора Лапласа. Затем решение исходной задачи ищется в виде ряда Фурье по этим собственным функциям.

В настоящей работе метод разделения переменных применен для получения аналитической формы решения краевой задачи теплового процесса в однородном стержне. На основе этого результата разработан алгоритм и компьютерная программа численного моделирования теплового процесса в тонком однородном стержне.

1 Математическая постановка задачи

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

На рисунке 1 схематично изображён тонкий однородный стержень.

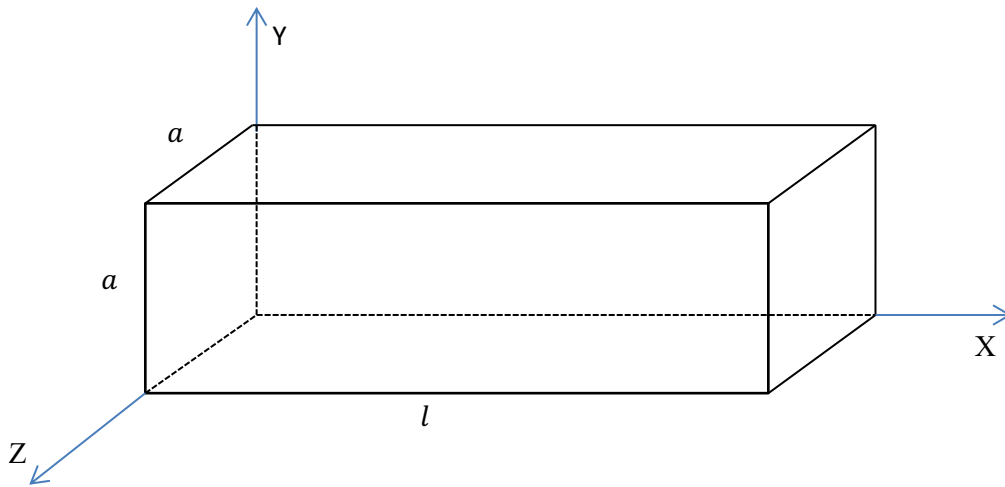


Рисунок 1 – Схематичное изображение стержня

Для вывода уравнения распределения тепла выделим произвольный объем V и рассмотрим изменение количества тепла в этом объеме за промежуток времени $0 < t \leq T$:

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

Будем считать, что сечение стержня представляет собой квадрат со стороной a . Найдем среднее распределение тепла по сечению стержня следующим образом:

$$v = \frac{1}{a^2} \int_0^a \int_0^a u(x, y, z, t) dy dz.$$

Найдём двойные интегралы, воспользовавшись теоремой Фубини [1]:

$$\frac{1}{a^2} \int_0^a \int_0^a \frac{\partial u}{\partial t} dy dz = \frac{1}{a^2} \int_0^a \left(\int_0^a \frac{\partial u}{\partial t} dy \right) dz = \frac{\partial v}{\partial t};$$

$$\frac{1}{a^2} \int_0^a \int_0^a \frac{\partial^2 u}{\partial x^2} dy dz = \frac{1}{a^2} \int_0^a \left(\int_0^a \frac{\partial^2 u}{\partial x^2} dy \right) dz = \frac{\partial^2 v}{\partial x^2};$$

$$\frac{1}{a^2} \int_0^a \int_0^a \frac{\partial^2 u}{\partial y^2} dy dz = \frac{1}{a^2} \int_0^a \left(\int_0^a \frac{\partial^2 u}{\partial y^2} dy \right) dz = 0;$$

$$\frac{1}{a^2} \int_0^a \int_0^a \frac{\partial^2 u}{\partial z^2} dy dz = \frac{1}{a^2} \int_0^a \left(\int_0^a \frac{\partial^2 u}{\partial z^2} dz \right) dy = 0.$$

В итоге получаем:

$$\frac{\partial v}{\partial t} = \frac{k}{c} \left(\frac{\partial^2 v}{\partial x^2} \right).$$

Обозначим через q плотность теплового потока - количество тепла, проходящего через единицу площади поверхности за единицу времени [2]:

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

Теплообмен на границе стержня $x = l$ описывается законом Ньютона с коэффициентом теплообмена α , то есть плотность теплового потока q будет определяться следующим образом:

$$q|_{x=l} = \alpha(u|_{x=l} - u_0).$$

Продифференцируем исходную функцию по нормали на грани $x = l$:

$$\left. \frac{\partial u}{\partial \bar{n}_x} \right|_{x=l} = \left(\frac{\partial u}{\partial x} \cos(\bar{n}_x, \bar{e}_x) + \frac{\partial u}{\partial y} \cos(\bar{n}_x, \bar{e}_y) + \frac{\partial u}{\partial z} \cos(\bar{n}_x, \bar{e}_z) \right) \Big|_{x=l} = \frac{\partial u}{\partial x}.$$

Выразим значение плотности теплового потока в точке l по оси x :

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

Выражая из последней формулы частную производную получаем:

$$\frac{\partial u}{\partial x} \Big|_{x=l} = -\frac{\alpha}{k}(u|_{x=l} - u_0).$$

В итоге математическая модель будет выглядеть следующим образом:

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

Сделаем замену $w = v - u_0$.

После замены итоговая математическая модель будет иметь вид:

$$\begin{cases} \frac{\partial w}{\partial t} = \frac{k}{c} \left(\frac{\partial^2 w}{\partial x^2} \right), & x \in [0, l], \quad t \in [0, T]; \\ w|_{t=0} = \psi(x), & x \in [0, l]; \\ w|_{x=0} = 0, & t \in (0, T]; \\ \frac{\partial w}{\partial x} \Big|_{x=l} = -\frac{\alpha}{k}w|_{x=l}, & t \in (0, T]. \end{cases} \quad (1.1)$$

2 Аналитическое решение краевой задачи теплопроводности методом разделения переменных

Будем решать нашу задачу методом разделения переменных.

Представим функцию $w(x, t)$ в виде произведения двух функций:

$$w(x, t) = X(x)T(t) \quad (2.1)$$

Подставив (2.1) в первое уравнение системы (1.1) получим:

$$X(x)T'(t) = \frac{k}{c}X''(x)T(t).$$

Преобразуем и используем метод разделения переменных:

$$\frac{X''(x)}{X(x)} = \frac{c}{k} \frac{T'(t)}{T(t)} = -\lambda = \text{const}. \quad (2.2)$$

Заметим, что решение нашей задачи относительно X сводится к решению задачи Штурма-Лиувилля [3]:

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

Характеристическое уравнение будет иметь вид:

$$k^2 + \lambda = 0.$$

При $\lambda \leq 0$ собственные функции $X(x) \equiv 0$, то есть решение задачи тривиально, а такие решения нас не интересуют.

Рассмотрим случай $\lambda > 0$:

$$\begin{aligned} k &= \pm\sqrt{-\lambda} = \pm i\sqrt{\lambda}; \\ X(x) &= C_1 \cos(\sqrt{\lambda} x) + C_2 \sin(\sqrt{\lambda} x). \end{aligned} \quad (2.3)$$

Подставим в (2.3) граничные условия:

$$\begin{aligned} X(0) &= C_1 = 0; \\ X'(l) &= C_2 \sqrt{\lambda} \cos(\sqrt{\lambda} l) = -\frac{\alpha}{k} C_2 \sin(\sqrt{\lambda} l). \end{aligned}$$

Получим соотношение:

$$\sqrt{\lambda} \cos(\sqrt{\lambda} l) = -\frac{\alpha}{k} \sin(\sqrt{\lambda} l). \quad (2.4)$$

Упростив соотношение (2.4) получим трансцендентное уравнение:

$$\frac{-\sqrt{\lambda}k}{\alpha} = \operatorname{tg}(\sqrt{\lambda}l). \quad (2.5)$$

Упростим трансцендентное уравнение. Для этого сделаем следующие замены:

$$\sqrt{\lambda}l = \mu; \quad (2.6)$$

$$\frac{k}{\alpha l} = z. \quad (2.7)$$

После замен уравнение (2.5) будет иметь вид:

$$\operatorname{tg}(\mu) + \mu z = 0. \quad (2.8)$$

Подробное решение уравнения (2.8) описано в разделе 3.

Подставим в (2.3) найденные μ и получим собственные функции:

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

Решим теперь нашу задачу относительно T . Из (2.2) получим:

$$\frac{c}{k} \frac{T'(t)}{T(t)} = -\lambda.$$

Преобразуем и решим дифференциальное уравнение с разделяющимися переменными:

$$cT'(t) + k\lambda T(t) = 0;$$

$$c \frac{dT}{dt} + k\lambda T(t) = 0;$$

$$\int \frac{dT}{T} = \int \frac{-k\lambda dt}{c};$$

$$T(t) = C_1 e^{\frac{-k\lambda t}{c}}.$$

Воспользуемся заменой (2.6):

$$T_n(t) = C_1 e^{\frac{-k\mu_n^2 t}{cl^2}}.$$

Линейная комбинация частных решений является общим решением уравнения с краевыми условиями, поэтому функция $w(x, t)$ представима в виде ряда:

$$w(x, t) = \sum_{n=0}^{\infty} A_n X_n(x) T_n(t), \quad (2.9)$$

где A_n – некоторые константы.

Подставим найденные функции $X(x)$ и $T(t)$ в (2.9) получим:

$$w(x, t) = \sum_{n=1}^{\infty} A_n \sin\left(\frac{\mu_n x}{l}\right) e^{\frac{-k\mu_n^2 t}{cl^2}}. \quad (2.10)$$

Для того чтобы найти A_n подставим в (2.10) начальное условие и воспользуемся свойством ортогональности [4]:

$$w(x, 0) = \psi(x) = \sum_{n=1}^{\infty} A_n \sin\left(\frac{\mu_n x}{l}\right). \quad (2.11)$$

Собственные функции задачи Штурма-Лиувилля образуют ортогональный базис гильбертова пространства, и любой его элемент представим рядом Фурье.

По свойству ортогональности будет иметь место следующее равенство:

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

Умножим (2.11) на $\sin\left(\frac{\mu_m x}{l}\right)$, проинтегрируем с обеих сторон и воспользуемся свойством, что интеграл суммы равен сумме интегралов, получим:

$$\begin{aligned} \int_0^l \psi(x) \sin\left(\frac{\mu_m x}{l}\right) dx &= \sum_{n=0}^{\infty} A_n \int_0^l \sin\left(\frac{\mu_n x}{l}\right) \sin\left(\frac{\mu_m x}{l}\right) dx; \\ \int_0^l \psi(x) \sin\left(\frac{\mu_m x}{l}\right) dx &= A_m \int_0^l \sin^2\left(\frac{\mu_m x}{l}\right) dx. \end{aligned}$$

В итоге, коэффициент A_m может быть найден из следующего соотношения:

$$A_m = \frac{\int_0^l \psi(x) \sin\left(\frac{\mu_m x}{l}\right) dx}{\int_0^l \sin^2\left(\frac{\mu_m x}{l}\right) dx}. \quad (2.12)$$

Найдём значение знаменателя формулы (2.12):

$$\begin{aligned} \int_0^l \sin^2\left(\frac{\mu_m x}{l}\right) dx &= \left(\frac{x}{2} - \frac{l \sin \frac{2\mu_m x}{l}}{4\mu_m} \right) \Big|_0^l = \frac{l}{2} - \frac{l \sin \frac{2\mu_m l}{l}}{4\mu_m} = \\ &= \frac{2\mu_m l - l \sin(2\mu_m)}{4\mu_m}. \end{aligned}$$

Возьмём из начальных данных функцию ψ :

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

Найдём значение числителя формулы (2.12):

$$\begin{aligned} \int_0^l \psi(x) \sin\left(\frac{\mu_m x}{l}\right) dx &= \int_0^l \sin\left(\frac{\pi x}{l}\right) \sin\left(\frac{\mu_m x}{l}\right) dx = \\ &= \frac{1}{2} \int_0^l \left(\cos\left(\frac{\pi x}{l} - \frac{\mu_m x}{l}\right) - \cos\left(\frac{\pi x}{l} + \frac{\mu_m x}{l}\right) \right) dx = \\ &= \frac{1}{2} \left(\frac{l \sin\left(\frac{\pi x}{l} - \frac{\mu_m x}{l}\right)}{\pi - \mu_m} - \frac{l \sin\left(\frac{\pi x}{l} + \frac{\mu_m x}{l}\right)}{\pi + \mu_m} \right) \Big|_0^l = \\ &= \frac{1}{2} \left(\frac{l \sin(\pi - \mu_m)}{\pi - \mu_m} - \frac{l \sin(\pi + \mu_m)}{\pi + \mu_m} \right) - 0 = \\ &= \frac{1}{2} \left(\frac{l \sin(\mu_m)}{\pi - \mu_m} + \frac{l \sin(\mu_m)}{\pi + \mu_m} \right). \end{aligned}$$

Таким образом, по формуле (2.12) теперь мы можем найти коэффициент A_m :

$$A_m = \frac{1}{2} \left(\frac{l \sin(\mu_m)}{\pi - \mu_m} + \frac{l \sin(\mu_m)}{\pi + \mu_m} \right) \frac{4\mu_m}{2\mu_m l - l \sin(2\mu_m)}.$$

Подставим найденный коэффициент A_m в (2.10):

$$w(x, t) = \sum_{n=1}^{\infty} \frac{1}{2} \left(\frac{l \sin(\mu_n)}{\pi - \mu_n} + \frac{l \sin(\mu_n)}{\pi + \mu_n} \right) \frac{4\mu_n}{2\mu_n l - l \sin(2\mu_n)} \sin\left(\frac{\mu_n x}{l}\right) e^{\frac{-k\mu_n^2 t}{cl^2}}.$$

Сделаем обратную замену и упростим. Полученный итоговый ряд Фурье имеет вид:

$$v(x, t) = u_0 + \sum_{n=1}^{\infty} \left(\frac{\sin(\mu_n)}{\pi - \mu_n} + \frac{\sin(\mu_n)}{\pi + \mu_n} \right) \frac{2\mu_n}{2\mu_n - \sin(2\mu_n)} \sin\left(\frac{\mu_n x}{l}\right) e^{\frac{-k\mu_n^2 t}{cl^2}};$$

$$v(x, t) = u_0 + \sum_{n=1}^{\infty} \frac{4\pi\mu_n \sin(\mu_n)}{(\pi^2 - \mu_n^2)(2\mu_n - \sin(2\mu_n))} \sin\left(\frac{\mu_n x}{l}\right) e^{\frac{-k\mu_n^2 t}{cl^2}}.$$

Последняя формула использована в расчётах для получения решения исходной задачи численного моделирования теплового процесса в тонком однородном стержне.

3 Расчёт собственных чисел оператора Лапласа

Из решения нашей краевой задачи методом разделения переменных мы получили, что собственные числа определяются из трансцендентного уравнения (2.8).

Изобразим на рисунке 2 решение уравнения (2.8) при $z = 2$ (то есть для параметров, которые даны в задании):

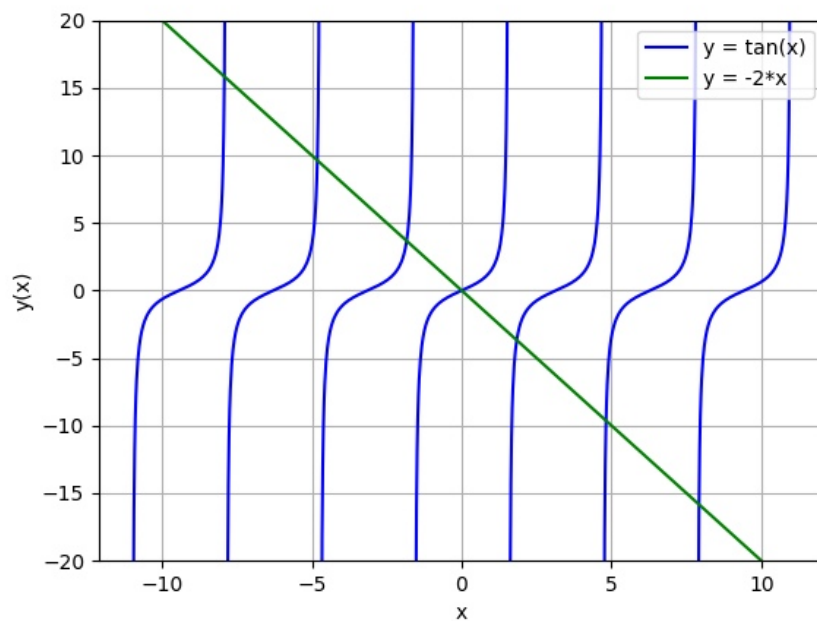


Рисунок 2 – Решение трансцендентного уравнения (2.8) при $z = 2$

Будем решать уравнение (2.8) с помощью метода Ньютона, так как он обладает квадратичной скоростью сходимости и очень прост в реализации.

Алгоритм нахождения численного решения уравнения $f(x) = 0$ сводится к итерационной процедуре вычисления [5]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Заметим, что уравнение (2.8) имеет полюса:

$$p = \frac{\pi}{2} + \pi k.$$

Из-за полюсов и разрывов метод Ньютона может давать много неточностей или вовсе неправильные результаты. Для устранения полюсов и

других особенностей функции умножим уравнение (2.8) на $\cos(\mu)$ и получим наше уравнение в более удобном виде уже без полюсов и разрывов:

$$f(x) = \sin(\mu) + \mu z \cos(\mu) = 0.$$

Продифференцируем $f(x)$:

$$f'(x) = \cos(\mu) + z \cos(\mu) - \mu z \sin(\mu).$$

Метод Ньютона надёжно находит корень только тогда, когда начальная точка достаточно близка к корню. Если точка находится дальше, то корень может быть очень далеко от правильного. Таким образом, чтобы применить метод Ньютона, с самого начала нужно найти хорошие корневые аппроксимации.

Возьмём $\arctg$ от обеих частей уравнения (2.8) и получим эквивалентное уравнение:

$$\mu + \arctg(\mu z) = \pi k;$$

$$\mu = \pi k - \arctg(\mu z).$$

Мы знаем, что все наши корни удовлетворяют следующему неравенству:

$$-\frac{\pi}{2} + \pi k \leq \mu_k \leq \frac{\pi}{2} + \pi k.$$

Таким образом, мы можем выбрать некоторую среднюю точку πk и построить итерационный процесс и получить наши начальные корневые аппроксимации:

$$\mu_k = \pi k - \arctg(\pi k z).$$

На рисунке 3 приведён график корневой аппроксимации уравнения (2.8) при $z = 2$.

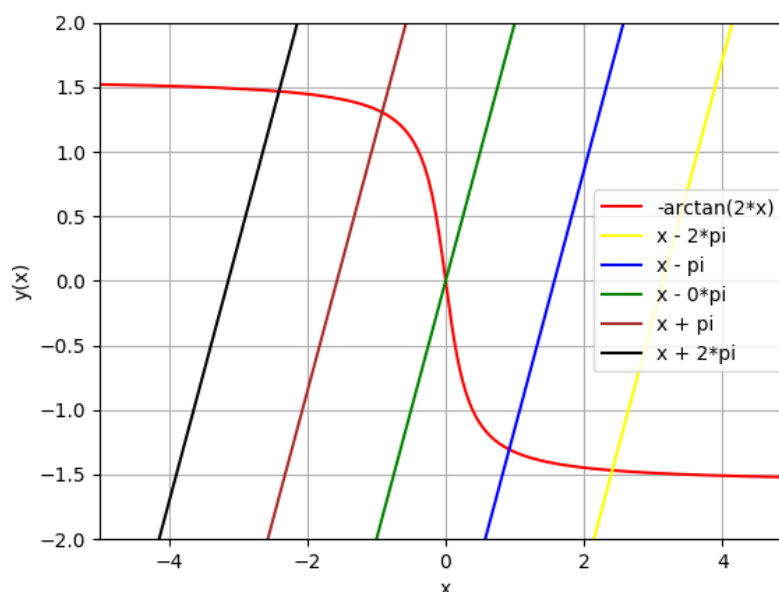


Рисунок 3 – Корневые аппроксимации для уравнения (2.8) при $z = 2$

Арктангенс – это гладкая и ограниченная функция, поэтому даже при малых значениях k с помощью метода Ньютона получится довольно хорошее приближение.

В таблице 1 приведены результаты использования метода Ньютона для нахождения μ из уравнения (2.8).

Таблица 1 – Первые 60 найденных значений μ_k по методу Ньютона

k	μ_k	k	μ_k	k	μ_k	k	μ_k
1	1,83660	16	48,70495	31	95,82379	46	142,94596
2	4,81584	17	51,84592	32	98,96522	47	146,08748
3	7,91705	18	54,98696	33	102,10666	48	149,22900
4	11,04083	19	58,12806	34	105,24811	49	152,37053
5	14,17243	20	61,26922	35	108,38956	50	155,51205
6	17,30764	21	64,41041	36	111,53102	51	158,65358
7	20,44480	22	67,55164	37	114,67249	52	161,79511
8	23,58314	23	70,69291	38	117,81397	53	164,93665
9	26,72225	24	73,83420	39	120,95545	54	168,07818
10	29,86187	25	76,97552	40	124,09694	55	171,21972
11	33,00187	26	80,11685	41	127,23843	56	174,36126
12	36,14215	27	83,25821	42	130,37993	57	177,50280
13	39,28264	28	86,39958	43	133,52143	58	180,64435
14	42,42329	29	89,54097	44	136,66294	59	183,78589
15	45,56407	30	92,68238	45	139,80445	60	186,92744

4 Исследование сходимости ряда Фурье. Оценка остатка ряда

Найдём сначала оценку в момент времени $t = 0$:

Выполним оценку сверху следующим образом:

$$\sin(\mu_n) \leq 1;$$

$$\sin\left(\frac{\mu_n x}{l}\right) \leq 1;$$

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

Собственные числа повторяются с периодом π , исходя из этого, для μ_n мы можем записать следующее неравенство:

$$-\frac{\pi}{2} + \pi n \leq \mu_n \leq \frac{\pi}{2} + \pi n. \quad (4.1)$$

Используя неравенство (4.1) получим:

$$\begin{aligned}
 & \left| \sum_{n=N+1}^{\infty} \frac{2\pi}{(\mu_n^2 - \pi^2)} \right| \leq \left| \sum_{n=N+1}^{\infty} \frac{2\pi}{\left(\left(-\frac{\pi}{2} + \pi n\right)^2 - \pi^2\right)} \right| = \\
 & = \left| \frac{2}{\pi} \sum_{n=N+1}^{\infty} \frac{1}{\left(n^2 - n - \frac{3}{4}\right)} \right| \leq \frac{2}{\pi} \int_N^{\infty} \frac{dn}{\left(n^2 - n - \frac{3}{4}\right)} = \\
 & = \frac{1}{\pi} \ln\left(\frac{3 - 2N}{2N + 1}\right) = \Phi(N, 0) \leq \varepsilon.
 \end{aligned}$$

Проведем аналогичные рассуждения для момента времени $t \neq 0$:

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

Воспользуемся неравенством:

$$\frac{1}{\left(\pi n - \frac{\pi}{2}\right)\left(\left(\pi n - \frac{\pi}{2}\right)^2 - \pi^2\right)} \leq \frac{1}{\left(\pi N - \frac{\pi}{2}\right)\left(\left(\pi N - \frac{\pi}{2}\right)^2 - \pi^2\right)}. \quad (4.2)$$

Умножим и разделим наш ряд на $\left(\pi n - \frac{\pi}{2}\right)$ и оценим его сверху интегралом, получим:

$$2\pi \int_N^{\infty} \frac{\left(\pi n - \frac{\pi}{2}\right) e^{\frac{-k\mu_n^2 t}{cl^2}} dn}{\left(\pi n - \frac{\pi}{2}\right) \left(\left(\pi n - \frac{\pi}{2}\right)^2 - \pi^2\right)}.$$

Теперь используя неравенство (4.2) переходим к обычному табличному интегралу и берём его:

$$\begin{aligned} & 2\pi \int_N^{\infty} \frac{\left(\pi n - \frac{\pi}{2}\right) e^{\frac{-k\mu_n^2 t}{cl^2}} dn}{\left(\pi n - \frac{\pi}{2}\right) \left(\left(\pi n - \frac{\pi}{2}\right)^2 - \pi^2\right)} \leq \\ & \leq \frac{2\pi}{\left(\pi N - \frac{\pi}{2}\right) \left(\left(\pi N - \frac{\pi}{2}\right)^2 - \pi^2\right)} \int_N^{\infty} \left(\pi n - \frac{\pi}{2}\right) e^{\frac{-k\left(\pi n - \frac{\pi}{2}\right)^2 t}{cl^2}} dn = \\ & = \frac{cl^2}{kt \left(\pi N - \frac{\pi}{2}\right) \left(\left(\pi N - \frac{\pi}{2}\right)^2 - \pi^2\right)} \int_N^{\infty} e^{\frac{-k\left(\pi n - \frac{\pi}{2}\right)^2 t}{cl^2}} d\left(e^{\frac{-k\left(\pi n - \frac{\pi}{2}\right)^2 t}{cl^2}}\right) = \\ & = \frac{cl^2 e^{\frac{-k\left(\pi N - \frac{\pi}{2}\right)^2 t}{cl^2}}}{kt \left(\pi N - \frac{\pi}{2}\right) \left(\left(\pi N - \frac{\pi}{2}\right)^2 - \pi^2\right)} = \Phi(N, t) \leq \varepsilon. \end{aligned}$$

Используя интегральный признак Коши, делаем вывод о сходимости ряда. Таким образом, мы получили оценку остатка ряда в любой момент времени. Для заданной точности ε мы можем получить конечное число N суммирований ряда, благодаря чему мы можем свести задачу к конечному ряду Фурье.

Для нахождения N в момент времени t возьмём $N_0 = 1$ получим $\Phi_0 = \Phi(N_0, t)$, если $\Phi_0 < \varepsilon$, то $N = N_0$, иначе $N_0 = N_0 + 1$ и возвращаемся к расчету Φ_0 при новом N_0 .

5 Исследование практической пригодности оценки остатка ряда с помощью вычислительных экспериментов

Проведём исследование практической пригодности оценки остатка ряда. Для определения качества оценки остатка ряда сравним рассчитанные по нашей оценке значения количества слагаемых ряда N_T со значениями экспериментального количества слагаемых N_ε при различных ε . Для того, чтобы определить N_ε возьмем $N = N_T$ и будем уменьшать N пока не получим $|u_N - u_{N_T}| > \varepsilon$. Исходным результатом будем считать $N_\varepsilon = N + 1$ при котором погрешность все еще меньше ε . Результаты сравнения приведены в таблицах 2,3 и 4.

Таблица 2 – Результаты сравнения N_T и N_ε при $t = 0,001, x = 2$

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
N_T	40	79	138	204	268	327
N_ε	6	33	98	173	243	306

Таблица 3 – Результаты сравнения N_T и N_ε при $t = 0, x = 2$

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
N_T	4	29	277	2766	27649
N_ε	3	20	194	1929	19277

Таблица 4 – Результаты сравнения N_T и N_ε при $t = 2, x = 1,5$

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
N_T	4	5	7	8	9	10
N_ε	2	4	5	7	8	9

Из таблиц 2,3 и 4 видно, что самый плохой результат даёт оценка остатка ряда в момент времени $t = 0$ в точке $x = 2$. Это связано с тем, что ряд в этой точке медленнее всего сходится, потому что точка $x = 2$ это правый конец стержня на котором происходит теплообмен с окружающей средой по закону Ньютона.

6 Сценарий использования программы

В данной работе реализована программа численного моделирования теплового процесса в тонком однородном стержне на языке программирования Python. Расчётную часть программы выполнил студент Катков П.И. Графический интерфейс реализовал студент Гайдар А.И.

Для удобства использования приложения был написан графический интерфейс с помощью пакета PyQt5. На рисунке 4 изображён графический интерфейс приложения.

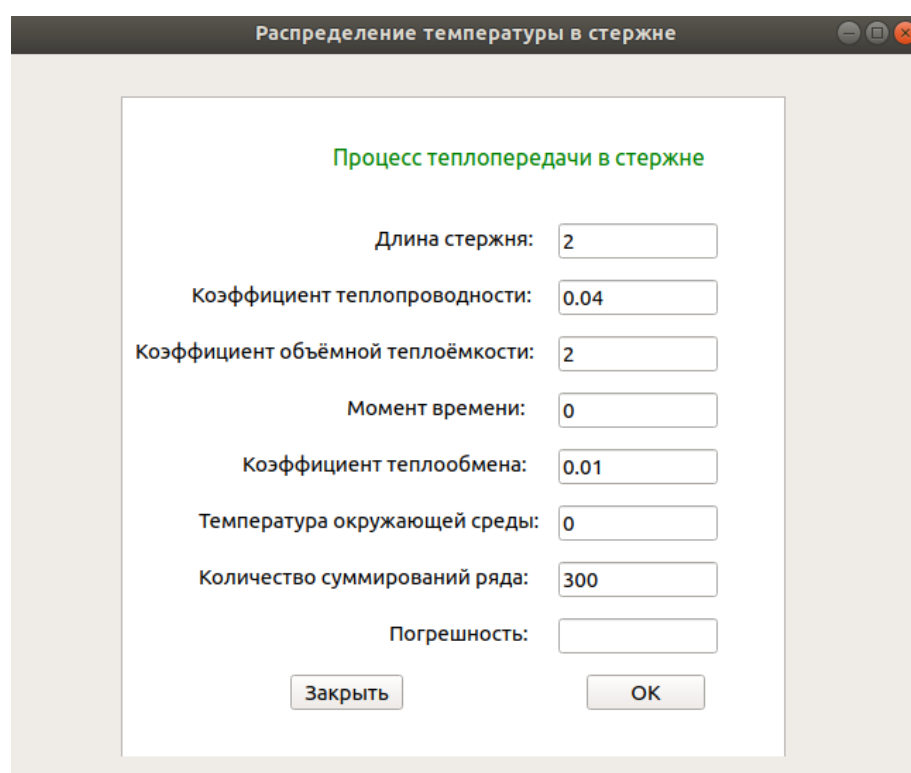


Рисунок 4 – графический интерфейс приложения

Для удобства в поля ввода добавлены параметры по умолчанию и обработчик ошибок для случаев, когда пользователь вводит неправильные данные. Следует также сказать, что поле погрешность является необязательным для заполнения. Для получения результата достаточно заполнения одного из двух полей – «Количество суммирований ряда» или «Погрешность».

На рисунках 5,6,7 представлены примеры работы программной реализации при различных параметрах.

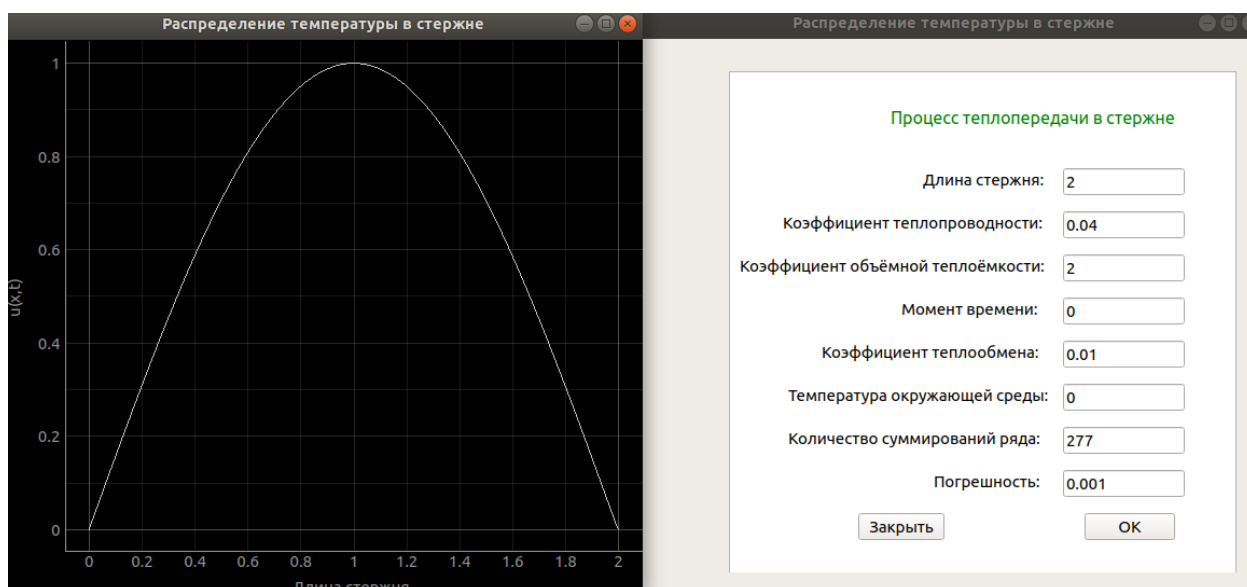


Рисунок 5 – Результат работы программы в момент времени $t = 0$

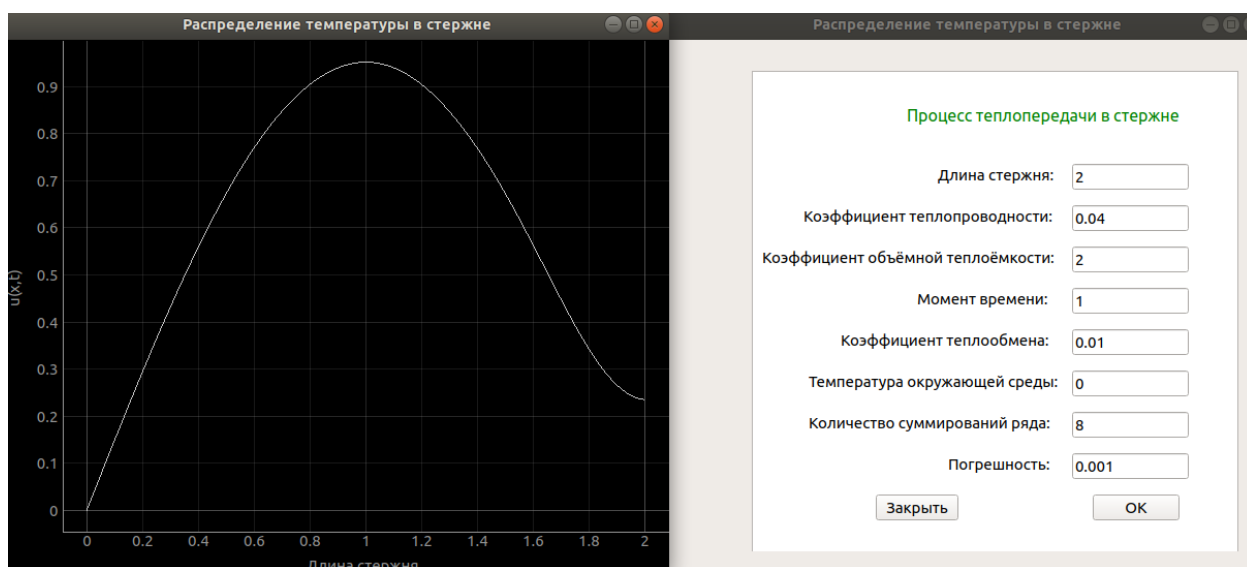


Рисунок 6 - Результат работы программы в момент времени $t = 1$

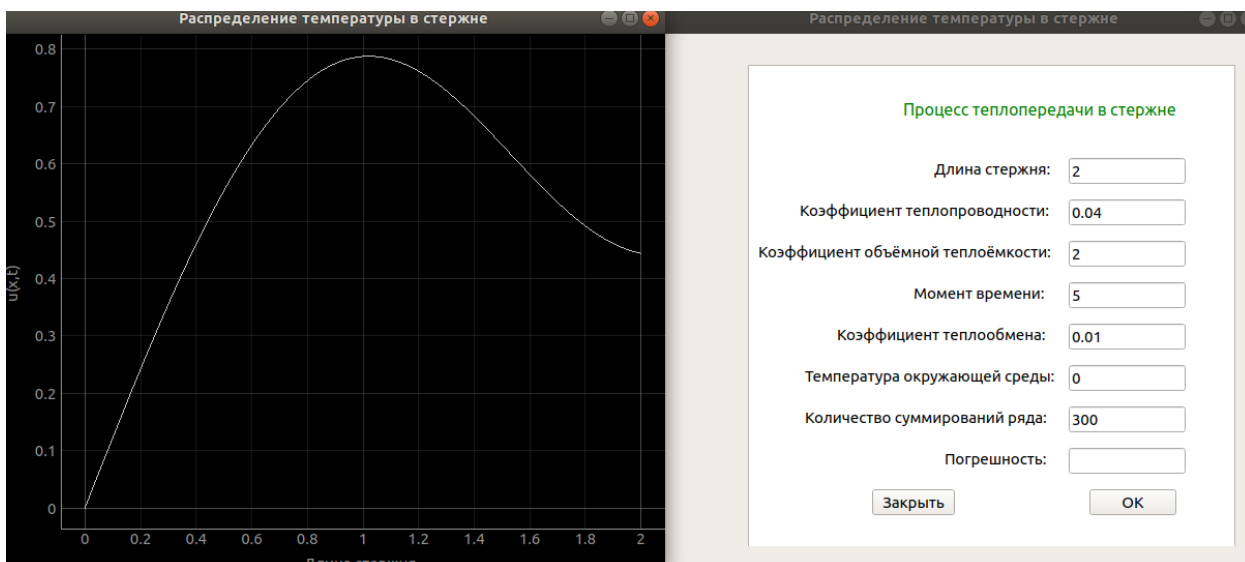


Рисунок 7 - Результат работы программы в момент времени $t = 5$

Как видно из рисунка 5 в момент времени $t = 0$ программа выдаёт синусоидальное распределение, что соответствует начальному условию. Из рисунков 6 и 7 можно видеть, как на правом конце стержня при увеличении времени происходит процесс теплообмена с окружающей средой, описываемый законом Ньютона. Во всех случаях температура левого конца стержня поддерживается постоянной и равна $u_0 = 0$. Таким образом, все начальные и граничные условия в программной разработке выполняются.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы осуществлена постановка краевой задачи для физического процесса, описанного в задании к курсовой работе, получено аналитическое решение задачи в виде бесконечного ряда Фурье.

Было установлено, что ряд сходится. Была получена оценка сверху остатка ряда, которая позволила обеспечить контроль погрешности усечения бесконечного ряда.

Для анализа полученного решения была разработана программа на языке программирования Python 3.6 для численного моделирования теплообмена в тонком однородном стержне. Программа производит расчёт температуры с заданными параметрами и точностью, а также обеспечивает визуализацию распределения температуры по длине стержня.

В результате серии вычислительных экспериментов установлено, что использование в программе полученной оценки остатка ряда может приводить к существенной избыточности числа суммируемых элементов ряда. Например, если значение переменной времени задать $t = 0$ с., то избыточность суммируемых элементов ряда составляет примерно 40-45%. Однако, при дальнейшем увеличении значений времени избыточность оказывается практически несущественной и почти не сказывается на времени работы программы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Смирнов В.И. Курс высшей математики / В.И. Смирнов - М.: Наука, 1974.-Т. 2. - 479 с.
2. Тихонов А. Н. Уравнения математической физики: учеб. Пособие для университетов / А. Н. Тихонов, А. А. Самарский. - М.: Наука, 1972 –736 с.
3. Владимиров В. С. Уравнения математической физики: учеб. Для вузов / В. С. Владимиров, В. В. Жаринов. - М.: Физико-математическая литература, 2000 - 399 с.
4. Голоскоков Д. П. Уравнения математической физики: решение задач в системе Maple / Д. П. Голоскоков. - СПб.; М.; Нижний Новгород: Питер: Питер принт, 2004 - 538 с.
5. 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.

ПРИЛОЖЕНИЕ А

Код программы на языке Python

```

from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QMessageBox
from math import tan, cos, pi, sqrt, sin, exp, fabs, atan, log10

import matplotlib.pyplot as plt
import numpy as np
import pyqtgraph as pg
import os

from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
from matplotlib.figure import Figure
import matplotlib.pyplot as plt

class Ui_Temperature_in_rod(object):
    def setupUi(self, Temperature_in_rod):
        Temperature_in_rod.setObjectName("Распределение температуры в стержне")
        Temperature_in_rod.resize(650, 520)
        self.centralwidget = QtWidgets.QWidget(Temperature_in_rod)
        self.centralwidget.setObjectName("centralwidget")
        self.listWidget = QtWidgets.QListWidget(self.centralwidget)
        self.listWidget.setGeometry(QtCore.QRect(80, 30, 471, 480))
        self.listWidget.setObjectName("listWidget")
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(200, 440, 80, 25))
        self.pushButton.setObjectName("pushButton")
        self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(410, 440, 80, 25))
        self.pushButton_2.setObjectName("pushButton_2")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setEnabled(True)
        self.label.setGeometry(QtCore.QRect(230, 60, 264, 27))
        self.label.setTextFormat(QtCore.Qt.AutoText)
        self.label.setStyleSheet('color: green')
        font=app.font()
        font.setPointSize(16)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit.setGeometry(QtCore.QRect(390, 120, 113, 25))
        self.lineEdit.setObjectName("lineEdit")
        self.lineEdit.setText("2")
        self.lineEdit_2 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_2.setGeometry(QtCore.QRect(390, 160, 113, 25))
        self.lineEdit_2.setObjectName("lineEdit_2")
        self.lineEdit_2.setText("0.04")
        self.lineEdit_3 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_3.setGeometry(QtCore.QRect(390, 200, 113, 25))
        self.lineEdit_3.setObjectName("lineEdit_3")
        self.lineEdit_3.setText("2")
        self.lineEdit_4 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_4.setGeometry(QtCore.QRect(390, 240, 113, 25))
        self.lineEdit_4.setObjectName("lineEdit_4")
        self.lineEdit_4.setText("0")
        self.lineEdit_5 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_5.setGeometry(QtCore.QRect(390, 280, 113, 25))
        self.lineEdit_5.setObjectName("lineEdit_5")
        self.lineEdit_5.setText("0.01")
        self.lineEdit_6 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_6.setGeometry(QtCore.QRect(390, 320, 113, 25))
        self.lineEdit_6.setObjectName("lineEdit_6")
        self.lineEdit_6.setText("0")
        self.lineEdit_7 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_7.setGeometry(QtCore.QRect(390, 360, 113, 25))
        self.lineEdit_7.setObjectName("lineEdit_7")
        self.lineEdit_7.setText("300")
        self.lineEdit_8 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_8.setGeometry(QtCore.QRect(390, 400, 113, 25))

```

```

self.lineEdit_8.setObjectName("lineEdit_8")
self.label_2 = QtWidgets.QLabel(self.centralwidget)
self.label_2.setGeometry(QtCore.QRect(260, 120, 121, 20))
self.label_2.setObjectName("label_2")
self.label_3 = QtWidgets.QLabel(self.centralwidget)
self.label_3.setGeometry(QtCore.QRect(130, 160, 251, 20))
self.label_3.setObjectName("label_3")
self.label_4 = QtWidgets.QLabel(self.centralwidget)
self.label_4.setGeometry(QtCore.QRect(90, 200, 291, 20))
self.label_4.setObjectName("label_4")
self.label_5 = QtWidgets.QLabel(self.centralwidget)
self.label_5.setGeometry(QtCore.QRect(240, 240, 141, 20))
self.label_5.setObjectName("label_5")
self.label_6 = QtWidgets.QLabel(self.centralwidget)
self.label_6.setGeometry(QtCore.QRect(166, 280, 241, 20))
self.label_6.setObjectName("label_6")
self.label_7 = QtWidgets.QLabel(self.centralwidget)
self.label_7.setGeometry(QtCore.QRect(135, 320, 241, 20))
self.label_7.setObjectName("label_7")
self.label_8 = QtWidgets.QLabel(self.centralwidget)
self.label_8.setGeometry(QtCore.QRect(135, 360, 241, 20))
self.label_8.setObjectName("label_8")
self.label_9 = QtWidgets.QLabel(self.centralwidget)
self.label_9.setGeometry(QtCore.QRect(270, 400, 241, 20))
self.label_9.setObjectName("label_9")
Temperature_in_rod.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(Temperature_in_rod)
self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 22))
self.menubar.setObjectName("menubar")
Temperature_in_rod.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(Temperature_in_rod)
self.statusbar.setObjectName("statusbar")
Temperature_in_rod.setStatusBar(self.statusbar)

self.retranslateUi(Temperature_in_rod)
QtCore.QMetaObject.connectSlotsByName(Temperature_in_rod)
self.pushButton_2.clicked.connect(self.buttonClicked_2)
self.pushButton.clicked.connect(self.buttonClicked)

def retranslateUi(self, Temperature_in_rod):
    _translate = QtCore.QCoreApplication.translate
    Temperature_in_rod.setWindowTitle(_translate("Temperature_in_rod", "Распределение
температуры в стержне"))
    self.pushButton.setText(_translate("Temperature_in_rod", "Заккрыть"))
    self.pushButton_2.setText(_translate("Temperature_in_rod", "OK"))
    self.label.setText(_translate("Temperature_in_rod", "Теплообмен в стержне"))
    self.label_2.setText(_translate("Temperature_in_rod", "Длина стержня:"))
    self.label_3.setText(_translate("Temperature_in_rod", "Коэффициент
теплопроводности:"))
    self.label_4.setText(_translate("Temperature_in_rod", "Коэффициент объёмной
теплоёмкости:"))
    self.label_5.setText(_translate("Temperature_in_rod", "Момент времени:"))
    self.label_6.setText(_translate("Temperature_in_rod", "Коэффициент теплообмена:"))
    self.label_7.setText(_translate("Temperature_in_rod", "Температура окружающей
среды:"))
    self.label_8.setText(_translate("Temperature_in_rod", "Количество суммирований
ряда:"))
    self.label_9.setText(_translate("Temperature_in_rod", "Погрешность:"))
def buttonClicked_2(self):
    err = True
    flag = False
    str = ""
    str1 = ""
    str2 = ""
    try:
        l = float(self.lineEdit.text())
        if l <= 0:
            str += "Длина стержня должна быть > 0; \n"

```

```

except:
    str += "Длина стержня введёна неправильно; \n"
try:
    k = float(self.lineEdit_2.text())
    if k <= 0:
        str += "Коэффициент теплопроводности должен быть > 0; \n"
except:
    str += "Коэффициент теплопроводности введён неправильно; \n"
try:
    c = float(self.lineEdit_3.text())
    if c <= 0:
        str += "Коэффициент объёмной теплоёмкости должен быть > 0; \n"
except:
    str += "Коэффициент объёмной теплоёмкости введён неправильно; \n"
try:
    t = float(self.lineEdit_4.text())
    if t < 0:
        str += "Момент времени не может быть отрицательным; \n"
except:
    str += "Момент времени введён неправильно; \n"
try:
    alp = float(self.lineEdit_5.text())
    if alp <= 0:
        str += "Коэффициент теплообмена должен быть > 0; \n"
except:
    str += "Коэффициент теплообмена введён неправильно; \n"
try:
    u0 = float(self.lineEdit_6.text())
except:
    str += "температура окружающей среды введена неправильно; \n"
try:
    n = int(self.lineEdit_7.text())
    if n < 0:
        str1 += "Количество итераций суммирования должно быть >= 0; \n"
except:
    str1 += "Количество итераций суммирования введёно неправильно (не целое число);
\n"

# if self.lineEdit_8.text() == "":
#     flag = False
try:
    eps = float(self.lineEdit_8.text())
    if eps < 0:
        eps = abs(eps)
    elif eps == 0:
        str2 += "Погрешность = 0 не следует задавать \n"
    flag = True
except:
    str2 += "Погрешность введена неправильно \n"
if len(str1) != 0 and len(str2) != 0:
    str += str1
    str += str2
    QMessageBox.about(Temperature_in_rod, "Ошибка", str)
    err = False
elif not (((len(str1) != 0 and len(str2) == 0) or (len(str1) == 0 and len(str2) != 0))
or len(str) == 0):
    str += str1
    str += str2
    QMessageBox.about(Temperature_in_rod, "Ошибка", str)
    err = False
elif len(str) != 0:
    QMessageBox.about(Temperature_in_rod, "Ошибка", str)
    err = False

def f(x,b):
    return sin(x)+b*x*cos(x)

```

```

def f1(x,b):
    return cos(x) + b*cos(x) - sin(x)*b*x

def f_eps(n):
    return abs(log10(abs((3 - 2*n)/(2*n + 1)))/pi)

def ff_eps(n,k,t,c,l):
    gamma = k*t/(c*l*1)
    return abs((1/((pi*n - pi/2)*((pi*n - pi/2)**2 - pi**2)*gamma)*exp(-gamma*((pi*n -
pi/2)**2))))

def newtons_method(x0, f, f1, e):
    x0 = float(x0)
    while True:
        x1 = x0 - (f(x0,b) / f1(x0,b))
        if abs(x1 - x0) < e:
            return x1
        x0 = x1

if err == True:
    if flag == True:
        if t == 0:
            kk = 0
            while f_eps(kk) > eps:
                kk = kk + 1
            self.lineEdit_7.setText("%.i" % kk)
            n = kk
        else:
            kk = 0
            while ff_eps(kk,k,t,c,l) > eps:
                kk = kk + 1
            self.lineEdit_7.setText("%.i" % kk)
            n = kk

    b = float(k/(alp*1))
    e = 0.000000000001
    result = []
    for i in range(n):
        ii=i+1
        x0 = ii*pi-atan(b*ii*pi)
        result.append(newtons_method(x0, f , f1, e))
    mu = sorted(list(set(result)))
    print(mu)
    print(len(mu))

def Fourier_series(mu, l, k, c, x, t, n, u0):
    suma = 0
    for i in range(n):
        suma = suma + ((4*pi*mu[i]*sin(mu[i]))*sin(mu[i]*x/l)* exp((-
k*mu[i]*mu[i]*t)/(c*l*1)))/((pi*pi - mu[i]*mu[i])*(2*mu[i] - sin(2*mu[i])))
    return suma + u0

def Experiment_Fourier(mu, l, k,c,x,t,N,u0):
    suma = 0
    for i in range(N,len(mu)):
        suma = suma + ((4*pi*mu[i]*sin(mu[i]))*sin(mu[i]*x/l)* exp((-
k*mu[i]*mu[i]*t)/(c*l*1)))/((pi*pi - mu[i]*mu[i])*(2*mu[i] - sin(2*mu[i])))
    return suma + u0

if err == True:
    arr_x = np.linspace(0, l, 350).tolist()
    result_arr = []
    for elem in arr_x:
        result_arr.append(Fourier_series(mu, l, k, c, elem, t, n, u0))

plt = pg.plot(arr_x, result_arr, title = 'Распределение температуры в стержне')
plt.setLabel('left', 'u(x,t)')

```

```

        plt.setLabel('bottom', 'Длина стержня')
        plt.showGrid(True, True)

def buttonClicked(self):
    print('Close button pressed')
    import sys
    sys.exit(0)

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Temperature_in_rod = QtWidgets.QMainWindow()
    ui = Ui_Temperature_in_rod()
    ui.setupUi(Temperature_in_rod)
    Temperature_in_rod.show()
    sys.exit(app.exec_())

```