

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Исследование численных алгоритмов в обратных задачах теплопроводности»

Обучающийся

В.С. Юматов

(Инициалы Фамилия)



(личная подпись)

Руководитель

д.ф.-м.н, доцент, С.В. Талалов

ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н, доцент, А.В. Егорова

ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Название выпускной работы: «Исследование численных алгоритмов в обратных задачах теплопроводности».

Работа выполнена в объеме 50 страниц, включая 17 иллюстраций и 2 таблиц. Структура включает введение, три главы, заключение и список использованной литературы.

Основным вопросом работы является исследование и сравнение двух численных методов - метода решения системы линейных алгебраических уравнений (СЛАУ) и алгоритма Ландвебера - для решения обратных задач теплопроводности, которые имеют широкое применение в науке и технике.

Цель работы заключается в разработке и исследовании численных алгоритмов для решения обратных задач теплопроводности, а также в реализации этих алгоритмов в виде программного обеспечения.

Выпускная квалификационная работа может быть структурирована следующим образом: анализ существующих численных методов для решения обратных задач теплопроводности, программная реализация выбранных методов, сравнительный анализ их эффективности и практическое применение полученных результатов.

В завершении работы представлен обзор результатов сравнительного анализа, показавшего, что метод СЛАУ более точен, но требует больших вычислительных ресурсов, тогда как алгоритм Ландвебера более стабилен в использовании памяти и гибок в настройке точности. Работа подчеркивает важность выбора метода в зависимости от конкретных условий и требований задачи.

Abstract

The title of the graduation work is «Study of Numerical Algorithms in Inverse Heat Conduction Problems».

The graduation work consists of 50 pages, including 17 illustrations and 2 tables. The structure includes an introduction, three chapters, a conclusion, and a list of references.

The key issue addressed in the graduation work is the investigation and comparison of two numerical methods - the method for solving systems of linear algebraic equations (SLAE) and the Landweber algorithm - for solving inverse heat conduction problems, which have wide applications in science and engineering.

The aim of the work is to develop and investigate numerical algorithms for solving inverse heat conduction problems, as well as to implement these algorithms as software.

The graduation work can be structured as follows: analysis of existing numerical methods for solving inverse heat conduction problems, software implementation of the selected methods, comparative analysis of their effectiveness, and practical application of the results obtained.

The conclusion of the work presents an overview of the comparative analysis results, demonstrating that the SLAE method is more accurate but requires significant computational resources, while the Landweber algorithm is more stable in memory usage and flexible in accuracy settings. The work emphasizes the importance of selecting the appropriate method based on specific task conditions and requirements.

Оглавление

Введение.....	5
Глава 1 Теоретические основы обратных задач теплопроводности	7
1.1 Уравнение теплопроводности и граничные условия	7
1.2 Обратные задачи теплопроводности	10
1.3 Метод сведения задачи к интегральному уравнению Фредгольма I рода.....	15
Глава 2 Численные методы решения интегральных уравнений Фредгольма I рода	19
2.1 Проблема некорректности задачи и её решение	19
2.2 Аппроксимация ядра интегрального уравнения вырожденным ядром	23
2.3 Алгоритм Ландвебера.....	26
Глава 3 Программная реализация и сравнительный анализ численных алгоритмов	30
3.1 Программная реализация численных алгоритмов.....	30
3.2 Сравнение результатов, полученных различными методами	38
Заключение	46
Список используемой литературы	48
Приложение А Листинг программного кода.....	50

Введение

Обратные задачи теплопроводности являются важной областью исследования в прикладной математике и теплофизике. Эти задачи включают восстановление неизвестных параметров или начальных условий на основе измерений температуры, сделанных на границе или внутри тела. История изучения этих задач насчитывает несколько десятилетий, и за это время были разработаны различные численные методы для их решения, включая методы регуляризации и итерационные алгоритмы.

Актуальность исследования численных алгоритмов в обратных задачах теплопроводности обусловлена широким применением этих задач в различных областях науки и техники, таких как геофизика, медицина, инженерия и материаловедение. Решение таких задач позволяет, например, восстанавливать внутренние структуры объектов по тепловым данным, что имеет большое практическое значение для неразрушающего контроля и диагностики.

Цель данной работы заключается в исследовании численных алгоритмов для решения обратных задач теплопроводности, а также в реализации этих алгоритмов в виде программного обеспечения.

Для достижения этой цели необходимо решить следующие задачи:

- изучить теоретические основы уравнений теплопроводности и граничных условий;
- исследовать методы решения обратных задач теплопроводности и их математическое обоснование;
- рассмотреть методы сведения обратных задач к интегральным уравнениям Фредгольма I рода и их аппроксимацию;
- изучить проблему некорректности задач и методы её решения;
- разработать и реализовать численные алгоритмы для решения обратных задач, включая алгоритм Ландвебера;

- провести тестирование и анализ разработанных алгоритмов на модельных задачах;
- оценить практическую значимость полученных результатов и возможности их применения.

Объектом исследования являются обратные задачи теплопроводности, связанные с восстановлением неизвестных параметров или начальных условий по данным наблюдений.

Предметом исследования являются численные методы и алгоритмы для решения этих задач, включая методы регуляризации и итерационные алгоритмы.

Работа состоит из трех глав.

В первой главе рассматриваются теоретические основы теплопроводности, уравнение теплопроводности и граничные условия, обратные задачи теплопроводности, а также метод сведения задачи к интегральному уравнению Фредгольма I рода.

Во второй главе анализируется проблема некорректности задач и методы её решения, аппроксимация ядра интегрального уравнения вырожденным ядром, а также подробно рассматривается алгоритм Ландвебера.

Третья глава посвящена программной реализации численных алгоритмов, включая разработку программного обеспечения, а также анализ результатов тестирования и их интерпретацию.

Глава 1 Теоретические основы обратных задач теплопроводности

1.1 Уравнение теплопроводности и граничные условия

Общее уравнение теплопроводности описывает процесс распределения температуры в пространстве и времени. Оно является фундаментальным в теории теплопроводности и в пространстве с произвольной системой координат $r = (r_1, r_2, \dots, r_n)$ уравнение теплопроводности имеет следующий вид [6]:

$$\frac{\partial u}{\partial t} - \alpha^2 \Delta u = f(r, t), \quad (1)$$

где $u(x, t)$ – температура в точке x в момент времени t ;

$\Delta = \nabla^2$ – оператор Лапласа;

α^2 – коэффициент теплопроводности;

$f(r, t)$ – функция тепловых источников.

В декартовом пространстве с координатами $x = (x_1, x_2, \dots, x_n)$ уравнение теплопроводности принимает вид [6]:

$$\frac{\partial u}{\partial t} - \alpha^2 \left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \dots + \frac{\partial^2 u}{\partial x_n^2} \right) = f(x, t). \quad (2)$$

Если $f(x, t) \equiv 0$, то уравнение называется однородным, в противном случае – неоднородным [6].

Для более глубокого понимания уравнения теплопроводности, важно рассмотреть физическую интерпретацию его компонентов. Уравнение теплопроводности моделирует процесс распространения тепла в однородном

материале, основываясь на законе Фурье о теплопроводности. Согласно закону Фурье, поток тепла пропорционален градиенту температуры [4]:

$$q = -k \frac{\partial u}{\partial x}, \quad (3)$$

где q – плотность теплового потока;

k – коэффициент теплопроводности материала.

Уравнение теплопроводности можно получить, объединяя закон сохранения энергии и закон Фурье. Рассмотрим элементарный объем в материале. Изменение внутренней энергии в этом объеме равно разности теплового потока, входящего и выходящего из объема [3]:

$$\rho c \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \right), \quad (4)$$

где ρ – плотность материала;

c – удельная теплоемкость.

Для однородного материала с постоянными коэффициентами уравнение принимает упрощенный вид [2]:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad (5)$$

где α – коэффициент температуропроводности.

Для данной задачи рассматривается частный пример уравнения теплопроводности следующего вида:

$$\frac{\partial \psi}{\partial t} = \frac{\partial^2 \psi}{\partial x^2}, \quad (6)$$

где $\psi(x, t)$ – температура в точке x в момент времени t .

Граничные условия играют ключевую роль в определении конкретного решения уравнения теплопроводности [4]. Рассмотрим типичные граничные условия для задачи о теплопроводности стержня:

- дирихле: температура фиксирована на концах стержня [19]

$$u(0, t) = u_0, \quad u(L, t) = u_L; \quad (7)$$

- неймана: тепловой поток фиксирован на концах стержня [19]

$$\left. \frac{\partial u}{\partial x} \right|_{x=0} = q_0, \quad \left. \frac{\partial u}{\partial x} \right|_{x=L} = q_L; \quad (8)$$

- робин: комбинированные условия, связывающие температуру и тепловой поток [23]

$$k \left. \frac{\partial u}{\partial x} \right|_{x=0} + hu(0, t) = T_0, \quad k \left. \frac{\partial u}{\partial x} \right|_{x=L} + hu(L, t) = T_L. \quad (9)$$

Рассматриваемая область исследования задается как $\Omega = [0, 1] \times [0, \infty)$, $x \in [0, 1]$ и $t \in [0, \infty)$. Это означает, что исследуемая область представляет собой одномерный стержень длиной 1 единица, и процесс теплопередачи рассматривается для всех моментов времени от нуля до бесконечности.

Граничные условия для данной задачи задаются следующим образом:

$$\psi(t, 0) = \psi(t, b) = 0 \quad \forall t \geq 0. \quad (10)$$

Эти условия означают, что температура на концах стержня всегда остается нулевой, что может соответствовать идеализации ситуации, когда концы стержня поддерживаются при постоянной нулевой температуре.

Решение уравнения теплопроводности в момент времени $t = T$ обозначается как $\varphi(T, x) = \psi(T, x)$. В данном случае функция $\varphi(T, x)$ определяется следующим выражением:

$$\psi(T, x) = \varphi_T(x) \equiv 4x(1 - x) + \sin(2\pi x^2)^2. \quad (11)$$

Данная функция $\varphi(T, x)$ известна с относительной точностью ε по норме пространства $L_\infty[0,1]$. Это означает, что измерения температуры имеют определенную погрешность, и задача заключается в восстановлении исходного распределения температуры $\varphi(x)$ при $t=0$.

Таким образом, задача сводится к нахождению функции $\varphi(x)$ на основе известного значения $\varphi(T, x)$. Это представляет собой типичную обратную задачу теплопроводности, в которой по конечному состоянию системы необходимо определить её начальное состояние.

1.2 Обратные задачи теплопроводности

Обратные задачи теплопроводности (ОЗТ) представляют собой класс задач математической физики, в которых необходимо определить предыдущее состояние системы на основе известных данных о её текущем или конечном состоянии [11]. В отличие от прямых задач, где по известным начальным и граничным условиям вычисляется будущее состояние системы, обратные задачи являются существенно более сложными и часто некорректными, так как малые изменения в исходных данных могут приводить к значительным отклонениям в решении [16].

ОЗТ имеют широкое применение и важность в различных областях техники и науки [15]. Все обратные задачи можно разделить на три основных класса, независимо от рассматриваемого физического процесса или технической системы:

- диагностика и идентификация физических процессов: Эти задачи связаны с экспериментальными исследованиями и направлены на восстановление входных параметров по измеренным выходным характеристикам. Такие задачи являются основополагающими для построения математических моделей и других типов обратных задач [15];
- проектирование технических объектов: в этих задачах определяются проектные характеристики по заданным качественным показателям и ограничениям. Искомые характеристики являются причинными относительно показателей и ограничений [15];
- управление процессами и объектами: здесь управляющие воздействия изменяют состояние системы, и эти воздействия являются причинными характеристиками [15].

Между задачами диагностики и идентификации и задачами проектирования и управления существует принципиальное различие. В задачах проектирования и управления расширение класса допустимых решений упрощает ситуацию, поскольку требуется найти технически реализуемое решение, обеспечивающее экстремум критерия качества с заданной точностью. Для задач идентификации и диагностики, наоборот, чем шире класс возможных решений, тем сложнее задача из-за увеличения погрешностей определения причинных характеристик.

Далее рассмотрим виды обратных задач теплопроводности.

Основные формы теплообмена включают теплопроводность, конвективный и радиационный теплообмен. В рамках каждого из этих видов теплообмена можно выделить следующие виды обратных задач:

- граничные задачи: в этих задачах определяется нахождение функций и параметров, входящих в граничные условия [4];
- коэффициентные задачи: здесь находят функции и параметры, входящие в коэффициенты уравнений [4];

- ретроспективные задачи: эти задачи направлены на нахождение начальных условий [4];
- геометрические задачи: в этих задачах происходит реконструирование геометрических характеристик области или каких-либо характерных точек, линий, поверхностей внутри этой области [4].

Далее распишем области применения обратных задач. Начнем с проектирования.

Оптимизация проектных параметров с учетом тепловых ограничений базируется на математических моделях, которые связывают проектные характеристики с переменными состояниями и различными воздействиями. Например, оптимальное тепловое проектирование систем рассматривается как обратная задача теплообмена: по известным условиям, определяющим допустимое тепловое состояние системы, находятся требуемые причинные характеристики, удовлетворяющие этому состоянию и критерию оптимальности системы.

Далее рассмотрим практическое применение:

- теплозащитные конструкции: определение условий теплообмена или подбор толщин покрытия сводится к проектным обратным задачам теплопроводности [2];
- экспериментальные исследования и испытания: определение температур, плотностей тепловых потоков и коэффициентов теплоотдачи требует решения граничных обратных задач теплопроводности [2];
- системы тепловой защиты: исследование процессов разрушения и прогрева материалов, а также изучение тепловых эффектов вдува газообразных продуктов разрушения материала [2];

- восстановление температурных полей: решение задач теплопроводности для анализа термопрочности материалов и конструкций минимизирует погрешности измерений [2];
- исследование массообменных процессов: обработка экспериментальных данных с использованием обратных задач теплопроводности;
- идентификация и корректировка математических моделей: определение теплофизических характеристик материалов и их корректировка для точного расчета температурных полей в теплозащитных покрытиях;
- определение тепловых контактных сопротивлений: важность для конструкций с множеством соединений, определение контактных сопротивлений в многослойных покрытиях.

В рассматриваемой задаче необходимо восстановить начальное распределение температуры $\varphi(x)$ при $t=0$ по известному распределению температуры $\varphi(T, x)$ в момент времени $t = T$. Формально, задача формулируется следующим образом:

$$\left\{ \begin{array}{l} \frac{\partial \psi}{\partial t} = \frac{\partial^2 \psi}{\partial x^2}, \quad x \in [0,1], t \in [0, \infty) \\ \psi(T, x) = \varphi(T, x), \quad x \in [0,1] \\ \psi(t, 0) = \psi(t, 1), \quad t \geq 0 \end{array} \right. , \quad (12)$$

где $\varphi(T, x)$ задается выражением (11).

Основная цель – найти начальное распределение температуры $\psi(0, x) = \varphi(x)$.

Обратные задачи теплопроводности являются некорректно поставленными задачами в смысле Адамара, так как их решение не всегда существует, не обязательно единственно и не всегда устойчиво относительно небольших возмущений исходных данных. Это требует применения методов

регуляризации для стабилизации решения и получения приемлемых результатов [1].

Некорректность обратных задач теплопроводности приводит к следующим основным трудностям:

- чувствительность к шуму данных: малые ошибки в измерении конечного состояния могут приводить к большим ошибкам в восстановленном начальном состоянии [18];
- неустойчивость решений: малые изменения в входных данных могут вызвать значительные изменения в результатах [18];
- множественность решений: для некоторых данных может существовать более одного решения, что требует введения дополнительных критериев для выбора, наиболее подходящего [18].

Для преодоления этих трудностей используются различные методы, ниже перечислены некоторые из них.

Регуляризация Тихонова: введение дополнительного члена в функционал, минимизирующего отклонение решения от некоторого априорного предположения [9].

Метод усеченных сингулярных разложений: ограничение числа учитываемых сингулярных значений при решении уравнения [10].

Алгоритм Ландвебера: итерационный метод, использующий априорные предположения для стабилизации решения.

Упрощенная регуляризация: частный метод регуляризации Тихонова, применяющийся для стабилизации решений плохо обусловленных систем.

В данной работе будут рассмотрены два подхода к решению обратной задачи теплопроводности:

- аппроксимация ядра интегрального уравнения Фредгольма I рода вырожденным ядром и сведение задачи к решению системы линейных алгебраических уравнений (СЛАУ);

– использование алгоритма Ландвебера для итеративного восстановления начального состояния системы.

1.3 Метод сведения задачи к интегральному уравнению Фредгольма I рода

Обратные задачи теплопроводности являются важной областью исследований, так как они позволяют определить распределение температуры внутри объекта или материала на основе данных о его теплопроводности и граничных условиях. Они имеют широкое применение в различных областях, включая инженерию, медицину, геологию и многие другие [5].

Метод сведения обратной задачи теплопроводности к интегральному уравнению Фредгольма I рода представляет собой один из эффективных подходов к их решению. Этот метод позволяет сформулировать задачу в терминах интегрального уравнения, что обеспечивает математическую основу для применения различных методов анализа и численных методов [20].

Интегральные уравнения Фредгольма I рода выражают связь между неизвестной функцией (распределением температуры) и известными данными через интегральные выражения [22]. Они широко используются в науке и инженерии для моделирования различных процессов, включая теплопроводность.

В связи с этим, важно рассмотреть собственные функции (СФ) и собственные числа (СЧ) оператора K с ядром:

$$K(x, y) = \frac{2}{b} \sum_{n=1}^{\infty} \exp \left[-a \left(\frac{\pi n}{b} \right)^2 T \right] \sin \left(\frac{\pi n x}{b} \right) \sin \left(\frac{\pi n y}{b} \right). \quad (13)$$

Эти СФ и СЧ можно определить следующим образом:

$$\varphi_n(x) = \sqrt{\frac{2}{b}} \left(\frac{\pi n x}{b} \right), \quad (14)$$

$$\lambda_n = \exp \left[-a \left(\frac{\pi n}{b} \right)^2 T \right], \quad (15)$$

Использование этих спектральных характеристик позволяет более эффективно анализировать и решать интегральные уравнения [13].

Преимущества метода сведения включают его математическую стройность и возможность использования мощных методов теории интегральных уравнений для анализа и решения задачи. Кроме того, этот метод обеспечивает основу для разработки численных алгоритмов, которые могут быть применены для получения численных решений с высокой точностью и эффективностью [23].

Также уравнения могут быть эффективно решены с помощью их дискретизации, что приводит к системе линейных алгебраических уравнений (СЛАУ) [17].

Решение систем линейных алгебраических уравнений является классической задачей вычислительных методов. Особенно интересны разреженные матрицы, которые возникают в результате конечно-разностной или конечно-элементной аппроксимации. Эти матрицы необходимо хранить в специализированных форматах, поскольку даже при аппроксимации небольших задач размерность матрицы системы может стать непосильной для современных параллельных компьютеров [21].

Обратная задача теплопроводности сводится к решению системы линейных алгебраических уравнений (СЛАУ), которая зависит от типа ядра интегрального оператора, используемого при формулировке задачи.

В случае вырожденного ядра интегрального оператора обратная задача теплопроводности решается путем редукции к СЛАУ, где матрица системы является вырожденной. Это означает, что система уравнений может быть

недостаточно обратимой, и для её решения может потребоваться использование специализированных методов или регуляризации.

Если ядро интегрального оператора невырожденное, обратная задача теплопроводности сводится к СЛАУ с невырожденной матрицей. Это означает, что система уравнений имеет единственное решение, которое может быть найдено стандартными методами решения СЛАУ, такими как метод Гаусса или LU-разложение [17].

В обоих случаях ключевым шагом является формирование и аппроксимация интегрального оператора, что определяет конечную размерность и структуру СЛАУ. Для вырожденного ядра требуется особое внимание к методам регуляризации или альтернативным методам численного решения, чтобы обеспечить стабильность и точность решения задачи.

Решение $\psi(t, x)$ уравнения (6) с граничными условиями (10) может быть представлено через функцию Грина для однородного уравнения теплопроводности. Пусть $G(x, t; \xi)$ – функция Грина для уравнения теплопроводности, удовлетворяющая тем же граничным условиям. Тогда решение можно выразить как [14]:

$$\psi(x, t) = \int_0^1 G(x, t; \xi) \varphi(\xi) d\xi. \quad (16)$$

Функция Грина для одномерного уравнения теплопроводности с нулевыми граничными условиями имеет вид [14]:

$$G(x, t; \xi) = \sum_{n=1}^{\infty} e^{-n^2 \pi^2 t} \sin(n\pi x) \sin(n\pi \xi). \quad (17)$$

Используя это представление, решение в момент времени $t = T$ будет:

$$\varphi(T, x) = \psi(T, x) = \int_0^1 G(x, t; \xi) \varphi(\xi) d\xi. \quad (18)$$

Это выражение представляет собой интегральное уравнение Фредгольма I рода:

$$\varphi(T, x) = \int_0^1 K(x, \xi) \varphi(\xi) d\xi, \quad (19)$$

где ядро $K(x, \xi)$ определяется как:

$$K(x, \xi) = \sum_{n=1}^{\infty} e^{-n^2 \pi^2 t} \sin(n\pi x) \sin(n\pi \xi). \quad (20)$$

Теперь задача сводится к нахождению функции $\varphi(\xi)$ из интегрального уравнения (16), что требует применения численных методов, так как аналитическое решение в общем случае найти сложно.

Выводы по главе 1

Первая глава посвящена теоретическим основам обратных задач теплопроводности. В ней рассматриваются основные уравнения теплопроводности и граничные условия, которые являются фундаментальными для понимания распределения температуры в пространстве и времени. Также подробно освещены методы сведения обратных задач к интегральным уравнениям Фредгольма I рода, что позволяет применять численные методы для их решения. Важным аспектом главы является обсуждение некорректности задач теплопроводности и методов их решения, что является ключевым для получения стабильных и точных результатов.

Глава 2 Численные методы решения интегральных уравнений Фредгольма I рода

2.1 Проблема некорректности задачи и её решение

Обратные задачи теплопроводности являются классическим примером некорректных задач, что влечет за собой ряд трудностей при их решении. Некорректность в данном контексте означает, что задача может не иметь единственного решения, решение может не существовать или быть неустойчивым по отношению к малым возмущениям входных данных [7]. Основные проблемы, связанные с некорректностью задач, которые были рассмотрены в предыдущей главе.

Для решения этих проблем используются различные методы регуляризации, направленные на стабилизацию решения и улучшение его устойчивости к шумам и возмущениям [8]. Перечислим основные методы регуляризации.

Регуляризация Тихонова: Этот метод добавляет к функционалу минимизации дополнительный член, контролирующий гладкость решения [18]. Формально метод можно записать как:

$$\min_{\varphi} \{ \|A\varphi - \varphi_{\text{наб}}\|^2 + \alpha \|\varphi\|^2 \}, \quad (21)$$

где A – оператор, представляющий интегральное уравнение;

$\varphi_{\text{наб}}$ – наблюдаемое распределение температуры;

α – параметр регуляризации.

Метод усеченных сингулярных разложений: Этот метод ограничивает количество учитываемых сингулярных значений оператора A , что позволяет отфильтровать высокочастотные компоненты шума [24]. Решение представляется в виде:

$$\varphi \approx \sum_{i=1}^k \frac{\langle \varphi_{\text{наб}}, u_i \rangle}{\sigma_i} v_i, \quad (22)$$

где u_i и v_i – сингулярные векторы;

σ_i – сингулярные значения;

k – выбранное число сингулярных значений.

Итерационные методы: Один из популярных итерационных методов – алгоритм Ландвебера, который позволяет последовательно приближаться к решению, учитывая априорные данные и шумы в исходных данных [25]. Итерационная формула имеет вид:

$$\varphi^{(n+1)} = \varphi^n + \tau A^*(\varphi_{\text{наб}} - A\varphi^n), \quad (23)$$

где φ^n – приближение решения на n -ой итерации;

τ – шаг итерации;

A^* – сопряженный оператор к A .

Упрощенная регуляризация. Является частным методом регуляризации Тихонова и применяется для стабилизации решений плохо обусловленных систем, имеет вид:

$$A + \alpha I = b, \quad (24)$$

где I – единичная матрица того же размера, что и A ;

α – малый положительный параметр регуляризации.

Рассмотрим некоторые проблемы, возникающие при нахождении решений СЛАУ $Ax = z$ высокого порядка.

Здесь: $A = A_{ij}$ – матрица размерности $N \times N$ (число N считается достаточно большим), x – искомый вектор N -мерного евклидова пространства E_N , z – некоторый заданный вектор данного пространства.

На практике правая часть всегда известна с некоторой погрешностью $\delta z = (\delta z_1, \dots, \delta z_N)$, где z_i – компоненты вектора z .

В качестве относительной погрешности, принимают величину $\epsilon_z = \frac{\|\delta z\|}{\|z\|}$.

Элементы матрицы A так же, как правило заданы с некоторой (относительной) погрешностью $\epsilon_A = \frac{\max_{ij} \|\delta A_{ij}\|}{\|A\|}$.

При этом естественной выглядит постановка задачи нахождения искомого вектора x с погрешностью $\epsilon_x \simeq \max(\epsilon_z, \epsilon_A)$.

Для невырожденных матриц имеем:

$$x = A^{-1}z. \quad (25)$$

Проблема заключается в том, что на практике при больших N часто приходится иметь дело с «почти» вырожденными матрицами, т.е. такими матрицами A , для которых $\det A \simeq 0$.

Однако тот факт, что определитель матрицы, решаемой СЛАУ «почти» нулевой, не всегда означает, что погрешность ϵ_x будет значительно больше, чем величина $\max(\epsilon_z, \epsilon_A)$.

Мерой, определяющей отклонение величины ϵ_x от величин ϵ_z и ϵ_A , является число обусловленности $Cond(A)$ матрицы A , которое определяется как:

$$cond(A) = \|A\| \|A^{-1}\|, \quad (26)$$

где $\|A\|$ – норма матрицы A ;

$\|A^{-1}\|$ – норма обратной матрицы.

Для эрмитовых матриц используется отношение наибольшего и наименьшего собственных значений матрицы A :

$$\text{cond}(A) = \frac{\sigma_{max}}{\sigma_{min}}, \quad (27)$$

где σ_{max} и σ_{min} – максимальное и минимальное собственные значения матрицы соответственно.

Пусть, например, $\delta A_{ij} = 0$. Из определения нормы имеем неравенства

$$\|z\| \leq \|A\| \times \|x\|, \quad \|\delta x\| \leq \|A^{-1}\| \times \|\delta z\|. \quad (28)$$

В итоге находим $\epsilon_x \leq \text{Cond}(A) \times \epsilon_z$.

Пусть теперь $\delta z = 0$, $\delta A_{ij} \neq 0$. Имеем:

$$z = Ax, z = (A + \delta A)(x + \delta x). \quad (29)$$

Откуда находим $\delta x = -A^{-1}\delta A(x + \delta x)$. Отсюда получаем:

$$\epsilon_x \simeq \frac{\|\delta x\|}{\|x + \delta x\|} \leq \text{Cond}(A) \times \epsilon_A. \quad (30)$$

Матрицы, для которых $\text{Cond}(A) \gg 1$, называют плохо обусловленными.

Видно, что плохо обусловленные матрицы коэффициентов приводят при решении СЛАУ к потере устойчивости. Это означает, что здесь пример некорректной задачи [12].

Решение таких задач есть всегда некоторый компромисс между устойчивостью и точностью.

Для решения этой проблемы применяется метод упрощенной регуляризации. Рассмотрим вместо системы $Ax = z$ систему

$$(\alpha I + A)x = z, \quad (31)$$

где I – единичная матрица и $\alpha > 0$.

При достаточно малых α новая система отличается от исходной.

Для коэффициентов $c_i = (x, w_i)$ теперь имеем «исправленное» выражение $c_i = \frac{b_i}{(\alpha + \lambda_i)}$.

Таким образом, малый параметр α оказывает влияние именно на малые, то есть «плохие», собственные числа (СЧ), частично исправляя ситуацию.

Выбор конкретного значения числа α зависит от задачи и определяет тот самый компромисс, о котором говорилось выше.

2.2 Аппроксимация ядра интегрального уравнения вырожденным ядром

Аппроксимация ядра интегрального уравнения Фредгольма первого рода методом вырожденного ядра позволяет свести решение интегрального уравнения к решению системы линейных алгебраических уравнений (СЛАУ). Рассмотрим интегральное уравнение Фредгольма первого рода:

$$\varphi(x) = \int_a^b K(x, y)\varphi(y) dy, \quad (32)$$

где $K(x, y)$ – ядро интегрального уравнения;

$\varphi(x)$ – неизвестная функция;

Далее будут рассмотрены шаги аппроксимации:

1. Ядро $K(x, y)$ представляется в виде конечного ряда:

$$K(x, y) = \sum_{i=1}^N g_i(x)h_i(y), \quad (33)$$

где $g_i(x)$ и $h_i(y)$ – известные функции.

2. Вырожденное ядро подставляется в исходное интегральное уравнение:

$$\varphi(x) = \sum_{i=1}^N g_i(x) \int_a^b h_i(y)\varphi(y)dy. \quad (34)$$

3. Вводятся обозначения:

$$c_j = \int_a^b h_j(y)\varphi(y)dy, \quad (35)$$

$$f_j = \int_a^b h_j(y)f(y)dy, \quad (36)$$

$$K_{ij} = \int_a^b h_j(y)g_i(y)dy. \quad (37)$$

Тогда уравнение можно записать в виде СЛАУ:

$$c_j = \sum_{i=1}^N K_{ji} c_i + f_j. \quad (38)$$

4. Решается система линейных уравнений для нахождения коэффициентов c_j . После нахождения c_j , они подставляются в выражение для $\varphi(x)$:

$$\varphi(x) = \sum_{i=1}^N g_i(x)c_i. \quad (39)$$

Пусть \hat{K} – некоторый компактный оператор в гильбертовом пространстве H . Тогда $\forall \epsilon > 0 \exists N = N(\epsilon)$, такое, что

$$\hat{K} = \hat{K}_N + \hat{K}_\epsilon, \quad (40)$$

где \hat{K}_N – конечномерный оператор, действующий в N -мерном пространстве $V_N \subset H$ и $\|\hat{K}_\epsilon\| \leq \epsilon$.

Выбор числа N напрямую связан с требуемой точностью ϵ . Точность ϵ определяет допустимую погрешность в конечной температуре, что необходимо учитывать при аппроксимации. Чем меньше ϵ , тем большее количество членов N потребуется для достижения данной точности. Таким образом, N выбирается так, чтобы ошибка аппроксимации не превышала ϵ :

$$\left| \varphi(x) - \sum_{i=1}^N g_i(x)c_i \right| < \epsilon, \quad (41)$$

Процесс выбора N заключается в следующем:

1. Начальное значение N определяется на основе грубой оценки, исходя из физических соображений и начальных условий задачи.
2. Для заданного ϵ производится численный расчет и сравнение результатов аппроксимации с известными аналитическими решениями или экспериментальными данными.
3. Если ошибка превышает ϵ , значение N увеличивается до тех пор, пока ошибка не станет меньше ϵ .

Таким образом, параметр ϵ используется для контроля точности аппроксимации и напрямую влияет на выбор N . Это обеспечивает необходимую точность в вычислениях и гарантирует соответствие численных решений заданным требованиям.

Решение исходного интегрального уравнения сводится к решению СЛАУ.

2.3 Алгоритм Ландвебера

Алгоритм Ландвебера является итерационным методом для решения некорректных задач, таких как интегральные уравнения Фредгольма первого рода. Он направлен на нахождение приближенного решения уравнения вида

$$\widehat{K}y = f, \quad (42)$$

где \widehat{K} – компактный оператор;

y – неизвестная функция;

f – заданная функция.

Алгоритм Ландвебера также может быть улучшен путем использования собственных функций и собственных чисел оператора:

$$\varphi(x) = \sum_{s=1}^N \Omega_s c_s \varphi_s(x), \quad (43)$$

$$c_i = \int_0^b \varphi_i(x) \varphi_\tau(x) dx, \quad (44)$$

где $\Omega_s = \frac{1}{\lambda_s} = \exp \left[a \left(\frac{\pi s}{b} \right)^2 T \right]$. Этот подход позволяет более эффективно стабилизировать процесс итераций и улучшить сходимость к точному решению.

Алгоритм основан на методе градиентного спуска и обеспечивает устойчивость и сходимость к точному решению при правильном выборе параметров.

Прибавим к левой и правой частям уравнения (42) y . В итоге получим уравнение:

$$y = f + (\hat{I} - \hat{K})y. \quad (45)$$

Решением данного уравнения является формальный ряд:

$$y = \sum_{i=0}^N (\hat{I} - \hat{K})^i f. \quad (46)$$

Данный ряд может быть получен при помощи итерационной процедуры

$$y^{(n+1)} = f + (\hat{I} - \hat{K})y^{(n)}, \quad (47)$$

$$y^{(0)} = f. \quad (48)$$

Можно показать, что ряд сходится, если исходное уравнение разрешимо.

Обрываем ряд (46) на конечном числе слагаемых:

$$y^{(n+1)} = \sum_{i=0}^n (\hat{I} - \hat{K})^i f. \quad (49)$$

Подставим в правую часть (49) разложение:

$$f = \sum_{i=0}^N c_i w_i, \quad (50)$$

где w_i – собственная функция(СФ) оператора \widehat{K} и $c_i = (f, w_i)$.

В итоге находим:

$$y^{(n+1)} = \sum_{k=1}^N c_k w_k \sum_{i=0}^n (1 - \lambda_k)^i = \sum_{k=1}^N \frac{1}{\lambda_k} [1 - (1 - \lambda_k)^{(n+1)}] c_k w_k. \quad (51)$$

Таким образом, мы получили приближенное решение:

$$y^{(n+1)} \simeq y. \quad (52)$$

Которое означает использование передаточной функции

$$\Omega_k^{(n)} = \frac{1}{\lambda_k} [1 - (1 - \lambda_k)^{(n+1)}]. \quad (53)$$

Это означает, что можно записать явное решение

$$\varphi(x) = \sum_{s=1}^N \Omega_s c_s \varphi_s(x), \quad c_i = \int_0^b \varphi_i(x) \varphi_T(x) dx. \quad (54)$$

Далее можно считать, что $|\lambda_k| < 1$.

Поэтому мы видим, что при $n \rightarrow \infty$ выполняется $\Omega_k \rightarrow 1/\lambda_k$, так что решение стремится к точному.

Выбор числа n в данной формуле – это всегда компромисс между точностью и устойчивостью.

Выводы по главе 2

Вторая глава посвящена численным методам решения интегральных уравнений Фредгольма I рода, которые применяются для решения обратных задач теплопроводности. Основное внимание уделено проблеме некорректности этих задач, что часто приводит к нестабильным и неточным решениям. Рассматриваются различные методы регуляризации, такие как метод Тихонова, усеченные сингулярные разложения и итерационные методы, включая алгоритм Ландвебера. Каждый метод подробно анализируется с точки зрения его эффективности в стабилизации решения и уменьшении чувствительности к шуму данных. В главе также обсуждается аппроксимация ядра интегрального уравнения вырожденным ядром, что позволяет свести задачу к решению системы линейных алгебраических уравнений (СЛАУ). Этот подход обеспечивает основу для разработки численных алгоритмов, которые могут быть применены для получения точных и устойчивых решений. Глава подчеркивает важность выбора правильного численного метода и параметров регуляризации для конкретных задач теплопроводности, что является ключевым для достижения надежных и практических результатов.

Глава 3 Программная реализация и сравнительный анализ численных алгоритмов

3.1 Программная реализация численных алгоритмов

Текущий раздел посвящен программной реализации численных алгоритмов, что является важным аспектом при решении сложных задач, связанных с теплопроводностью. Программная реализация позволяет не только проверять теоретические модели, но и проводить многочисленные вычислительные эксперименты, анализировать поведение различных численных методов в различных условиях, а также сравнивать их эффективность и точность.

Современные подходы к программной реализации численных алгоритмов включают использование мощных библиотек и инструментов, таких как NumPy и Matplotlib, которые обеспечивают высокую производительность и удобство работы с данными. В данной работе реализуются два основных численных метода: численный метод аппроксимации ядра и алгоритм Ландвебера, предназначенные для решения обратных задач теплопроводности. Эти методы позволяют стабилизировать процесс решения и обеспечить надежные результаты при решении некорректных задач.

В дальнейшем тексте будет представлен план анализа программы, выполненной на языке Python, включающий разбор кода, блок-схему алгоритма и детальное описание каждого блока программного кода. В заключении будут подведены итоги проведенного анализа и оценка эффективности использованных методов.

План работы программы состоит из:

- Введение;
- инициализация данных;

- дискретизация и построение матрицы;
- регуляризация матрицы и формирование вектора правой части;
- решение задачи с использованием аппроксимации;
- реализация алгоритма Ландвебера;
- визуализация результатов;
- заключение.

Далее будет представлена блок-схемы двух алгоритмов программы, которые наглядно демонстрируют основные этапы численных расчетов, начиная с инициализации данных и заканчивая визуализацией полученных результатов. Это позволит более детально рассмотреть каждый шаг программы и понять взаимосвязь между различными блоками кода.

На рисунке 1 представлена блок схема работы алгоритма Ландвебера.

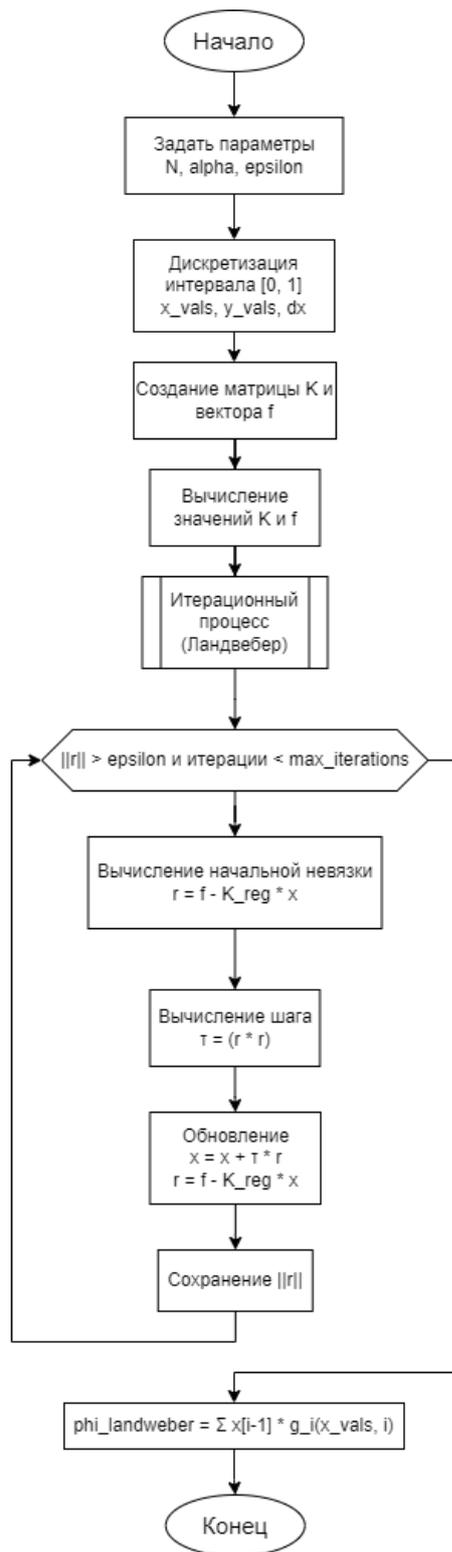


Рисунок 1 – Блок -схема работы алгоритма Ландвебера

Далее на рисунке 2 продемонстрируем блок-схему работы численного метода аппроксимации ядра.



Рисунок 2 – Блок-схема работы численного метода аппроксимации ядра

Далее на рисунках 3, 4, 5, 6, 7, 8, 9, 10 представлен программный код и алгоритма Ландвебера и решение численного метода аппроксимации ядра. Полный листинг программного кода представлен в Приложении А.

```
import numpy as np
import matplotlib.pyplot as plt
```

Рисунок 3 – Импорт библиотек

В данном программном коде numpy импортируется как np для работы с массивами. Matplotlib.pyplot импортируется как plt для построения графиков.

```
# Задаем количество членов N для аппроксимации с вырожденным ядром
N = 100
alpha_approx = 0.01 # параметр для регуляризации (аппроксимация)
alpha_landweber = 0.1 # параметр для регуляризации (Ландвебер)
epsilon = 1e-6 # Точность для алгоритма Ландвебера
max_tau = 1e-2 # Максимальное значение шага
```

Рисунок 4 – Задание параметров

В данном программном коде n - количество базисных функций для аппроксимации. Alpha_approx и alpha_landweber – параметры регуляризации для различных методов. Epsilon – точность, при которой остановится алгоритм Ландвебера. Max_tau – максимальный шаг итерации.

```
# Определяем функции g_i(x) и h_i(y)
def g_i(x, i):
    return np.sin(i * np.pi * x)

def h_i(y, i):
    return np.sin(i * np.pi * y)

# Определяем ядро интегрального уравнения K(x, y)
def K(x, y):
    return sum(g_i(x, i) * h_i(y, i) for i in range(1, N + 1))

# функция phi_T(x, T)
def phi_T(x, T):
    return 4 * x * (1 - x) + np.sin(2 * np.pi * x**2)**2 * np.exp(-T)
```

Рисунок 5 – Определение функций

В данном программном коде $g_i(x, i)$ и $h_i(y, i)$ - базисные функции. $K(x, y)$ - ядро интегрального уравнения, вычисляемое как сумма произведений базисных функций. $\Phi_T(x, T)$ - целевая функция, зависящая от времени T .

```
# Дискретизируем интервал [0, 1]
x_vals = np.linspace(0, 1, 100)
y_vals = np.linspace(0, 1, 100)
dx = x_vals[1] - x_vals[0]
```

Рисунок 6 – Дискретизация интервала

В данном программном коде делается дискретизация интервала $[0, 1]$ на 100 точек. Dx - шаг дискретизации.

```
def solve_approximation(T):
    # Создаем матрицу K_ij и вектор f_i для заданного T
    K_matrix = np.zeros((N, N))
    f_vector = np.zeros(N)

    for i in range(1, N + 1):
        for j in range(1, N + 1):
            K_matrix[i-1, j-1] = np.sum(h_i(y_vals, i) * h_i(y_vals, j)) * dx
        f_vector[i-1] = np.sum(h_i(y_vals, i) * phi_T(y_vals, T)) * dx

    # Решение с помощью аппроксимации с регуляризацией
    K_matrix_reg = K_matrix + alpha_approx * np.eye(N)
    c_vector_approx = np.linalg.solve(K_matrix_reg, f_vector)

    # Восстанавливаем функцию phi(x) для заданного T с помощью аппроксимации
    phi_approx = np.zeros_like(x_vals)
    for i in range(1, N + 1):
        phi_approx += c_vector_approx[i-1] * g_i(x_vals, i)

    return phi_approx
```

Рисунок 7 – Решение задачи с использованием аппроксимации

В данном программном коде происходит создание матрицы K_matrix и вектора f_vector . Решение системы уравнений с регуляризацией. Восстановление функции $\phi(x)$ с помощью аппроксимации.

```
def landweber(K_matrix, f_vector, epsilon, max_iterations=10000, max_tau=1e-2):
    x = np.zeros_like(f_vector) # Начальное приближение
    r = f_vector - K_matrix @ x # Начальная невязка
    residuals = [np.linalg.norm(r)] # Список для хранения значений невязки
    iteration = 0

    while np.linalg.norm(r) > epsilon and iteration < max_iterations: # Условие остановки
        tau = (r @ r) / (r @ (K_matrix @ r)) # Шаг итерации
        tau = min(tau, max_tau) # Ограничение на максимальный шаг
        x += tau * r # Обновление решения
        r = f_vector - K_matrix @ x # Обновление невязки
        residuals.append(np.linalg.norm(r)) # Запись значения невязки
        iteration += 1

    return x, iteration, residuals
```

Рисунок 8 – Реализация алгоритма Ландвебера

В данном программном коде происходит инициализация начального приближения x и невязки r . Итерационное обновление решения и невязки. Список $residuals$ для хранения значений невязки.

```
def solve_landweber(T):
    # Создаем матрицу K_ij и вектор f_i для заданного T
    K_matrix = np.zeros((N, N))
    f_vector = np.zeros(N)

    for i in range(1, N + 1):
        for j in range(1, N + 1):
            K_matrix[i-1, j-1] = np.sum(h_i(y_vals, i) * h_i(y_vals, j)) * dx
            f_vector[i-1] = np.sum(h_i(y_vals, i) * phi_T(y_vals, T)) * dx

    # Решение с помощью аппроксимации с регуляризацией
    K_matrix_reg = K_matrix + alpha_landweber * np.eye(N)

    x_landweber, iterations, residuals = landweber(K_matrix_reg, f_vector, epsilon, max_tau=max_tau)

    # Восстанавливаем функцию phi(x) для заданного T с помощью алгоритма Ландвебера
    phi_landweber = np.zeros_like(x_vals)
    for i in range(1, N + 1):
        phi_landweber += x_landweber[i-1] * g_i(x_vals, i)

    return phi_landweber, residuals
```

Рисунок 9 – Решение задачи с использованием алгоритма Ландвебера

В данном программном коде происходит создание матрицы K_matrix и вектора f_vector . Решение с использованием алгоритма Ландвебера. Восстановление функции $\phi(x)$.

```
# Пробуем разные значения T и смотрим на результаты
T_values = [0.1, 1, 2, 5]
for T in T_values:
    phi_approx = solve_approximation(T)
    phi_landweber, residuals = solve_landweber(T)

    plt.figure(figsize=(12, 6))
    plt.plot(x_vals, phi_T(x_vals, T-T), 'k-', label=f'Исходная функция при T={T-T}') # черная линия
    plt.plot(x_vals, phi_approx, 'b--', label=f'Аппроксимация phi(x) при T={T}') # синяя пунктирная линия
    plt.plot(x_vals, phi_landweber, 'r:', label=f'Алгоритм Ландвебера phi(x) при T={T}') # красная точечная линия
    plt.legend()
    plt.xlabel('Длина стержня (x)')
    plt.ylabel('Температура (phi)')
    plt.title(f'функции при T={T}')
    plt.show()

    plt.figure(figsize=(12, 6))
    plt.plot(residuals, 'r-', label='Невязка Ландвебера') # красная линия
    plt.yscale('log')
    plt.xlabel('Итерации')
    plt.ylabel('Невязка')
    plt.title(f'Сходимость алгоритма Ландвебера при T={T}')
    plt.legend()
    plt.show()
```

Рисунок 10 – Построение графиков

В данном программном коде для каждого значения T строятся графики. Сравниваются исходная функция, аппроксимация и решение алгоритма Ландвебера. Построение графика сходимости алгоритма Ландвебера.

В данной работе реализованы методы аппроксимации и алгоритма Ландвебера для решения интегрального уравнения с вырожденным ядром. Аппроксимация выполняется с использованием регуляризации, что позволяет стабилизировать решение. Алгоритм Ландвебера применяется для итерационного решения задачи, обеспечивая постепенное улучшение результата. Различные значения параметра T демонстрируют, как изменяются результаты аппроксимации и поведения алгоритма. Построенные графики показывают хорошее совпадение аппроксимированных функций с исходной функцией и наглядно отображают процесс сходимости алгоритма Ландвебера.

3.2 Сравнение результатов, полученных различными методами

В данном разделе будет проведено сравнение результатов, полученных численными методами аппроксимации ядра и метода Ландвебера.

Цель сравнения заключается в выявлении наиболее эффективного метода для решения обратных задач теплопроводности в различных условиях. Для этого будут проанализированы точность, стабильность, скорость вычислений и использование памяти для каждого метода.

Исходная функция продемонстрирована на рисунке 11.

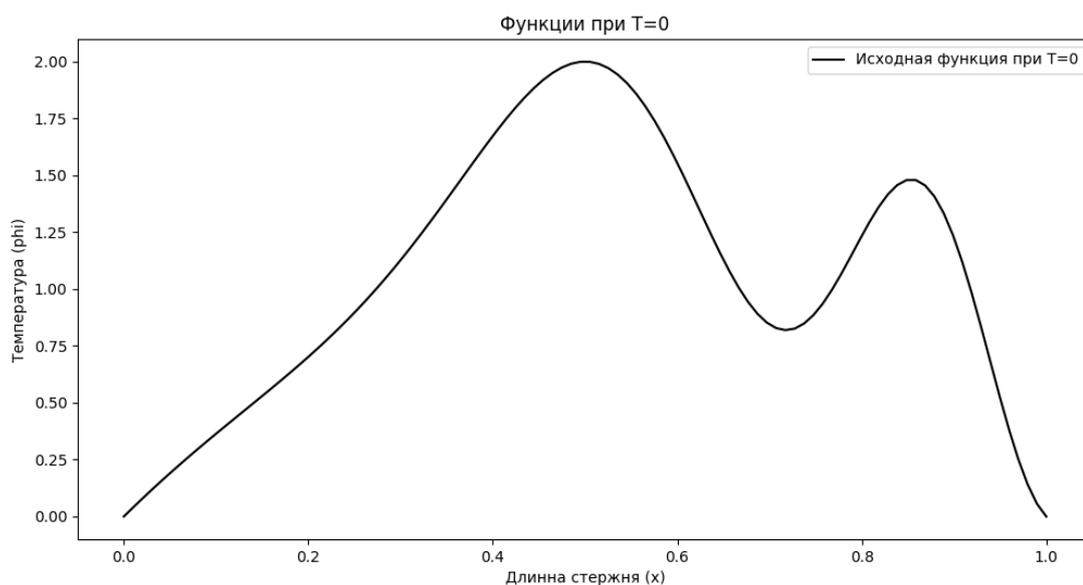


Рисунок 11 – Исходная функция в момент $T=0$

На рисунке представлена исходная функция, используемая в качестве начальных условий для численного моделирования обратных задач теплопроводности. Функция изображает распределение температуры по стержню в начальный момент времени $T=0$, что служит основой для дальнейших вычислений и анализа.

График функций в момент $T=0.1$ показан на рисунке 12.

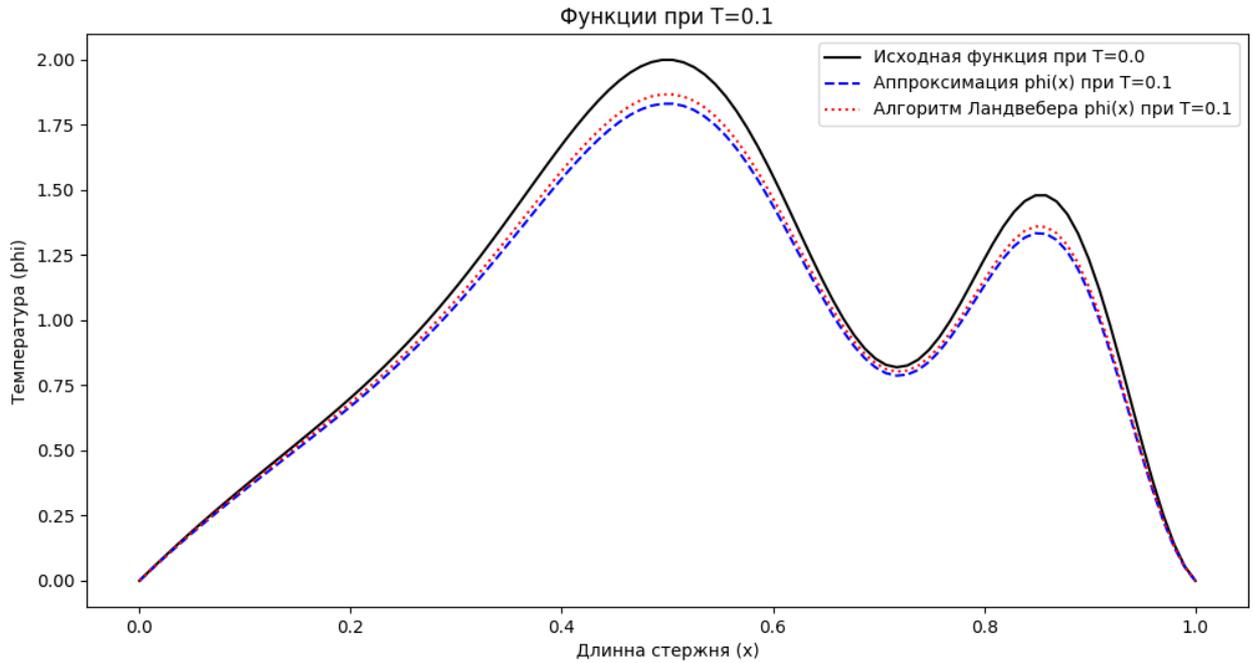


Рисунок 12 – График функций в момент T=0.1

Этот рисунок демонстрирует изменение температуры по стержню через короткий промежуток времени после начального момента, $T=0.1$. График включает как исходную функцию, так и результаты аппроксимации и решения алгоритма Ландвебера, что позволяет визуально сравнить точность и поведение различных методов.

Невязка алгоритма Ландвебера в момент $T=0.1$ представлена на рисунке 13.

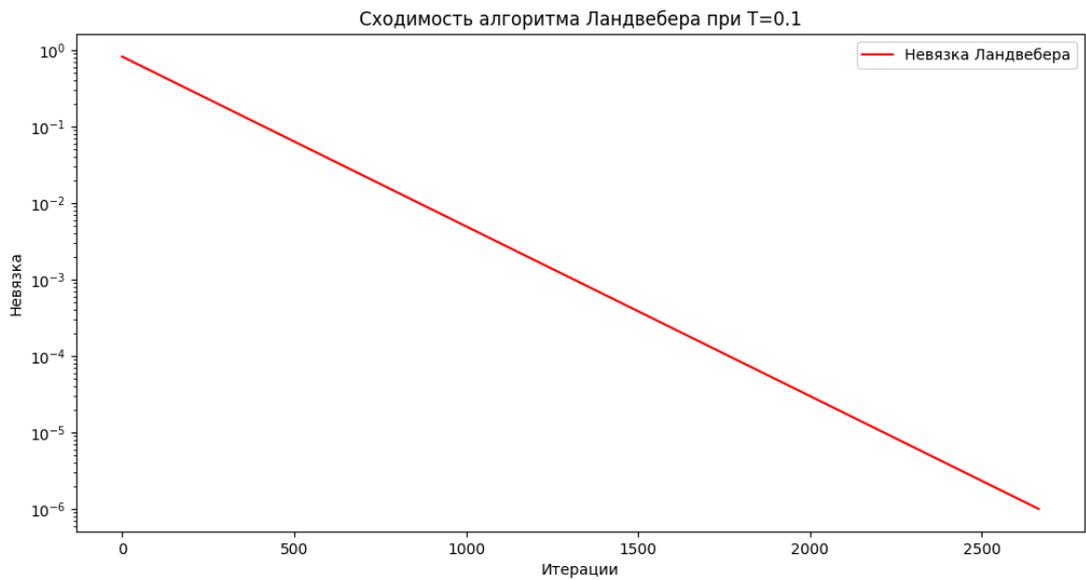


Рисунок 13 – Невязка алгоритма Ландвебера в момент $T=0.1$

На данном рисунке показана невязка алгоритма Ландвебера при $T=0.1$. Невязка иллюстрирует разницу между текущим приближением и точным решением, что помогает понять, насколько близко алгоритм подходит к истинному значению.

График функций в момент $T=1$ показан на рисунке 14.

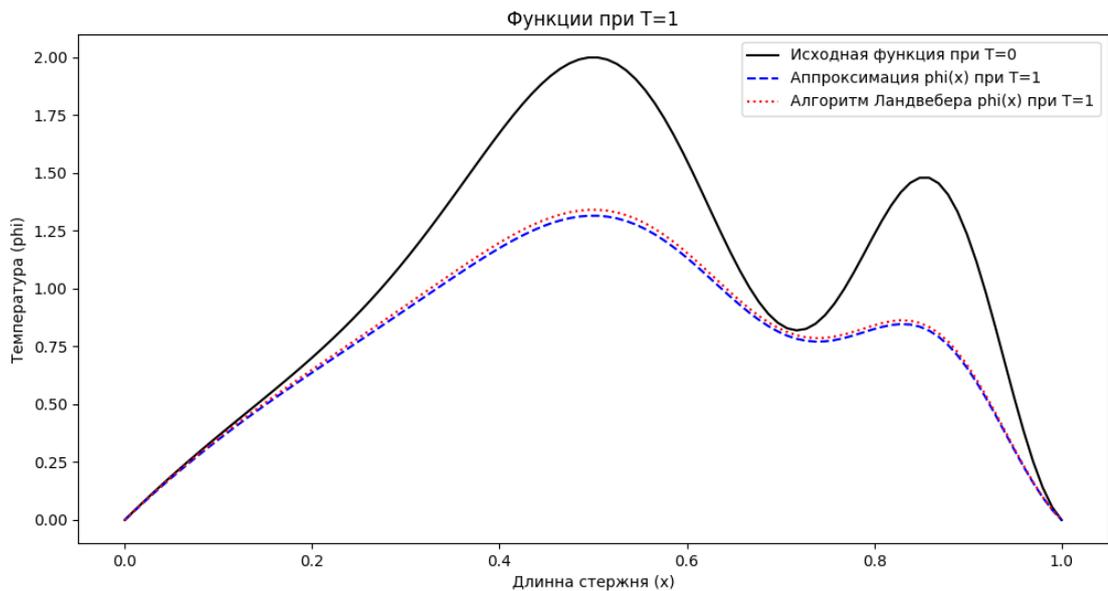


Рисунок 14 – График функций в момент $T=1$

Этот график отображает результаты моделирования через более продолжительный промежуток времени, $T=1$. Включены исходная функция, аппроксимация и решение алгоритма Ландвебера, что позволяет оценить долгосрочную точность и стабильность методов.

График функций в момент $T=2$ представлен на рисунке 15.

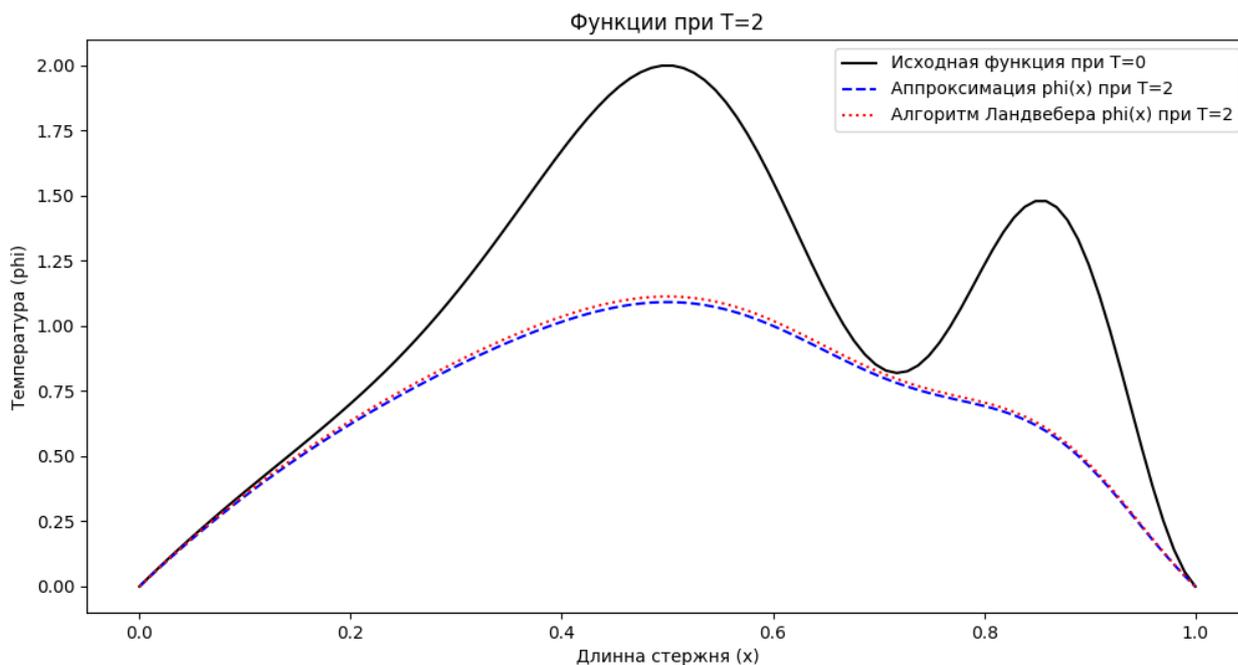


Рисунок 15 – График функций в момент $T=2$

На этом рисунке представлены результаты через еще более длительное время, $T=2$. График позволяет проанализировать, как методы справляются с моделированием на протяжении более длительного временного интервала и как они сохраняют точность по сравнению с исходной функцией.

График функций в момент $T=5$ продемонстрирован на рисунке 16.

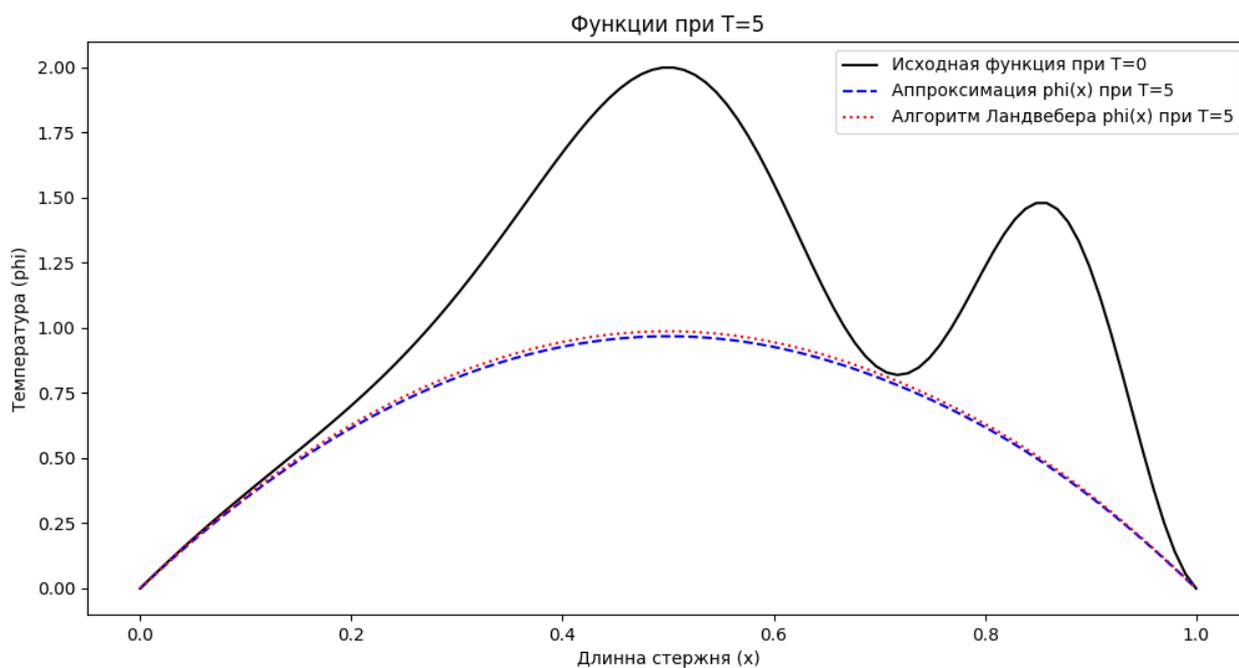


Рисунок 16– График функций в момент $T=5$

Последний график демонстрирует результаты при $T=5$. Здесь видны долгосрочные эффекты и точность методов аппроксимации и алгоритма Ландвебера. Это наиболее удаленный по времени момент, что дает возможность оценить поведение алгоритмов на большом временном интервале.

На рисунке 17 представлен график восстановленной функции алгоритмом Ландвебера при $T=4$.

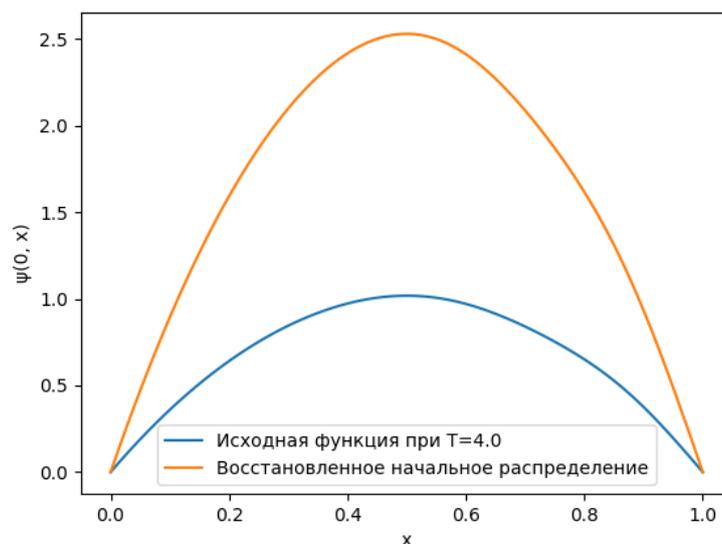


Рисунок 17 – Восстановленное начальное распределение при $T=4$

Далее представлены сравнительные таблицы 1 и 2 для методов аппроксимации ядра и алгоритма Ландвебера, включающая время и использование памяти для каждого метода при различных значениях параметров T , N и ϵ .

Таблица 1 – Значения метода аппроксимации ядра

T	N	ϵ	Время (с)	Память (МБайт)
0,1	50	1e-5	1,25	117,65
0,1	100	1e-6	1,29	132,71
1,0	50	1e-5	1,16	107,68
1,0	100	1e-6	1,65	147,67
2	50	1e-5	1,38	141,21
2	100	1e-6	1,58	150,43
5	50	1e-5	1,66	141,38
5	100	1e-6	1,78	167,19

Таблица 2 – Значения метода алгоритма Ландвебера

T	N	ϵ	Время (с)	Память (МБайт)
0,1	50	1e-5	0,8	87,28
0,1	100	1e-6	1,03	100,55
1,0	50	1e-5	0,71	74,25
1,0	100	1e-6	1,06	121,88
2	50	1e-5	1,25	107
2	100	1e-6	1,28	128,54
5	50	1e-5	1,21	116,99
5	100	1e-6	1,26	149,26

Сравнительный анализ времени выполнения методов аппроксимации ядра и алгоритма Ландвебера показывает, что алгоритм Ландвебера в целом работает быстрее. Время выполнения метода аппроксимации ядра варьируется от 1.16 до 1.78 секунд в зависимости от параметров T, N и ϵ , тогда как время выполнения алгоритма Ландвебера колеблется от 0.71 до 1.28 секунд. Особенно заметно это различие при больших значениях параметров, где метод аппроксимации ядра требует больше времени.

Анализ использования памяти демонстрирует, что метод аппроксимации ядра также требует больше памяти. Память, используемая методом аппроксимации ядра, варьируется от 107.68 до 167.19 МБ, тогда как алгоритм Ландвебера использует от 74.25 до 149.26 МБ. Разница в потреблении памяти особенно очевидна при больших значениях N, где метод аппроксимации ядра использует на 10-20 МБ больше памяти.

В заключение, алгоритм Ландвебера показывает лучшие результаты по времени выполнения и использованию памяти в сравнении с методом аппроксимации ядра. Это делает его более эффективным выбором для задач, где важна оперативность и экономия ресурсов. Метод аппроксимации ядра, хотя и менее эффективен по этим показателям, может быть полезен в ситуациях, где требуются специфические его свойства или точность.

Выводы по главе 3

Третья глава посвящена программной реализации численных алгоритмов и сравнительному анализу их эффективности. В главе описываются детали реализации методов аппроксимации ядра и алгоритма Ландвебера с использованием библиотек Python, таких как NumPy и Matplotlib. Приведены блок-схемы и листинг программного кода, что позволяет детально рассмотреть каждый этап численных расчетов. Основное внимание уделено сравнению эффективности различных методов по времени выполнения, использованию памяти и точности результатов. Анализ показывает, что алгоритм Ландвебера демонстрирует более высокую производительность и меньшие требования к памяти по сравнению с методом аппроксимации ядра. Однако метод аппроксимации может быть предпочтительным в ситуациях, требующих высокой точности. Глава включает результаты тестирования алгоритмов на модельных задачах, что подтверждает их практическую применимость. Графики и таблицы, представленные в работе, визуально демонстрируют поведение различных методов в динамике и их сходимость. Заключение главы подчеркивает важность выбора подходящего численного метода в зависимости от специфических условий и требований задачи, что является ключевым аспектом для достижения оптимальных результатов в решении обратных задач теплопроводности.

Заключение

В данной выпускной квалификационной работе были рассмотрены численные алгоритмы для решения обратных задач теплопроводности, а также проведен их сравнительный анализ. Основным вопросом исследования стало сравнение двух методов: метода аппроксимации ядра и алгоритма Ландвебера. Эти методы имеют широкое применение в различных областях науки и техники, таких как геофизика, медицина, инженерия и материаловедение.

Обратные задачи теплопроводности представляют собой важную область исследований, так как они позволяют восстановить неизвестные параметры или начальные условия на основе данных измерений температуры. В работе были рассмотрены теоретические основы этих задач, включая уравнение теплопроводности и различные граничные условия. Основное внимание уделено методу сведения задачи к интегральному уравнению Фредгольма I рода, что позволяет формулировать задачу в виде, удобном для применения численных методов.

Во второй главе работы подробно рассмотрены проблемы некорректности обратных задач и методы их решения. Были исследованы методы регуляризации, включая регуляризацию Тихонова, метод усеченных сингулярных разложений и итерационные методы, такие как алгоритм Ландвебера. Эти методы позволяют стабилизировать решение и обеспечить надежные результаты при решении некорректных задач.

Третья глава работы была посвящена программной реализации численных методов и их сравнительному анализу. Реализованные алгоритмы были протестированы на модельных задачах, что позволило провести детальный анализ их эффективности. Сравнительные таблицы и графики, представленные в работе, показали, что алгоритм Ландвебера имеет преимущество по времени выполнения и использованию памяти. Это делает его более подходящим для задач, где важна оперативность и экономия ресурсов.

Таким образом, в результате проведенных исследований было установлено, что выбор численного метода для решения обратных задач теплопроводности должен зависеть от конкретных условий и требований задачи. Метод аппроксимации ядра может быть полезен в ситуациях, где требуется высокая точность, несмотря на большие затраты ресурсов. Алгоритм Ландвебера, в свою очередь, демонстрирует более высокую производительность и стабильность в использовании памяти, что делает его предпочтительным выбором в большинстве практических приложений.

Практическая значимость полученных результатов заключается в возможности их применения для решения реальных задач теплопроводности, что открывает перспективы для дальнейших исследований и разработок в этой области. Представленные методы и алгоритмы могут быть использованы для разработки программного обеспечения, предназначенного для анализа и решения обратных задач теплопроводности в различных научных и инженерных приложениях.

Работа подчеркивает важность выбора подходящего численного метода в зависимости от конкретных условий задачи, что является ключевым аспектом для достижения наилучших результатов в решении обратных задач теплопроводности.

Список используемой литературы

1. Адамар Ж. Некорректные задачи математической физики. М. : Наука, 1978. 372 с.
2. Аналитическое описание процесса нестационарной теплопроводности / Б.А. Вороненко, А.Г. Крысин, В.В. Пеленко, О.А. Цуранов : Учеб.-метод. пособие. – СПб. : НИУ ИТМО; ИХиБТ, 2014. 48 с.
3. Вентцель Е.С. Теоретические основы теплотехники. Теплообмен : учебное пособие / Е.С. Вентцель; Ульяновский гос. техн. ун-т. – Ульяновск : УлГТУ, 2013. 204 с.
4. Гоглачев А.В., Лойко А.Э. Теплофизика. Практические занятия и лабораторный практикум: Учебное электронное текстовое издание. – Екатеринбург : Уральский федеральный университет, 2015. 244 с.
5. Гласко В. Б. Обратные задачи математической физики. – М. : Изд-во МГУ, 1984.
6. Егоров В.И. Аналитические методы решения задач теплопроводности. Учебное пособие. – Университет ИТМО, 2015. 38 с.
7. Исаев А.А. Введение в теорию некорректных задач. М. : МГУ, 2009. 320 с.
8. Кабанихин С. И. Обратные и некорректные задачи. – Новосибирск : Сибирское научное издательство, 2009.
9. Канторович Л.В., Крылов В.И. Приближенные методы высшего анализа. М. : Физматгиз, 1958. 840 с.
10. Лаврентьев М. М. Условно-корректные задачи для дифференциальных уравнений. — Новосибирск : ГУ, 1973.
11. Морозов В. А. Регулярные методы решения некорректно поставленных задач. — М. : Наука, 1987.
12. Талалов, С. В. Обратные и некорректные задачи : электронное учебное пособие / С. В. Талалов. – Тольятти : ТГУ, 2019. 60 с.

13. Тихонов, А.Н. Методы решения некорректных задач / А.Н. Тихонов, В.Я. Арсенин. – М. : Наука, 1979. 285 с.
14. Прокофьев В.Г. Методические указания к решению краевых задач для уравнения теплопроводности методом функций Грина: учебно-методическое пособие. Томск: Томский государственный университет, 2017. 19 с.
15. Пилипенко Н.В. Методы и приборы нестационарной теплотметрии на основе решения обратных задач теплопроводности. – СПб : СПбГУ ИТМО, 2011. 180 с.
16. Рид, М. Методы современной математической физики. Т. I. Функциональный анализ / М. Рид, Б. Саймон. – М. : Мир, 1977. 354 с.
17. Стренг, Г. Линейная алгебра и ее применения / Г. Стренг – М. : Мир, 1980. 454 с.
18. Сумин М.И. Метод регуляризации А.Н. Тихонова для решения операторных уравнений первого рода: Учебно-методическое пособие. – Нижний Новгород : Нижегородский госуниверситет, 2016. 56 с.
19. Ткаченко Д.С. Семинар 1. Основные уравнения и постановка задач математической физики. – Москва : НИЯУ МИФИ, 2015. 12 с.
20. Усков В.И., Небольсина В.И. Решение одного интегрального уравнения Фредгольма первого рода / В.И. Усков, В.И. Небольсина. // Молодой ученый. 2019. № 40. С. 80-86.
21. Hansen P.C. Discrete Inverse Problems: Insight and Algorithms. Philadelphia: SIAM, 2010. 272 p.
22. Kress R. Linear Integral Equations. New York: Springer, 1999. 412 p.
23. Luenberger D.G. Optimization by Vector Space Methods. New York : Wiley, 1969. 326 p.
24. Mazumder S. Numerical Methods for Partial Differential Equations. Ohio, 2015. 484 p.
25. Vogel C.R. Computational Methods for Inverse Problems. Philadelphia : SIAM, 2002. 200 p.

Приложение А

Листинг программного кода

```
import numpy as np
import matplotlib.pyplot as plt

# Задаем количество членов N для аппроксимации с вырожденным ядром
N = 100
alpha_approx = 0.02 # параметр для регуляризации (аппроксимация)
alpha_landweber = 0.01 # параметр для регуляризации (Ландвебер)
epsilon = 1e-6 # Точность для алгоритма Ландвебера
max_tau = 1e-2 # Максимальное значение шага

# Определяем функции g_i(x) и h_i(y)
def g_i(x, i):
    return np.sin(i * np.pi * x)

def h_i(y, i):
    return np.sin(i * np.pi * y)

# Определяем ядро интегрального уравнения K(x, y)
def K(x, y):
    return sum(g_i(x, i) * h_i(y, i) for i in range(1, N + 1))

# Функция phi_T(x, T)
def phi_T(x, T):
    return 4 * x * (1 - x) + np.sin(2 * np.pi * x**2)**2 * np.exp(-T)

# Дискретизируем интервал [0, 1]
x_vals = np.linspace(0, 1, 100)
y_vals = np.linspace(0, 1, 100)
dx = x_vals[1] - x_vals[0]

def solve_approximation(T):
```

Продолжение приложения А

```
# Создаем матрицу K_ij и вектор f_i для заданного T

K_matrix = np.zeros((N, N))
f_vector = np.zeros(N)

for i in range(1, N + 1):
    for j in range(1, N + 1):
        K_matrix[i-1, j-1] = np.sum(h_i(y_vals, i) * h_i(y_vals, j)) * dx
    f_vector[i-1] = np.sum(h_i(y_vals, i) * phi_T(y_vals, T)) * dx

# Решение с помощью аппроксимации с регуляризацией
K_matrix_reg = K_matrix + alpha_approx * np.eye(N)
c_vector_approx = np.linalg.solve(K_matrix_reg, f_vector)

# Восстанавливаем функцию phi(x) для заданного T с помощью аппроксимации
phi_approx = np.zeros_like(x_vals)
for i in range(1, N + 1):
    phi_approx += c_vector_approx[i-1] * g_i(x_vals, i)

return phi_approx

def landweber(K_matrix, f_vector, epsilon, max_iterations=10000, max_tau=1e-2):
    x = np.zeros_like(f_vector) # Начальное приближение
    r = f_vector - K_matrix @ x # Начальная невязка
    residuals = [np.linalg.norm(r)] # Список для хранения значений невязки
    iteration = 0

    while np.linalg.norm(r) > epsilon and iteration < max_iterations: # Условие остановки
        tau = (r @ r) / (r @ (K_matrix @ r)) # Шаг итерации
        tau = min(tau, max_tau) # Ограничение на максимальный шаг
        x += tau * r # Обновление решения
        r = f_vector - K_matrix @ x # Обновление невязки
```

Продолжение приложения А

```
residuals.append(np.linalg.norm(r)) # Запись значения невязки
iteration += 1

return x, iteration, residuals

def solve_landweber(T):
    # Создаем матрицу  $K_{ij}$  и вектор  $f_i$  для заданного  $T$ 
    K_matrix = np.zeros((N, N))
    f_vector = np.zeros(N)

    for i in range(1, N + 1):
        for j in range(1, N + 1):
            K_matrix[i-1, j-1] = np.sum(h_i(y_vals, i) * h_i(y_vals, j)) * dx
            f_vector[i-1] = np.sum(h_i(y_vals, i) * phi_T(y_vals, T)) * dx

    # Решение с помощью аппроксимации с регуляризацией
    K_matrix_reg = K_matrix + alpha_landweber * np.eye(N)

    x_landweber, iterations, residuals = landweber(K_matrix_reg, f_vector, epsilon,
max_tau=max_tau)

    # Восстанавливаем функцию  $\phi(x)$  для заданного  $T$  с помощью алгоритма
Ландвебера
    phi_landweber = np.zeros_like(x_vals)
    for i in range(1, N + 1):
        phi_landweber += x_landweber[i-1] * g_i(x_vals, i)

    return phi_landweber, residuals

# Пробуем разные значения  $T$  и смотрим на результаты
T_values = [ 0.1, 1, 2, 5]
for T in T_values:
```

Продолжение приложения А

```
phi_approx = solve_approximation(T)
phi_landweber, residuals = solve_landweber(T)

plt.figure(figsize=(12, 6))
plt.plot(x_vals, phi_T(x_vals, T-T), 'k-', label=f'Исходная функция при T={T-T}') #
черная линия
plt.plot(x_vals, phi_approx, 'b--', label=f'Аппроксимация phi(x) при T={T}') # синяя
пунктирная линия
plt.plot(x_vals, phi_landweber, 'r.', label=f'Алгоритм Ландвебера phi(x) при T={T}')
# красная точечная линия
plt.legend()
plt.xlabel('Длина стержня (x)')
plt.ylabel('Температура (phi)')
plt.title(f'Функции при T={T}')
plt.show()

plt.figure(figsize=(12, 6))
plt.plot(residuals, 'r-', label='Невязка Ландвебера') # красная линия
plt.yscale('log')
plt.xlabel('Итерации')
plt.ylabel('Невязка')
plt.title(f'Сходимость алгоритма Ландвебера при T={T}')
plt.legend()
plt.show()
```