



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт ИВТИ
Кафедра МКМ

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

Направление 01.03.02 Прикладная математика и информатика
(код и наименование)

Образовательная программа Математическое моделирование

Форма обучения очная
(очная/очно-заочная/заочная)

Тема: Численное решение задачи радиационно-кондуктивного теплообмена в полупрозрачной для излучения среде

Студент А-16-20  Симаков А.М.
группа подпись фамилия и инициалы

Руководитель ВКР д.ф.-м.н. профессор  Амосов А.А.
уч. степень должность подпись фамилия и инициалы

Консультант уч. степень должность подпись фамилия и инициалы

Внешний консультант уч. степень должность подпись фамилия и инициалы

организация

«Работа допущена к защите»

Заведующий кафедрой к.ф.-м.н. доцент  Зубков П.В.
уч. степень звание подпись фамилия и инициалы

Дата

Москва, 2024



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт

ИВТИ

Кафедра

МКМ

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
(БАКАЛАВРСКУЮ РАБОТУ)**

Направление 01.03.02 Прикладная математика и информатика
(код и наименование)

Образовательная программа Математическое моделирование

Форма обучения очная
(очная/очно-заочная/заочная)


Тема: Численное решение задачи радиационно-кондуктивного теплообмена в полупрозрачной для излучения среде

Студент А-16-20 группа  Симаков А.М.
подпись фамилия и инициалы

Руководитель ВКР д.ф.-м.н. профессор  Амосов А.А.
уч. степень должность подпись фамилия и инициалы

Консультант уч. степень должность подпись фамилия и инициалы

Внешний консультант уч. степень должность подпись фамилия и инициалы

организация
Заведующий кафедрой к.ф.-м.н. доцент  Зубков П.В.
уч. степень звание подпись фамилия и инициалы

Место выполнения работы ФГБОУ ВО «НИУ «МЭИ»

СОДЕРЖАНИЕ РАЗДЕЛОВ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ

1. Изучение постановки физической задачи.
2. Изучение литературы по теме выполнения работы.
3. Построение разностной аппроксимации исходной задачи.
4. Разработка алгоритма вычисления решения.
5. Разработка и отладка программы.
6. Проведение вычислительных экспериментов.

ПЕРЕЧЕНЬ ГРАФИЧЕСКОГО МАТЕРИАЛА

Количество листов

Количество слайдов в презентации

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Амосов А. А. Вычислительные методы для инженеров: учебное пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. - СПб.: Лань, 2014. - 672 с.
2. Зигель Р., Хауэлл ДЖ. Теплообмен излучением. - Москва.: Мир, 1975. - 935 с.

Примечания:

1. Задание брошюруется вместе с выпускной работой после титульного листа (страницы задания имеют номера 2, 3).
2. Отзыв руководителя, рецензия(и), отчет о проверке на объем заимствований и согласие студента на размещение работы в открытом доступе вкладываются в конверт (файловую папку) под обложкой работы.

Аннотация

В данной работе рассматривается задача радиационно-кондуктивного теплообмена в полупрозрачной для излучения среде. Разработан и программно реализован численный метод решения. Проведены вычислительные эксперименты, позволяющие исследовать зависимость распределения температуры и интенсивности излучения от физических свойств слоя вещества (коэффициенты теплопроводности, светового поглощения и рассеяния), толщины слоя, а также от интенсивности излучения, падающего на его поверхность.

Abstract

This paper considers the problem of radiation-conduction heat transfer in a medium translucent to radiation. A numerical solution method has been developed and implemented in software. Computational experiments have been carried out to study the dependence of the distribution of temperature and radiation intensity on the physical properties of the layer of matter (coefficients of thermal conductivity, light absorption and scattering), the thickness of the layer, as well as on the intensity of radiation incident on its surface.

Введение

Изучение процесса теплообмена в полупрозрачных средах, перенос тепла в которых происходит с помощью теплового излучения, поглощения и рассеяния представляет собой актуальную инженерную и научную задачу [1, 2]. Численное решение подобных задач почти всегда является единственным возможным вариантом. В связи с этим вопрос построения эффективных численных методов их решения остается актуальным.

В данной работе рассматривается задача радиационно-кондуктивного теплообмена в области, представляющей собой плоский слой теплопроводящего и полупрозрачного для излучения вещества.

Целью данной работы является математическое моделирование процесса радиационно-кондуктивного теплообмена в полупрозрачной для излучения среде. Исходя из этого были поставлены следующие задачи: построение и реализация численного метода решения поставленной задачи, проведение вычислительных экспериментов с различными значениями параметров.

В ходе выполнения работы было сделано следующее:

1. Построена разностная аппроксимация поставленной задачи;
2. Разработан и реализован алгоритм поиска численного решения поставленной задачи;
3. Проведена серия вычислительных экспериментов, показывающая работоспособность алгоритма и демонстрирующая зависимость распределения температуры и интенсивности излучения от различных параметров;

Настоящая выпускная квалификационная работа организована следующим образом. В главе 1 рассматриваются физическая и математическая постановки

задачи. В главе 2 производится разностная аппроксимация рассматриваемой задачи. В главе 3 приведены результаты вычислительных экспериментов и пояснения к ним.

1. Постановка задачи

1.1. Физическая постановка

Рассматривается задача радиационно-кондуктивного теплообмена в теплопроводящем, полупрозрачном для излучения плоском слое толщиной L м (Рис. 1.1). На верхнюю поверхность слоя падает излучение с заданной интенсивностью. Теплообмен в слое происходит посредством теплопроводности, переноса излучения, его рассеяния и поглощения. Предположим, что верхняя и нижняя границы слоя теплоизолированы. Значения коэффициентов теплопроводности, рассеяния и поглощения заданы. Искомыми величинами являются распределение абсолютной температуры и интегральная интенсивность излучения в рассматриваемом слое.

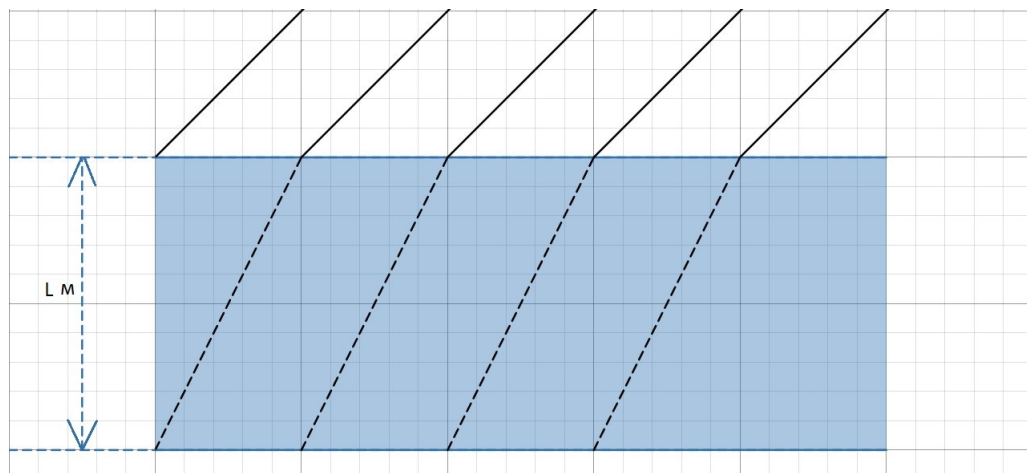


Рис. 1.1. Слой вещества

1.2. Математическая постановка

Рассматриваемая задача описывается системой, состоящей из стационарного уравнения теплопроводности

$$-\frac{d}{dx} \left(\lambda \frac{du}{dx} \right) + \kappa \sigma_0 |u|^3 u = \frac{\kappa}{2} \int_{-1}^1 I(x, \mu) d\mu \quad (1.1)$$

с краевыми условиями

$$-\lambda \frac{du}{dx} \Big|_{x=0} = 0 \quad (1.2)$$

$$\lambda \frac{du}{dx} \Big|_{x=L} = 0 \quad (1.3)$$

и интегро-дифференциального уравнения переноса излучения

$$-\mu \frac{\partial I}{\partial x} + (\kappa + s)I = \frac{s}{2} \int_{-1}^1 I(x, \mu) d\mu + \kappa \sigma_0 |u|^3 u \quad (1.4)$$

с краевыми условиями

$$I(x, \mu)|_{x=0} = I_0, \quad -1 \leq \mu < 0 \quad (1.5)$$

$$I(x, \mu)|_{x=L} = 0, \quad 0 < \mu \leq 1 \quad (1.6)$$

Здесь $u(x)$ - абсолютная температура среды, $I(x, \mu)$ - интенсивность излучения; μ - косинус угла распространения излучения, x - координата точки в слое, L - толщина слоя, λ - коэффициент теплопроводности, κ - коэффициент поглощения, s - коэффициент рассеяния, σ_0 - постоянная Стефана-Больцмана.

Следует отметить, что в оригинальном законе Стефана-Больцмана модуль не присутствует. Он добавлен потому, что без модуля математическая задача допускает существование решения с отрицательными значениями абсолютных температур, что противоречит физическому смыслу.

2. Построение разностной аппроксимации

2.1. Сведение интегро-дифференциального уравнения к интегральному уравнению

Сведём проблему решения интегро-дифференциального уравнения (1.4) – (1.6) к решению интегрального уравнения. Напомним, что функция $I(x, \mu)$ является решением задачи

$$\begin{cases} -\mu \frac{\partial I}{\partial x} + (\varkappa + s)I = \frac{s}{2} \int_{-1}^1 I(x, \mu) d\mu + \varkappa \sigma_0 u^4 \\ I(x, \mu)|_{x=0} = I_0, \quad -1 \leq \mu < 0 \\ I(x, \mu)|_{x=L} = 0, \quad 0 < \mu \leq 1 \end{cases}$$

Введём $\alpha = \varkappa + s$ - коэффициент экстинкции вещества. Также сделаем замену

$$\Xi(x) = \frac{s}{2} \int_{-1}^1 I(x, \mu) d\mu + \varkappa \sigma_0 u^4 \quad (2.1)$$

Тогда уравнение (1.4) примет вид

$$-\mu \frac{\partial I}{\partial x} + \alpha I = \Xi(x)$$

С учётом (1.5), (1.6) его решением будет

$$I(x, \mu) = \begin{cases} \frac{1}{|\mu|} \int_0^x \exp\left(-\frac{\alpha}{|\mu|}(x - x')\right) \Xi(x') dx' + I_0 \exp\left(-\frac{\alpha}{|\mu|}x\right), & \mu \in [-1, 0) \\ \frac{1}{\mu} \int_x^L \exp\left(-\frac{\alpha}{\mu}(x' - x)\right) \Xi(x') dx', & \mu \in (0, 1] \end{cases}$$

Интегрируя $I(x, \mu)$ по μ на отрезке $[-1, 1]$, получим

$$\begin{aligned} \int_{-1}^1 I(x, \mu) d\mu = I_0 \int_{-1}^0 \exp\left(-\frac{\alpha}{|\mu|}x\right) d\mu + \int_0^x \int_{-1}^0 \frac{1}{|\mu|} \exp\left(-\frac{\alpha}{|\mu|}(x-x')\right) d\mu \Xi(x') dx' + \\ + \int_x^L \int_0^1 \frac{1}{\mu} \exp\left(-\frac{\alpha}{\mu}(x'-x)\right) d\mu \Xi(x') dx' \quad (2.2) \end{aligned}$$

Введём интегро-экспоненциальную функцию порядка n ([3], Гл. 5 §1)

$$E_n(x) = \int_0^1 \mu^{n-2} \exp\left(-\frac{x}{\mu}\right) d\mu$$

В частности, при $n = 1$ и $n = 2$ получим

$$E_1(x) = \int_0^1 \frac{1}{\mu} \exp\left(-\frac{x}{\mu}\right) d\mu \quad (2.3)$$

$$E_2(x) = \int_0^1 \exp\left(-\frac{x}{\mu}\right) d\mu \quad (2.4)$$

Положим

$$\mathcal{S}(x) = \frac{1}{2} \int_{-1}^1 I(x, \mu) d\mu$$

Тогда с учётом (2.1), (2.3), (2.4) уравнение (2.2) примет вид

$$\begin{aligned} \mathcal{S}(x) = \frac{I_0}{2} E_2(\alpha x) + \frac{s}{2} \int_0^L E_1(\alpha|x-x'|) \mathcal{S}(x') dx' + \\ + \frac{\varkappa \sigma_0}{2} \int_0^L E_1(\alpha|x-x'|) u^4(x') dx' \quad (2.5) \end{aligned}$$

Введём оператор

$$\Lambda[\Psi](x) = \frac{1}{2} \int_0^L E_1(\alpha|x-x'|) \Psi(x') dx'$$

Тогда уравнение (2.5) примет вид

$$\mathcal{S}(x) = \frac{I_0}{2} E_2(\alpha x) + s\Lambda[\mathcal{S}](x) + \varkappa\sigma_0\Lambda[u^4](x) \quad (2.6)$$

2.2. Решение интегрального уравнения переноса излучения

Интегральное уравнение переноса излучения (2.6) будем решать методом Галёркина с кусочно-постоянными базисными функциями ([4], Гл. 16, §5).

Введём на отрезке $[0, L]$ две сетки (Рис. 2.1).

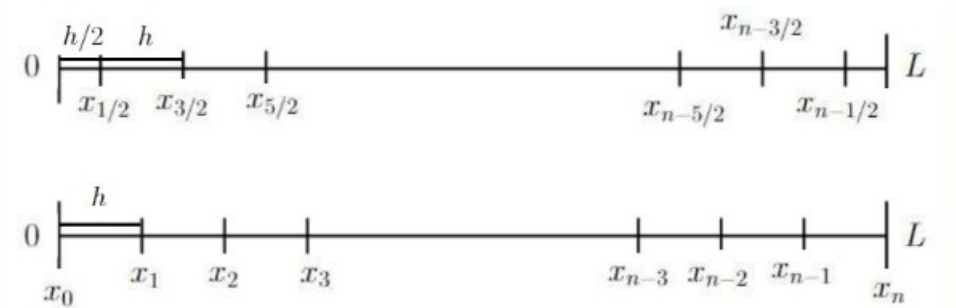


Рис. 2.1. Сетки

Заметим, что

$$x_{1/2} = \frac{h}{2}, \quad x_{n-1/2} = L - \frac{h}{2} \quad (2.7)$$

Для связи целых точек с полуцелыми на первой сетке будем использовать

$$x_j = jh - \frac{h}{2} \quad \forall j \in \overline{[2, n-2]} \quad (2.8)$$

В качестве базисных функций используем функции $\chi_{j-1/2}(x)$, где

$$\chi_{j-1/2}(x) = \begin{cases} 1, & x \in [x_{j-1}, x_j) \\ 0, & x \notin [x_{j-1}, x_j) \end{cases}$$

В дальнейшем для обозначения интервала $[x_{j-1}, x_j)$ будем использовать \mathfrak{X}_j .

Будем искать решение уравнения (2.6) в виде

$$\mathcal{S}^h(x) = \sum_{j=1}^n c_j \chi_{j-1/2}(x)$$

Заметим, что невязка, соответствующая приближённому решению $\mathcal{S}^h(x)$, имеет вид

$$r\mathcal{S}^h(x) = \mathcal{S}^h(x) - \frac{I_0}{2}E_2(\alpha x) - s\Lambda[\mathcal{S}^h](x) - \varkappa\sigma_0\Lambda[u^4](x)$$

Потребуем, чтобы невязка была ортогональна всем базисным функциям

$$\langle r\mathcal{S}^h, \chi_{i-1/2} \rangle = 0 \quad \forall i \in \overline{[1, n]}$$

Тогда получим

$$\begin{aligned} \langle \mathcal{S}^h, \chi_{i-1/2} \rangle &= \frac{I_0}{2} \langle E_2(\alpha x), \chi_{i-1/2} \rangle + \\ &+ s \langle \Lambda[\mathcal{S}^h], \chi_{i-1/2} \rangle + \varkappa\sigma_0 \langle \Lambda[u^4], \chi_{i-1/2} \rangle \quad \forall i \in \overline{[1, n]} \end{aligned}$$

В результате придём к следующей системе уравнений

$$\begin{aligned} \sum_{j=1}^n c_j \langle \chi_{j-1/2}, \chi_{i-1/2} \rangle &= \frac{I_0}{2} \langle E_2(\alpha x), \chi_{i-1/2} \rangle + \\ &+ s \sum_{j=1}^n c_j \langle \Lambda[\chi_{j-1/2}], \chi_{i-1/2} \rangle + \varkappa\sigma_0 \langle \Lambda[u^4], \chi_{i-1/2} \rangle \quad \forall i, j \in \overline{[1, n]} \quad (2.9) \end{aligned}$$

Далее отдельно рассмотрим каждое скалярное произведение, входящее в (2.9)

- $\langle \chi_{j-1/2}, \chi_{i-1/2} \rangle$

$$\langle \chi_{j-1/2}, \chi_{i-1/2} \rangle = \int_0^L \chi_{j-1/2} \chi_{i-1/2} dx$$

1. Пусть $i \neq j$. Тогда

$$\begin{aligned} \int_0^L \chi_{j-1/2} \chi_{i-1/2} dx &= \int_{\mathfrak{x}_j} \chi_{i-1/2} dx + \\ &+ \int_{\mathfrak{x}_i} \chi_{j-1/2} dx + \int_{[0, L] \setminus (\mathfrak{x}_j \cup \mathfrak{x}_i)} \chi_{i-1/2} \chi_{j-1/2} dx = 0 \end{aligned}$$

2. Пусть $i = j = 1$. Тогда

$$\int_0^L \chi_{j-1/2} \chi_{i-1/2} dx = \int_{\mathfrak{x}_1} \chi_{1/2}^2 dx + \int_{[0,L] \setminus \mathfrak{x}_1} \chi_{1/2}^2 dx = \int_{\mathfrak{x}_1} dx = x_1 - x_0 = \frac{h}{2}$$

3. Пусть $i = j = n$. Тогда

$$\begin{aligned} \int_0^L \chi_{j-1/2} \chi_{i-1/2} dx &= \int_{\mathfrak{x}_n} \chi_{n-1/2}^2 dx + \int_{[0,L] \setminus \mathfrak{x}_n} \chi_{n-1/2}^2 dx = \\ &= \int_{\mathfrak{x}_n} dx = x_n - x_{n-1} = \frac{h}{2} \end{aligned}$$

4. Пусть $(i = j) \wedge (i \notin \{1, n\})$. Тогда

$$\begin{aligned} \int_0^L \chi_{j-1/2} \chi_{i-1/2} dx &= \int_{\mathfrak{x}_i} \chi_{i-1/2}^2 dx + \\ &+ \int_{[0,L] \setminus \mathfrak{x}_i} \chi_{i-1/2}^2 dx = \int_{\mathfrak{x}_i} dx = x_i - x_{i-1} = h \end{aligned}$$

Итого

$$\langle \chi_{j-1/2}, \chi_{i-1/2} \rangle = \begin{cases} 0, & i \neq j \\ \frac{h}{2}, & (i = j = 1) \vee (i = j = n) \\ h, & (i = j) \wedge (i \notin \{1, n\}) \end{cases} \quad (2.10)$$

• $\langle E_2(\alpha x), \chi_{i-1/2} \rangle$

$$\begin{aligned} \langle E_2(\vartheta x), \chi_{i-1/2} \rangle &= \int_0^L E_2(\alpha x) \chi_{i-1/2} dx = \int_{\mathfrak{x}_i} E_2(\alpha x) dx = -\frac{1}{\alpha} E_3(\alpha x) \Big|_{x_{i-1}}^{x_i} = \\ &= -\frac{1}{\alpha} (E_3(\alpha x_i) - E_3(\alpha x_{i-1})) = \\ &= -\frac{1}{\alpha} (E_3(\alpha h(i - 1/2)) - E_3(\alpha h(i - 3/2))) \quad (2.11) \end{aligned}$$

- $\langle \Lambda[\chi_{j-1/2}], \chi_{i-1/2} \rangle$

$$\begin{aligned} \langle \Lambda[\chi_{j-1/2}], \chi_{i-1/2} \rangle &= \int_0^L \Lambda[\chi_{j-1/2}] \chi_{i-1/2} dx = \int_{\mathfrak{x}_i} \Lambda[\chi_{j-1/2}] dx = \\ &= \frac{1}{2} \int_{\mathfrak{x}_i} \int_0^L E_1(\alpha|x-x'|) \chi_{j-1/2} dx' dx = \frac{1}{2} \int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x-x'|) dx' dx \end{aligned}$$

Рассмотрим двойной интеграл

$$\int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x-x'|) dx' dx$$

1. Пусть $i < j$. Тогда

$$\begin{aligned} \int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x-x'|) dx' dx &= \int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha(x'-x)) dx' dx = \\ &= \frac{1}{\alpha} \int_{\mathfrak{x}_i} -E_2(\alpha(x'-x)) \Big|_{x_{j-1}}^{x_j} dx = \frac{1}{\alpha} \int_{\mathfrak{x}_i} -E_2(\alpha(x_j-x)) + E_2(\alpha(x_{j-1}-x)) dx \\ &= \frac{1}{\alpha^2} \left(-E_3(\alpha(x_j-x)) + E_3(\alpha(x_{j-1}-x)) \right) \Big|_{x_{i-1}}^{x_i} = \\ &= \frac{1}{\alpha^2} \left(-E_3(\alpha(x_j-x_i)) + E_3(\alpha(x_j-x_{i-1})) + E_3(\alpha(x_{j-1}-x_i)) - \right. \\ &\quad \left. -E_3(\alpha(x_{j-1}-x_{i-1})) \right) \end{aligned}$$

С учётом (2.8) получим

$$\begin{aligned} \int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x-x'|) dx' dx &= \frac{1}{\alpha^2} \left(E_3(\alpha h(j-i+1)) + \right. \\ &\quad \left. + E_3(\alpha h(j-i-1)) - 2E_3(\alpha h(j-i)) \right) = \Xi_1 \quad (2.12) \end{aligned}$$

2. Пусть $i > j$. Тогда аналогично п.1 получим

$$\int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x - x'|) dx' dx = \frac{1}{\alpha^2} \left(E_3(\alpha h(i - j - 1)) + \right. \\ \left. + E_3(\alpha h(i - j + 1)) - 2E_3(\alpha h(i - j)) \right) = \Xi_2 \quad (2.13)$$

3. Пусть $i = j$

В данном случае каждый квадрат, по которому происходит интегрирование, располагается на диагонали квадрата $[0, L] \times [0, L]$. Будем интегрировать по закрашенному треугольнику. В этой области $x > x'$ (Рис. 2.2). Поэтому имеем

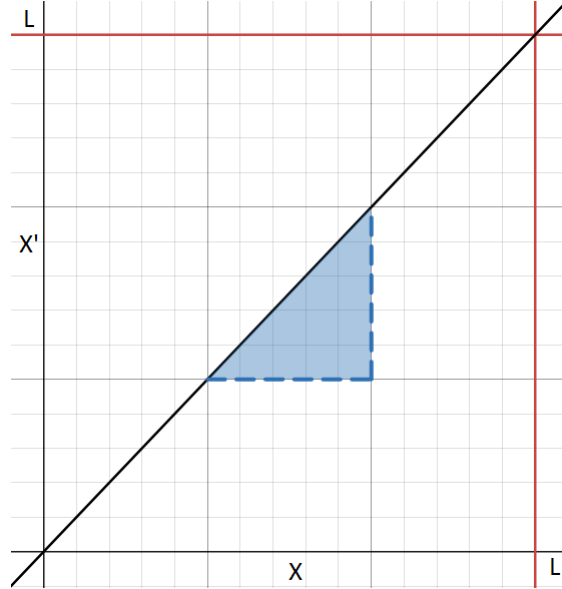


Рис. 2.2. Область интегрирования

$$\int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x - x'|) dx' dx = 2 \iint_{\Delta} E_1(\alpha(x - x')) dx' dx = \\ = 2 \int_{\mathfrak{x}_i} dx \int_{x_{j-1}}^x E_1(\alpha(x - x')) dx' = \frac{2}{\alpha} \int_{\mathfrak{x}_i} E_2(\alpha(x - x')) \Big|_{x_{j-1}}^x dx =$$

$$\begin{aligned}
&= \frac{2}{\alpha} \int_{\mathfrak{x}_i} E_2(\alpha(x - x)) - E_2(\alpha(x - x_{j-1})) dx = \\
&= \frac{2}{\alpha} \int_{\mathfrak{x}_i} dx - \frac{2}{\alpha} \int_{\mathfrak{x}_i} E_2(\alpha(x - x_{j-1})) dx = \\
&= \frac{2}{\alpha} h + \frac{2}{\alpha^2} E_3(\alpha(x - x_{j-1})) \Big|_{x_{i-1}}^{x_i} = \\
&= \frac{2}{\alpha} h + \frac{2}{\alpha^2} \left(E_3(\alpha(x_i - x_{j-1})) - E_3(\alpha(x_{i-1} - x_{j-1})) \right) = \\
&= \frac{2}{\alpha} h + \frac{2}{\alpha^2} \left(E_3(\alpha(ih - (j-1)h)) - E_3(\alpha((i-1)h - (j-1)h)) \right)
\end{aligned}$$

С учётом (2.8) получим

$$\int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x - x'|) dx' dx = \frac{2}{\alpha} h + \frac{2}{\alpha^2} E_3(\alpha h) - \frac{1}{\alpha^2} = \Xi_3 \quad (2.14)$$

Отдельно рассмотрим случаи для $j \in \{1, n\}$

4. Пусть $i > j = 1$. Тогда с учётом (2.7), (2.8), (2.13) получим

$$\begin{aligned}
\int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x - x'|) dx' dx &= \frac{1}{\alpha^2} \left(-E_3(\alpha h(i-1)) + E_3(\alpha h(i-2)) + \right. \\
&\quad \left. + E_3(\alpha h(i-1/2)) - E_3(\alpha h((i-3/2))) \right) = \Xi_4 \quad (2.15)
\end{aligned}$$

5. Пусть $(i = j = 1) \vee (i = j = n)$. Тогда с учётом (2.7), (2.8), (2.14) получим

$$\int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x - x'|) dx' dx = \frac{1}{\alpha} h + \frac{2}{\alpha^2} E_3\left(\alpha \frac{h}{2}\right) - \frac{1}{\alpha^2} = \Xi_5 \quad (2.16)$$

6. Пусть $i < j = n$. Тогда с учётом (2.7), (2.8), (2.12) получим

$$\begin{aligned}
\int_{\mathfrak{x}_i} \int_{\mathfrak{x}_j} E_1(\alpha|x - x'|) dx' dx &= \frac{1}{\alpha^2} \left(-E_3(\alpha(L + h(1/2 - i))) + \right. \\
&\quad \left. + E_3(\alpha(L + h(3/2 - i))) + E_3(\alpha(L - ih)) - \right. \\
&\quad \left. - E_3(\alpha(L - (i-1)h)) \right) = \Xi_6 \quad (2.17)
\end{aligned}$$

Объединив п.1-п.6, имеем

$$\int_{\mathfrak{X}_i} \int_{\mathfrak{X}_j} E_1(\alpha|x - x'|) dx' dx = \begin{cases} \Xi_5, & (i = j = 1) \vee (i = j = n) \\ \Xi_4, & i > j = 1 \\ \Xi_6, & i < j = n \\ \Xi_1, & (i < j) \wedge (j \notin \{1, n\}) \\ \Xi_2, & (i > j) \wedge (j \notin \{1, n\}) \\ \Xi_3, & (i = j) \wedge (j \notin \{1, n\}) \end{cases} \quad (2.18)$$

• $\langle \Lambda[u^4], \chi_{i-1/2} \rangle$

$$\begin{aligned} \langle \Lambda[u^4], \chi_{i-1/2} \rangle &= \int_0^L \Lambda[u^4] \chi_{i-1/2} dx = \int_{\mathfrak{X}_i} \Lambda[u^4] dx = \\ &= \frac{1}{2} \int_{\mathfrak{X}_i} \int_0^L E_1(\alpha|x - x'|) u^4(x') dx' dx \end{aligned} \quad (2.19)$$

Пусть на интервале \mathfrak{X}_j $u^4(x) \approx u_j^4 \quad \forall j \in \overline{[1, n]}$. Тогда (2.19) примет вид

$$\begin{aligned} \frac{1}{2} \int_{\mathfrak{X}_i} \int_0^L E_1(\alpha|x - x'|) u^4(x') dx' dx &= \\ &= \frac{1}{2} u_j^4 \sum_{j=1}^n \int_{\mathfrak{X}_i} \int_{\mathfrak{X}_j} E_1(\alpha|x - x'|) dx' dx, \end{aligned} \quad (2.20)$$

где

$$\int_{\mathfrak{X}_i} \int_{\mathfrak{X}_j} E_1(\alpha|x - x'|) dx' dx = \begin{cases} \Xi_1, & i < j \\ \Xi_2, & i > j \\ \Xi_3, & i = j \end{cases}$$

Тогда с учётом (2.10), (2.11), (2.18), (2.20) система уравнений (2.9) примет вид

$$\mathbf{A}^h c = \frac{I_0}{2} \mathbf{e}^h + \frac{s}{2} \mathbf{E}^h c + \kappa \sigma_0 \mathbf{K}^h u^4, \quad (2.21)$$

где \mathbf{A}^h матрица с элементами

$$a_{i,j} = \begin{cases} \frac{h}{2}, & (i = j = 1) \vee (i = j = n) \\ h, & (i = j) \wedge (i \notin \{1, n\}) \end{cases}$$

\mathbf{e}^h - вектор с элементами

$$\mathbf{e}_i^h = -\frac{1}{\alpha} \left(E_3(\alpha h(i - 1/2)) - E_3(\alpha h(i - 3/2)) \right)$$

\mathbf{E}^h - матрица с элементами

$$e_{i,j} = \begin{cases} \Xi_5, & (i = j = 1) \vee (i = j = n) \\ \Xi_4, & i > j = 1 \\ \Xi_6, & i < j = n \\ \Xi_1, & (i < j) \wedge (j \notin \{1, n\}) \\ \Xi_2, & (i > j) \wedge (j \notin \{1, n\}) \\ \Xi_3, & (i = j) \wedge (j \notin \{1, n\}) \end{cases}$$

\mathbf{K}^h - матрица с элементами

$$\mathbf{K}^h = \begin{cases} \frac{1}{2}\Xi_1, & i < j \\ \frac{1}{2}\Xi_2, & i > j \\ \frac{1}{2}\Xi_3, & i = j \end{cases}$$

Перепишем (2.21) в ином виде

$$\mathbf{B}^h c = \frac{I_0}{2} \mathbf{e}^h + \varkappa \sigma_0 \mathbf{K}^h u^4, \quad (2.22)$$

где $\mathbf{B}^h = \mathbf{A}^h - \frac{s}{2} \mathbf{E}^h$

В силу линейности системы (2.22) будем решать её в несколько этапов

1. Решим систему

$$\mathbf{B}^h c' = \frac{1}{2} \mathbf{e}^h$$

и умножим полученный вектор c' на I_0 и получим вектор c .

2. Решим систему

$$\mathbf{B}^h c_i = \mathbf{k}_i^h \quad \forall i \in \overline{[1, n]},$$

где \mathbf{k}_i^h - i -ый столбец матрицы \mathbf{K}^h . Далее составим матрицу \mathbf{S}^h , в которой на месте i -го столбца стоит вектор c_i .

В итоге приближённым решением уравнения (2.6) в точке $x_{j-1/2}$ будет следующая комбинация

$$\mathcal{S}(x_{j-1/2}) \approx \kappa \sigma_0 \sum_{\ell=1}^n \beta_{j,\ell} u_\ell^4 + c_j \quad \forall j \in \overline{[1, n]}, \quad (2.23)$$

где $\beta_{j,\ell}$ - элемент на пересечении j -ой строки и ℓ -ого столбца в матрице \mathbf{S}^h , c_j - j -ая компонента вектора c .

Стоит отметить, что полученные коэффициенты $\beta_{j,\ell}$ показывают, как влияет температура определённой точки слоя на температуру в точке x_j .

2.3. Решение уравнения теплопроводности

Рассмотрим уравнение теплопроводности (1.1) с краевыми условиями (1.2), (1.3). Для построения задачи, аппроксимирующей (1.1) – (1.3) будем использовать метод баланса.

Распишем баланс в точке x_j . Слагаемые без дифференцирования заменим их значениями в точке x_j . Проинтегрируем уравнение (1.1) по отрезку $[x_{j-1/2}, x_{j+1/2}]$. Также учтём (2.23). Тогда получим

$$\begin{aligned} \int_{x_{j-1/2}}^{x_{j+1/2}} -\lambda \frac{d^2 u}{dx^2} + \kappa \sigma_0 u_j^4 \, dx &= \int_{x_{j-1/2}}^{x_{j+1/2}} \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{j,\ell} u_\ell^4 + \kappa c_j \, dx \quad \forall j \in \overline{[1, n-1]} \\ -\lambda \frac{du}{dx} \Big|_{x_{j+1/2}} + \lambda \frac{du}{dx} \Big|_{x_{j-1/2}} + \kappa \sigma_0 u_j^4 (x_{j+1/2} - x_{j-1/2}) &= \\ &= \left(\kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{j,\ell} u_\ell^4 \right) (x_{j+1/2} - x_{j-1/2}) + \kappa c_j (x_{j+1/2} - x_{j-1/2}) \end{aligned} \quad (2.24)$$

Производные в (2.24) заменим разностными аналогами, а также учтём, что $x_{j+1/2} - x_{j-1/2} = h$. Тогда (2.24) примет вид

$$-\lambda \frac{u_{j+1} - u_j}{h} + \lambda \frac{u_j - u_{j-1}}{h} + \kappa \sigma_0 u_j^4 h = h \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{j,\ell} u_\ell^4 + h \kappa c_j$$

Полученное уравнение разделим на h . В результате получим

$$-\frac{\lambda}{h^2}(u_{j+1} - 2u_j + u_{j-1}) + \kappa \sigma_0 u_j^4 - \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{j,\ell} u_\ell^4 - \kappa c_j = 0 \quad \forall j \in \overline{[1, n-1]} \quad (2.25)$$

Аппроксимируем краевое условие (1.2). Для этого проинтегрируем (1.1) по отрезку $[x_0, x_{1/2}]$. Заметим, что в данном случае $j = 0$.

$$\int_{x_0}^{x_{1/2}} -\lambda \frac{d^2 u}{dx^2} + \kappa \sigma_0 u_0^4 \, dx = \int_{x_0}^{x_{1/2}} \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{0,\ell} u_\ell^4 + \kappa c_0 \, dx$$

$$\begin{aligned} -\lambda \frac{du}{dx} \Big|_{x_{1/2}} + \lambda \frac{du}{dx} \Big|_{x_0} + \kappa \sigma_0 u_0^4 (x_{1/2} - x_0) = \\ = \left(\kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{0,\ell} u_\ell^4 \right) (x_{1/2} - x_0) + \kappa c_0 (x_{1/2} - x_0) \end{aligned}$$

Учтем (1.2), а также, что $x_{1/2} - x_0 = \frac{h}{2}$, разделим на h . Тогда получим

$$-\frac{\lambda}{h^2}(u_1 - u_0) + \frac{1}{2} \left(\kappa \sigma_0 u_0^4 - \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{0,\ell} u_\ell^4 - \kappa c_0 \right) = 0 \quad (2.26)$$

Далее интегрируя (1.1) по отрезку $[x_{n-1/2}, x_n]$, аналогичным образом получим аппроксимацию для краевого условия (1.3), которая будет выглядеть следующим образом

$$\frac{\lambda}{h^2}(u_{n-1} - u_{n-2}) + \frac{1}{2} \left(\kappa \sigma_0 u_{n-1}^4 - \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{n-1,\ell} u_\ell^4 - \kappa c_{n-1} \right) = 0 \quad (2.27)$$

Объединяя (2.25) – (2.27) получаем систему нелинейных уравнений относительно вектора $\mathbf{u} = (u_0, u_1, \dots, u_{n-1})$, которую будем решать методом Ньютона.

2.4. Метод Ньютона

Введём следующие вспомогательные функции

$$\mathcal{F}_0(u_0, u_1, \dots, u_{n-1}) = -\frac{\lambda}{h^2}(u_1 - u_0) + \frac{1}{2} \left(\kappa \sigma_0 u_0^4 - \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{0,\ell} u_\ell^4 - \kappa c_0 \right)$$

$$\begin{aligned} \mathcal{F}_j(u_0, u_1, \dots, u_{n-1}) = & -\frac{\lambda}{h^2}(u_{j+1} - 2u_j + u_{j-1}) + \\ & + \kappa \sigma_0 u_j^4 - \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{j,\ell} u_\ell^4 - \kappa c_j \quad \forall j \in \overline{[1, n-2]} \end{aligned}$$

$$\begin{aligned} \mathcal{F}_{n-1}(u_0, u_1, \dots, u_{n-1}) = & \frac{\lambda}{h^2}(u_{n-1} - u_{n-2}) + \\ & + \frac{1}{2} \left(\kappa \sigma_0 u_{n-1}^4 - \kappa^2 \sigma_0 \sum_{\ell=0}^{n-1} \beta_{n-1,\ell} u_\ell^4 - \kappa c_{n-1} \right) \end{aligned}$$

Тогда систему (2.25) – (2.27) можно переписать в виде

$$\begin{cases} \mathcal{F}_0(u_0, u_1, \dots, u_{n-1}) = 0 \\ \mathcal{F}_1(u_0, u_1, \dots, u_{n-1}) = 0 \\ \dots \\ \mathcal{F}_{n-2}(u_0, u_1, \dots, u_{n-1}) = 0 \\ \mathcal{F}_{n-1}(u_0, u_1, \dots, u_{n-1}) = 0 \end{cases} \quad (2.28)$$

Частные производные функций \mathcal{F}_j по соответствующим аргументам будут выглядеть следующим образом

$$\frac{\partial \mathcal{F}_0}{\partial u_0} = \frac{\lambda}{h^2} + 2\kappa \sigma_0 u_0^3 - 2\kappa^2 \sigma_0 \beta_{0,0} |u_0|^3 = \frac{\lambda}{h^2} + 2\kappa \sigma_0 (1 - \kappa \beta_{0,0}) |u_0|^3$$

$$\frac{\partial \mathcal{F}_0}{\partial u_1} = -\frac{\lambda}{h^2} - 2\kappa^2 \sigma_0 \beta_{0,1} |u_1|^3$$

$$\frac{\partial \mathcal{F}_0}{\partial u_m} = -2\kappa^2 \sigma_0 \beta_{0,m} |u_m|^3 \quad \forall m \in \overline{[2, n-1]}$$

$$\begin{aligned}
\frac{\partial \mathcal{F}_j}{\partial u_{j-1}} &= -\frac{\lambda}{h^2} - 4\kappa^2 \sigma_0 \beta_{j,j-1} |u_{j-1}|^3 \\
\frac{\partial \mathcal{F}_j}{\partial u_j} &= \frac{2\lambda}{h^2} + 4\kappa \sigma_0 |u_j|^3 - 4\kappa^2 \sigma_0 \beta_{j,j} |u_j|^3 = \frac{2\lambda}{h^2} + 4\kappa \sigma_0 (1 - \kappa \beta_{j,j}) |u_j|^3 \\
\frac{\partial \mathcal{F}_j}{\partial u_{j+1}} &= -\frac{\lambda}{h^2} - 4\kappa^2 \sigma_0 \beta_{j,j+1} |u_{j+1}|^3 \\
\frac{\partial \mathcal{F}_j}{\partial u_m} &= -4\kappa^2 \sigma_0 \beta_{j,m} |u_m|^3 \quad \forall m \in \overline{[0, n-1]} \setminus \{j-1, j, j+1\} \\
\frac{\partial \mathcal{F}_{n-1}}{\partial u_{n-1}} &= \frac{\lambda}{h^2} + 2\kappa \sigma_0 |u_{n-1}|^3 - 2\kappa^2 \sigma_0 \beta_{n-1,n-1} |u_{n-1}|^3 = \frac{\lambda}{h^2} + 2\kappa \sigma_0 (1 - \kappa \beta_{n-1,n-1}) |u_{n-1}|^3 \\
\frac{\partial \mathcal{F}_{n-1}}{\partial u_{n-2}} &= -\frac{\lambda}{h^2} - 2\kappa^2 \sigma_0 \beta_{n-1,n-2} |u_{n-2}|^3 \\
\frac{\partial \mathcal{F}_{n-1}}{\partial u_m} &= -2\kappa^2 \sigma_0 \beta_{n-1,m} |u_m|^3 \quad \forall m \in \overline{[0, n-3]}
\end{aligned}$$

Каждое значение куба температуры было заменено на соответствующее абсолютное значение, чтобы в процессе итерирования метода Ньютона не появлялось отрицательных значений, которые могут нарушить сходимость.

Пусть ранее было построено приближение $\mathbf{u}^{(k)}$ искомого вектора \mathbf{u} . Заменим в системе (2.28) каждую функцию линейной частью её разложения по формуле Тейлора в точке $\mathbf{u}^{(k)}$.

Каждое следующее приближение $\mathbf{u}^{(k+1)}$ удовлетворяет равенству

$$\mathbf{F}(\mathbf{u}^{(k)}) + \nabla \mathbf{F}^{(k)}(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}) = 0,$$

которое можно переписать в виде

$$\nabla \mathbf{F}^{(k)} \Delta \mathbf{u}^{(k+1)} = -\mathbf{F}(\mathbf{u}^{(k)}),$$

где $\Delta \mathbf{u}^{(k+1)} = \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$.

2.5. Форма матрицы Якоби

Выпишем отдельно некоторые строки матрицы Якоби $\nabla \mathbf{F}^{(k)}$

Первая строка ($j = 0$)

$$\nabla \mathbf{F}_0^{(k)} = \begin{bmatrix} \frac{\lambda}{h^2} + 2\kappa\sigma_0(1 - \kappa\beta_{0,0})|u_0|^{3-2} & -\frac{\lambda}{h^2} - 2\kappa^2\sigma_0\beta_{0,1}|u_1|^{3-2} & -2\kappa^2\sigma_0\beta_{0,2}|u_2^{(k)}|^{3-2} & \dots \\ \dots & -2\kappa^2\sigma_0\beta_{0,n-2}|u_{n-2}^{(k)}|^{3-2} & -2\kappa^2\sigma_0\beta_{0,n-1}|u_{n-1}^{(k)}|^{3-2} & \end{bmatrix}$$

Вторая строка ($j = 1$)

$$\nabla \mathbf{F}_1^{(k)} = \begin{bmatrix} -\frac{\lambda}{h^2} + 4\kappa^2\sigma_0\beta_{1,0}|u_0^{(k)}|^{3-2} & \frac{2\lambda}{h^2} + 4\kappa\sigma_0(1 - \kappa\beta_{1,1})|u_1^{(k)}|^{3-2} & -\frac{\lambda}{h^2} - 4\kappa^2\sigma_0\beta_{1,2}|u_2^{(k)}|^{3-2} \\ -4\kappa^2\sigma_0\beta_{1,3}|u_3^{(k)}|^{3-2} & -4\kappa^2\sigma_0\beta_{1,4}|u_4^{(k)}|^{3-2} & \dots & -4\kappa^2\sigma_0\beta_{1,n-1}|u_{n-1}^{(k)}|^{3-2} \end{bmatrix}$$

Третья строка ($j = 2$)

$$\nabla \mathbf{F}_2^{(k)} = \begin{bmatrix} -4\kappa^2\sigma_0\beta_{2,0}|u_0^{(k)}|^{3-2} & -\frac{\lambda}{h^2} - 4\kappa^2\sigma_0\beta_{2,1}|u_1^{(k)}|^{3-2} & \frac{2\lambda}{h^2} + 4\kappa\sigma_0(1 - \kappa\beta_{2,2})|u_2^{(k)}|^{3-2} \\ -\frac{\lambda}{h^2} - 4\kappa^2\sigma_0\beta_{2,3}|u_3^{(k)}|^{3-2} & -4\kappa^2\sigma_0\beta_{2,4}|u_4^{(k)}|^{3-2} & \dots & -4\kappa^2\sigma_0\beta_{2,n-1}|u_{n-1}^{(k)}|^{3-2} \end{bmatrix}$$

...

$(i + 1)$ -ая строка ($j = i$)

$$\nabla \mathbf{F}_i^{(k)} = \begin{bmatrix} -4\kappa^2\sigma_0\beta_{i,0}|u_0^{(k)}|^{3-2} & -4\kappa^2\sigma_0\beta_{i,1}|u_1^{(k)}|^{3-2} & \dots & -\frac{\lambda}{h^2} - 4\kappa^2\sigma_0\beta_{i,i-1}|u_{i-1}^{(k)}|^{3-2} \\ \frac{2\lambda}{h^2} + 4\kappa\sigma_0(1 - \kappa\beta_{i,i})|u_i^{(k)}|^{3-2} & -\frac{\lambda}{h^2} - 4\kappa^2\sigma_0\beta_{i,i+1}|u_{i+1}^{(k)}|^{3-2} & \dots & -4\kappa^2\sigma_0\beta_{i,n-1}|u_{n-1}^{(k)}|^{3-2} \end{bmatrix}$$

...

Последняя строка ($j = n - 1$)

$$\nabla \mathbf{F}_{n-1}^{(k)} = \begin{bmatrix} -2\kappa^2\sigma_0\beta_{n-1,0}|u_0^{(k)}|^{3-2} & -2\kappa^2\sigma_0\beta_{n-1,1}|u_1^{(k)}|^{3-2} & -2\kappa^2\sigma_0\beta_{n-1,2}|u_2^{(k)}|^{3-2} & \dots \\ \dots & -\frac{\lambda}{h^2} - 2\kappa^2\sigma_0\beta_{n-1,n-2}|u_{n-2}^{(k)}|^{3-2} & \frac{\lambda}{h^2} + 2\kappa\sigma_0(1 - \kappa\beta_{n-1,n-1})|u_{n-1}^{(k)}|^{3-2} & \end{bmatrix}$$

3. Результаты вычислительных экспериментов

Эксперименты проводились для различных значений коэффициентов теплопроводности, поглощения и рассеяния. Также варьировались следующие параметры:

- Толщина слоя
- Интенсивность излучения, падающего на верхнюю границу слоя

3.1. Слой воды толщиной 1 м

Средняя величина коэффициента теплопроводности λ для воды равна 0.6. Пусть на верхнюю поверхность падает излучение интенсивностью $I_0 = 800$.

Результаты для различных значений коэффициента поглощения при фиксированном коэффициенте рассеяния

Параметры

- κ меняется от 0.2 до 2
- $s = 0.15$

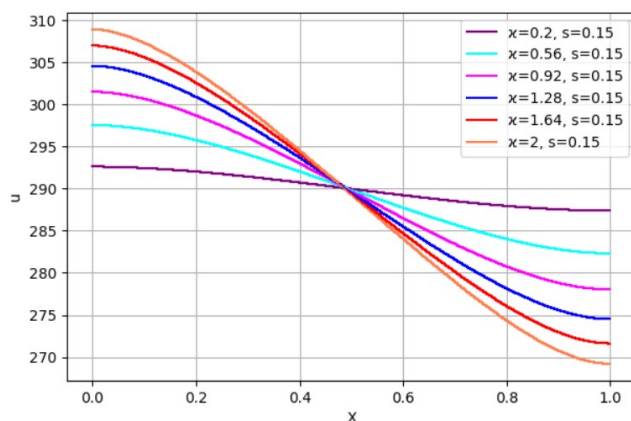


Рис. 3.1. Распределение температуры при различных значениях κ

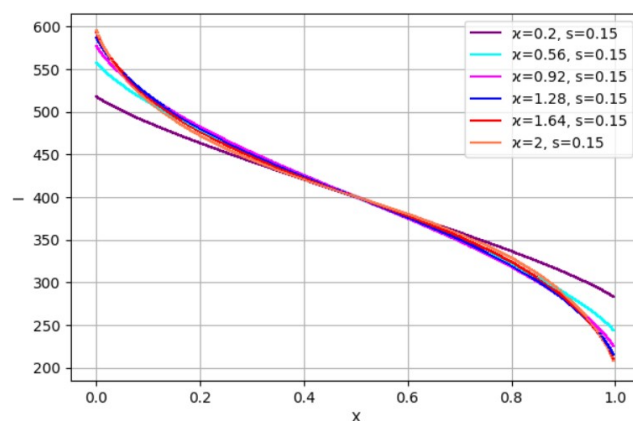


Рис. 3.2. Интенсивность излучения при различных значениях κ

Наибольшему значению коэффициента поглощения соответствует максимальный перепад температуры (Рис. 3.1), что логично, так как в этом случае большая часть энергии остаётся в верхней части слоя. Также заметим, что при $\kappa = 2$ начинается замерзание воды. Похожая ситуация наблюдается и для интенсивности излучения (Рис. 3.2).

Результаты для различных значений коэффициента рассеяния при фиксированном коэффициенте поглощения

Параметры

- $\kappa = 0.3$
- s меняется от 0.1 до 5

Графики распределения температуры стали более близкими друг к другу, что говорит о меньшей зависимости от s , чем от κ (Рис. 3.3). С интенсивностью излучения ситуация противоположная (Рис. 3.4).

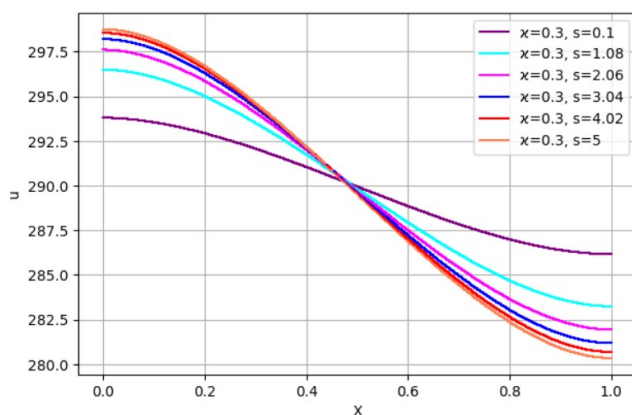


Рис. 3.3. Распределение температуры при различных значениях s

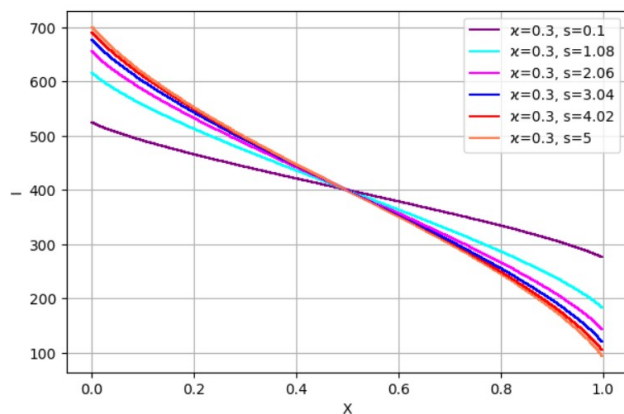


Рис. 3.4. Интенсивность излучения при различных значениях s

Результаты для различных значений интенсивности падающего излучения

Параметры

- $\kappa = 0.3$
- $s = 0.25$
- I_0 меняется от 600 до 1000 с шагом 100

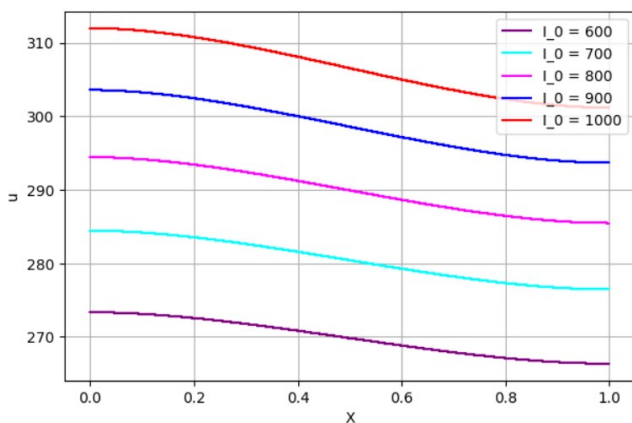


Рис. 3.5. Распределение температуры при различных значениях I_0

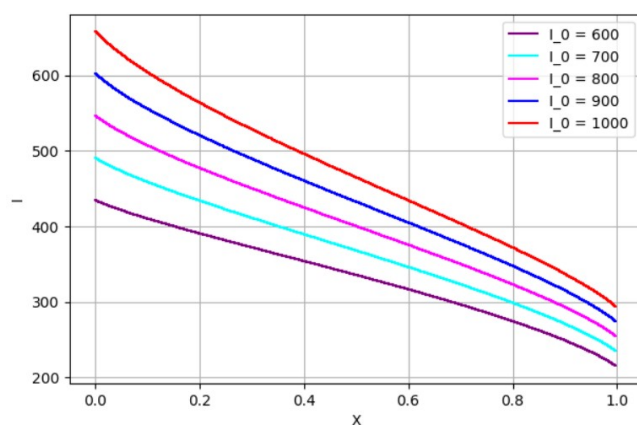


Рис. 3.6. Интенсивность излучения при различных значениях I_0

Поведение распределения температуры одинаковое, за исключением начального и конечного значений, что логично, так как κ , s одинаковые для

каждого случая (Рис. 3.5). Также заметим, что при $I_0 = 600$ начинается замерзание воды. Поведение интенсивности излучения похожее, но ближе к нижней поверхности слоя идёт сгущение (Рис. 3.6).

3.2. Слой воды толщиной 4 м

Результаты для различных значений коэффициента поглощения при фиксированном коэффициенте рассеяния

Параметры

- κ меняется от 0.2 до 2
- $s = 0.15$

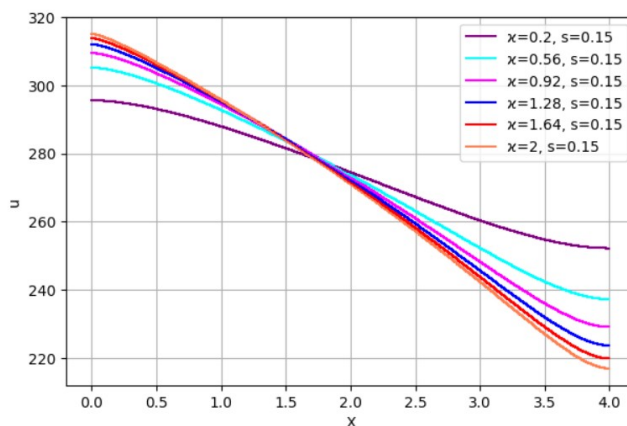


Рис. 3.7. Распределение температуры при различных значениях κ ($L = 4$)

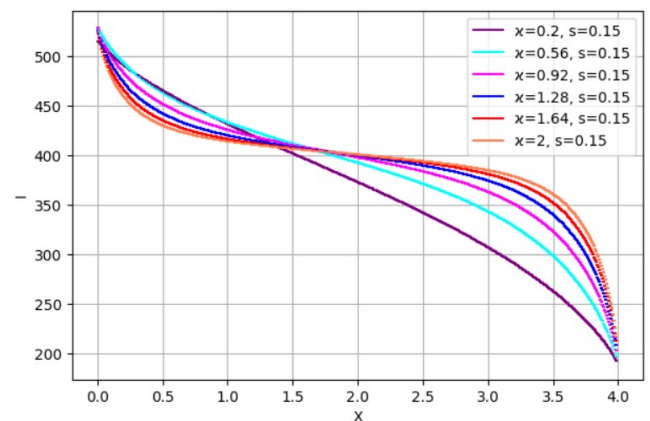


Рис. 3.8. Интенсивность излучения при различных значениях κ ($L = 4$)

Поведение распределения температуры совпадает со случаем $L = 1$, однако видно, что большая часть энергии задерживается в верхних частях слоя воды, от чего идёт большой перепад температуры (Рис. 3.7). В каждом случае начинается замерзание воды. Поведения интенсивности излучения меняется: с увеличением коэффициента поглощения зависимость становится всё более нелинейной (Рис. 3.8).

Результаты для различных значений коэффициента рассеяния при фиксированном коэффициенте поглощения

Параметры

- $\kappa = 0.3$
- s меняется от 0.1 до 5

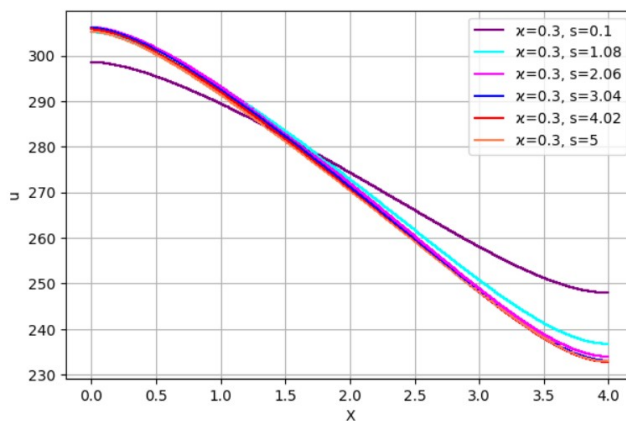


Рис. 3.9. Распределение температуры при различных значениях s ($L = 4$)

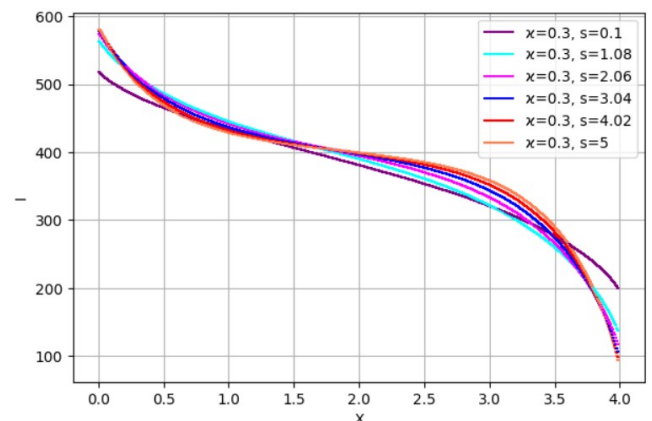


Рис. 3.10. Интенсивность излучения при различных значениях s ($L = 4$)

Как для распределения температуры (Рис. 3.9), так и для интенсивности излучения (Рис. 3.10) наблюдаются результаты схожие с таким же случаем при $L = 1$. Однако присутствуют особенности, описанные в предыдущем эксперименте.

Различные значения интенсивности падающего излучения

Параметры

- $\kappa = 0.3$
- $s = 0.25$
- I_0 меняется от 400 до 650

Результаты вновь похожи на тот же случай при $L = 1$, но вновь с ранее описанными особенностями (Рис. 3.11, Рис. 3.12).

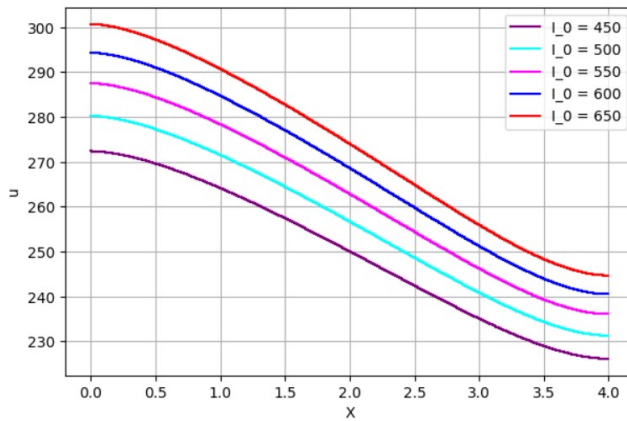


Рис. 3.11. Распределение температуры при различных значениях I_0 ($L = 4$)

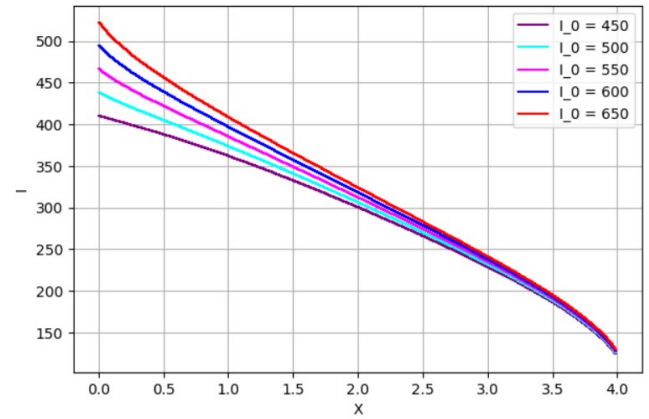


Рис. 3.12. Интенсивность излучения при различных значениях I_0 ($L = 4$)

3.3. Различные экстримальные ситуации для слоя воды толщиной 1 м

Малые значения коэффициента рассеяния при фиксированном значении коэффициента поглощения

Параметры

- $\kappa = 0.3$
- s меняется от 0 до 0.02

По мере уменьшения значений коэффициента рассеяния распределения температур (Рис. 3.13) практически совпадают. Аналогичный результат и для интенсивности излучения (Рис. 3.14).

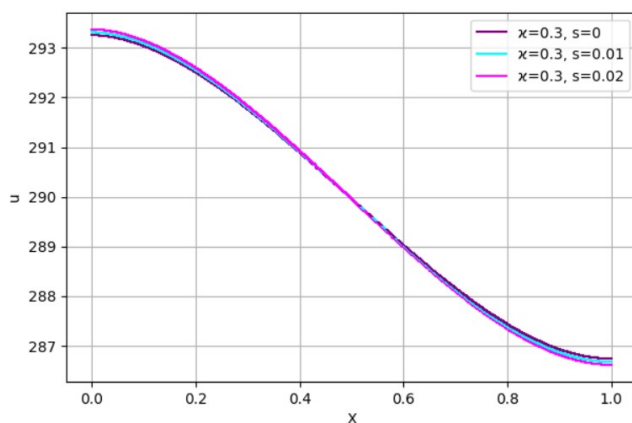


Рис. 3.13. Распределение температуры при малых значениях s

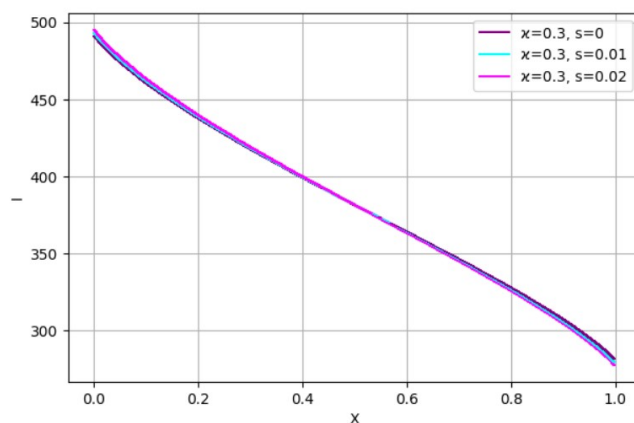


Рис. 3.14. Интенсивность излучения при малых значениях s

Малые значения коэффициента поглощения при фиксированном значении коэффициента рассеяния

Параметры

- $\chi = 0.1, \quad 0.01, \quad 0.001$
- $s = 0.15$

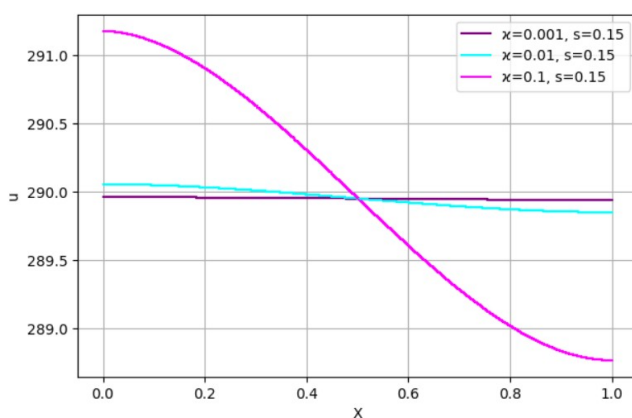


Рис. 3.15. Распределение температуры при малых значениях χ

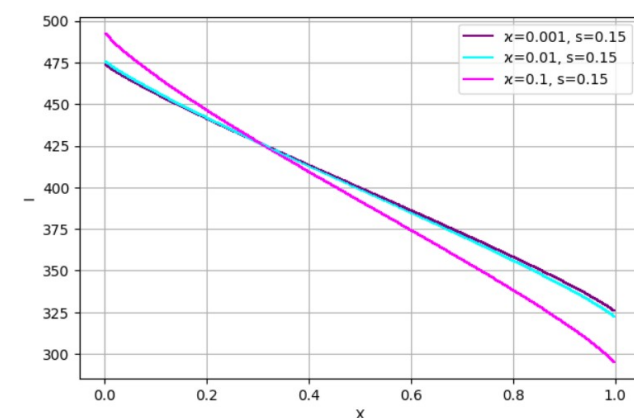


Рис. 3.16. Интенсивность излучения при малых значениях χ

При малых значениях χ изменение температуры практически отсутствует (Рис. 3.15). Интенсивность излучения, в свою очередь, начинает убывать медленнее, что логично, так как поглощение незначительно (Рис. 3.16).

Большие значения коэффициента рассеяния при постоянном коэффициенте поглощения

Параметры

- $\kappa = 0.3$
- s меняется от 10 до 100

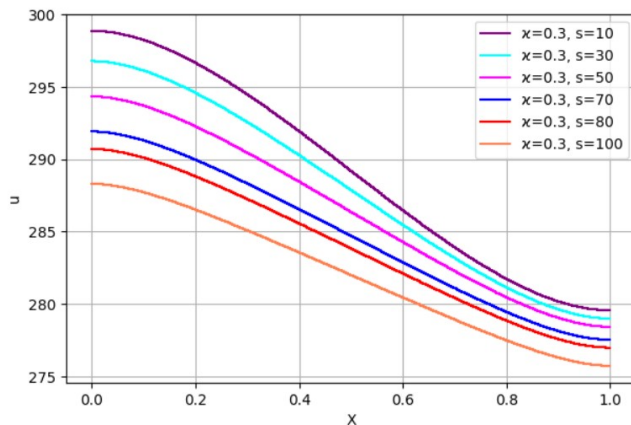


Рис. 3.17. Распределение температуры при больших значениях s

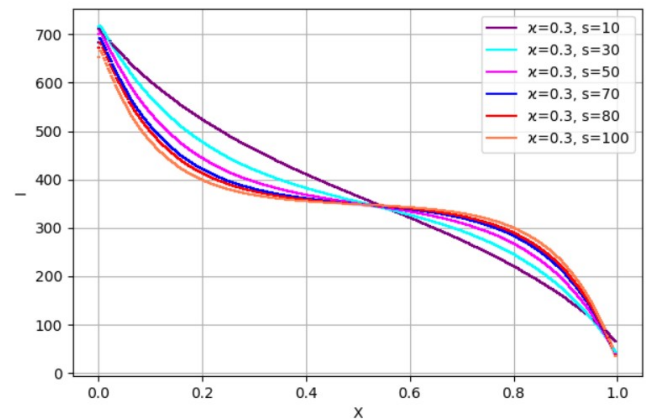


Рис. 3.18. Интенсивность излучения при больших значениях s

При больших значениях коэффициента рассеяния амплитуды температур становятся меньше (Рис. 3.17). Поведения интенсивности излучения меняется: с увеличением коэффициента рассеяния зависимость становится всё более нелинейной (Рис. 3.18).

Большие значения коэффициента поглощения при постоянном коэффициенте рассеяния

Параметры

- κ меняется от 10 до 100
- $s = 0.25$

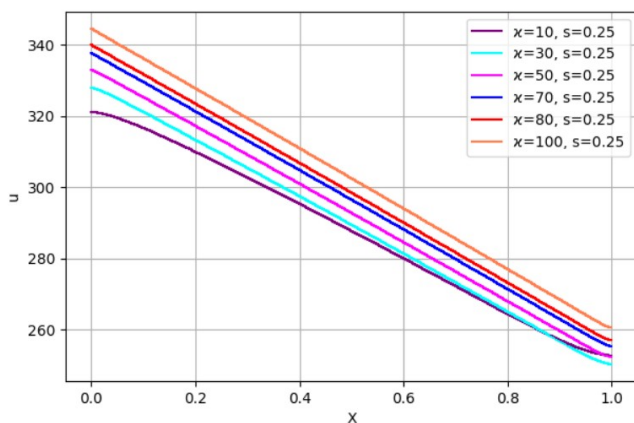


Рис. 3.19. Распределение температуры при больших значениях χ

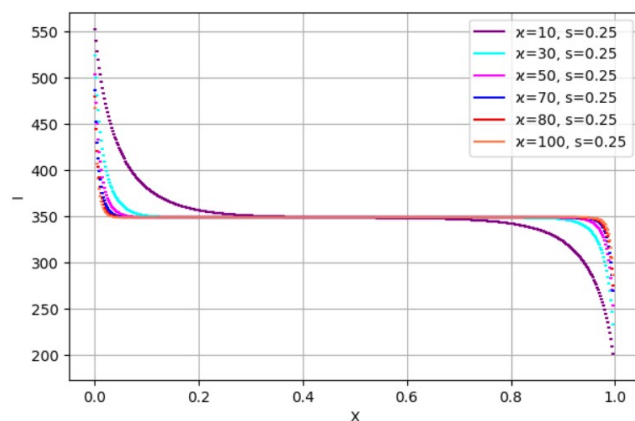


Рис. 3.20. Интенсивность излучения при больших значениях χ

При больших значениях коэффициента поглощения зависимость распределения температуры приближается к линейной (Рис. 3.19). Интенсивность излучения при тех же значениях убывает ближе к границам слоя (Рис. 3.20).

3.4. Другие вещества

Разные показатели теплопроводности

Коэффициент теплопроводности стекла λ приблизительно равен 1.15. Для кварца же этот показатель составляет 8, для нефтяных масел - 0.12. Рассмотрим, как изменятся результаты вычислений. Примем $\chi = 0.3$, $s = 0.25$.

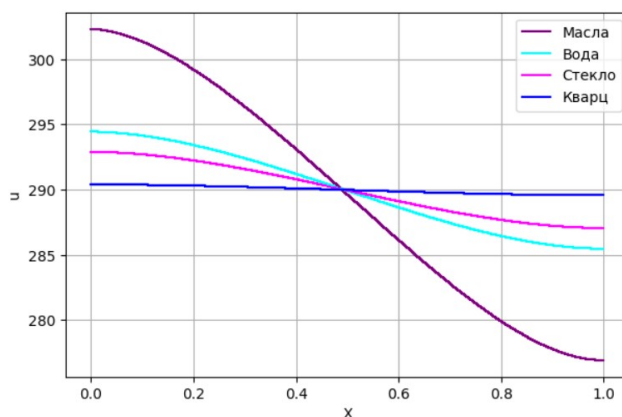


Рис. 3.21. Распределение температуры для разных веществ

Видим, что по мере увеличения коэффициента теплопроводности, уменьшается изменение температуры (Рис. 3.21).

Экстримально малые значения коэффициента теплопроводности

Параметры

- λ меняется от 0 до 0.1
- $s = 0.25$
- $\varkappa = 0.3$

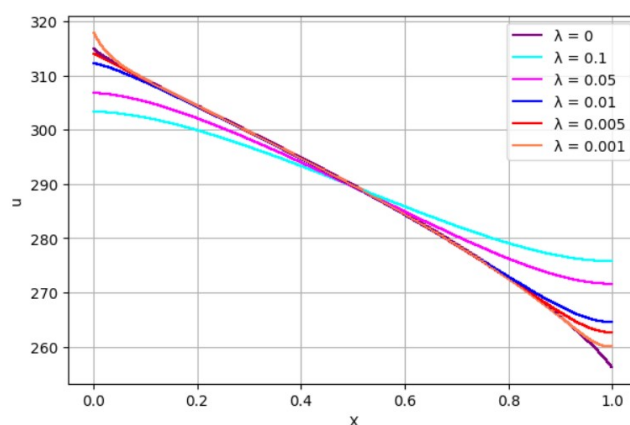


Рис. 3.22. Распределение температуры при малых значениях λ

При устремлении λ к нулю изменение температуры становится более резким (Рис. 3.22).

Экстримально большие значения коэффициента теплопроводности

Параметры

- λ меняется от 10 до 100

- $s = 0.25$

- $\varkappa = 0.3$

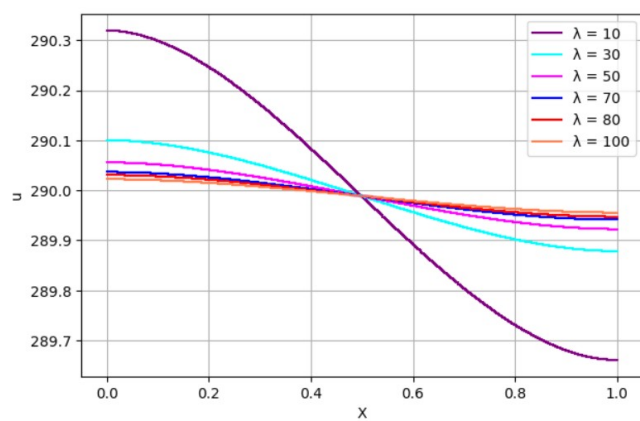


Рис. 3.23. Распределение температуры при больших значениях λ

При больших значениях коэффициента теплопроводности изменение температуры практически отсутствует. (Рис. 3.23)

Приложение 1. Интегро-экспоненциальная функция

В разделе 2.1 была введена интегро-экспоненциальная функция порядка n . Напомним её определение

$$E_n(x) = \int_0^1 \mu^{n-2} \exp\left(-\frac{x}{\mu}\right) d\mu$$

Приведем некоторые её свойства, используемые в настоящей работе [3]

1.

$$E_n(0) = \frac{1}{n-1}, \quad n \geq 2$$

2.

$$E_n(x) = \frac{(-x)^{n-1}}{(n-1)!} \left[-\ln x + \psi(n) \right] - \sum_{\substack{m=0 \\ m \neq n-1}}^{\infty} \frac{(-x)^m}{(m-n+1)m!},$$

где

$$\psi(n) = \begin{cases} -\gamma, & n = 1 \\ -\gamma + \sum_{m=1}^{n-1} \frac{1}{m}, & n > 1 \end{cases}$$

Здесь γ - константа Эйлера-Маскерони

3.

$$\frac{d}{dx} E_n(x) = -E_{n-1}(x), \quad n \geq 1$$

Приложение 2. Программный код

Ниже приведен код основной программы

```
import numpy as np
from matplotlib import pyplot as plt
import scipy

# exponential integral implementation
def psi(n):
    res = 0
    if (n == 1):
        res = -np.euler_gamma
    else:
        m = []
        for i in range(1, n, 1):
            m.append(i)
        sum = 0
        for i in range(len(m)):
            sum += 1/m[i]
        res = -np.euler_gamma + sum
    return res

def mCoeff(m, n):
    return (-1)**m/((m-n+1)*np.math.factorial(m))

def myEn(x0, m, n):
    row = x0*0
    for i in range(m):
        if (i != (n-1)):
            row += mCoeff(i, n)*(x0**i)
        else:
            row += 0
```

```

    res = ((-x0)**(n-1))/(np.math.factorial(n-1))*(-np.log(x0)+psi(n)) - row
    return res

def E1(x):
    return myEn(x, 200, 1)

def E2(x):
    return myEn(x, 200, 2)

def E3(x):
    return myEn(x, 200, 3)

#Gaussian method
def gauss(A, b):
    for j in range(len(A) - 1):
        for i in range(j + 1, len(A)):
            m = A[i, j] / A[j, j]
            A[i, j:] = A[i, j:] - m * A[j, j:]
            b[i] = b[i] - m * b[j]
    x = np.zeros(len(A))
    for i in range(len(A) - 1, -1, -1):
        x[i] = b[i] - np.sum(A[i, i + 1:] * x[i + 1:])
        x[i] /= A[i, i]
    return x

#grid point
def x(j, x_0, n, L):
    h_u = L/n
    if (j == 0):
        return x_0
    elif (j == n):
        return L
    else:
        return x_0 + j*h_u - h_u/2

#matrix A creation
def createA(L, n):
    h_u = L/n

```

```

A = np.zeros((n, n))
A[0, 0] = 1/2
A[-1, -1] = 1/2
for i in range(1, len(A)-1):
    A[i, i] = 1
A = h_u*A
return A

#vector e creation
def createe(n, varTh, x_0, L):
    e = np.zeros(n)
    for i in range(1, len(e)+2):
        e[i-2] = (-1/varTh)*(E3(varTh*x(i, x_0, n, L)) - E3(varTh*x(i-1, x_0, n,
            L)))
    return e

#matrix E creation
def createE(L, n, varTh):
    h_u = L/n
    E = np.zeros((n,n))
    for i in range(1, n+2):
        for j in range(1, n+2):
            if (i > j) and (j == 2):
                E[i-2, j-2] = (1/(varTh**2))*(-E3(varTh*h_u*(i-1)) +
                    E3(varTh*h_u*(i-2)) + E3(varTh*h_u*(i-1/2)) -
                    E3(varTh*h_u*(i-3/2)))

            elif (i < j) and (j == n+1):
                E[i-2, j-2] = (1/(varTh**2))*(-E3(varTh*(L + h_u*(1/2-i))) +
                    E3(varTh*(L + h_u*(3/2-i))) + E3(varTh*(L-i*h_u)) -
                    E3(varTh*(L-h_u*(i-1))))

            elif (i < j) and (j != 2) and (j != n+1):
                E[i-2, j-2] = (1/(varTh**2))*(E3(varTh*h_u*(j-i+1)) +
                    E3(varTh*h_u*(j-i-1)) - 2*E3(varTh*h_u*(j-i)))

            elif (i > j) and (j != 2) and (j != n+1):
                E[i-2, j-2] = (1/(varTh**2))*(E3(varTh*h_u*(i-j-1)) +

```

```

        E3(varTh*h_u*(i-j+1)) - 2*E3(varTh*h_u*(i-j)))

    elif (i == j) and (j != 2) and (j != n+1):
        E[i-2, j-2] = h_u*2/varTh + 2*(1/(varTh**2))*E3(varTh*h_u) -
            (1/(varTh**2))

E[0, 0] = h_u/varTh + 2*(1/(varTh**2))*E3(varTh*h_u/2) - (1/(varTh**2))
E[-1, -1] = h_u/varTh + 2*(1/(varTh**2))*E3(varTh*h_u/2) - (1/(varTh**2))

return E

#matrix K creation
def createK(L, n, varTh):
    h_u = L/n
    K = np.zeros((n, n))
    for i in range(1, n+2):
        for j in range(1, n+2):
            if (i < j):
                K[i-2, j-2] = (1/(varTh**2))*(E3(varTh*h_u*(j-i+1)) +
                    E3(varTh*h_u*(j-i-1)) - 2*E3(varTh*h_u*(j-i)))

            elif (i == j):
                K[i-2, j-2] = h_u*2/varTh + 2*(1/(varTh**2))*E3(varTh*h_u) -
                    (1/(varTh**2))

            elif (i > j):
                K[i-2, j-2] = (1/(varTh**2))*(E3(varTh*h_u*(i-j-1)) +
                    E3(varTh*h_u*(i-j+1)) - 2*E3(varTh*h_u*(i-j)))

    K = 1/2*K

    return K

#matrix B creation
def createB(n, A, E, s):
    B = np.zeros((n, n))
    B = A - (s/2)*E
    return B

```

```

#vector c creation
def createC1(B, e, I_0):
    c1 = gauss(B, 1/2*e)
    c1 = c1*I_0
    return c1

#matrix S creation (\beta_{j, 1})
def createS(n, K, B):
    S = np.zeros((n, n))
    for i in range(len(K)):
        c = gauss(B, K[:, i])
        S[:, i] = c
    return S

#Jacobian creation for Newton method
def createJac(L, n, kappa, s_0, S, lam, u):
    h_u = L/n
    nablaF = np.zeros((n, n))

    for j in range(len(nablaF)):
        for l in range(len(nablaF[j])):
            nablaF[j, l] = -4*kappa*kappa*s_0*S[j, l]*np.abs((u[l]))**3

    for j in range(len(nablaF)):
        nablaF[j, j] = 2*lam/(h_u**2) + 4*kappa*(s_0 - kappa*s_0*S[j, j])*np.abs((u[j]))**3

    for j in range(len(nablaF)):
        nablaF[j, j-1] = -lam/(h_u**2) - 4*kappa*kappa*s_0*S[j, j-1]*np.abs((u[j-1]))**3

    for j in range(len(nablaF)-1):
        nablaF[j, j+1] = -lam/(h_u**2) - 4*kappa*kappa*s_0*S[j, j+1]*np.abs((u[j+1]))**3

    for j in range(len(nablaF)):
        nablaF[0, j] = -2*kappa*kappa*s_0*S[0, j]*np.abs((u[j]))**3

```



```

nablaF[0, 0] = lam/(h_u**2) + 2*kappa*(s_0 - kappa*s_0*S[0,
    0])*np.abs((u[0]))**3
nablaF[0, 1] = -lam/(h_u**2) - 2*kappa*kappa*s_0*S[0, 1]*np.abs((u[1]))**3

for j in range(len(nablaF)):
    nablaF[-1, j] = -2*kappa*kappa*s_0*S[-1, j]*np.abs((u[j]))**3
nablaF[-1, -2] = -lam/(h_u**2) - 2*kappa*kappa*s_0*S[-1,
    -2]*np.abs((u[-2]))**3
nablaF[-1, -1] = lam/(h_u**2) + 2*kappa*(s_0 - kappa*s_0*S[-1,
    -1])*np.abs((u[-1]))**3

return nablaF

#sum of each row of a system
def sumOfRow(j, L, n, kappa, lam, s_0, S, u, c1):
    h_u = L/n
    z = np.zeros(n)
    if (j == 0):
        for i in range(len(z)):
            z[i] = 1/2*(-kappa*kappa*s_0*S[0, i]*(u[i])**4)
        z[0] = lam/(h_u**2)*u[0] + (1/2)*kappa*s_0*(u[0])**4 -
            (1/2)*kappa*kappa*s_0*S[0, 0]*(u[0])**4
        z[1] = -lam/(h_u**2)*u[1] - (1/2)*kappa*kappa*s_0*S[0, 1]*(u[1])**4

    elif (j == n-1):
        for i in range(len(z)):
            z[i] = 1/2*(-kappa*kappa*s_0*S[-1, i]*(u[i])**4)
        z[-2] = -lam/(h_u**2)*u[-2] - (1/2)*kappa*kappa*s_0*S[-1, -2]*(u[-2])**4
        z[-1] = lam/(h_u**2)*u[-1] + (1/2)*kappa*s_0*(u[-1])**4 -
            (1/2)*kappa*kappa*s_0*S[-1, -1]*(u[-1])**4

    else:
        for i in range(len(z)):
            z[i] = -kappa*kappa*s_0*S[j, i]*(u[i])**4
        z[j] = 2*lam/(h_u**2)*u[j] + kappa*s_0*(u[j])**4 - kappa*kappa*s_0*S[j,
            j]*(u[j])**4
        z[j+1] = -lam/(h_u**2)*u[j+1] - kappa*kappa*s_0*S[j, j+1]*(u[j+1])**4
        z[j-1] = -lam/(h_u**2)*u[j-1] - kappa*kappa*s_0*S[j, j-1]*(u[j-1])**4

```

```

    if (j == 0):
        q = sum(z) - (1/2)*kappa*c1[0]

    if (j == n-1):
        q = sum(z) - (1/2)*kappa*c1[-1]

    else:
        q = sum(z) - kappa*c1[j]

    return q

#vector for Newton method
def createF(L, n, kappa, lam, s_0, S, u, c1):
    F = np.zeros(n)
    for j in range(len(F)):
        F[j] = sumOfRow(j, L, n, kappa, lam, s_0, S, u, c1)
    return F

#grid for future graphs
def grid(L, n):
    h_u = L/n
    a_u = np.zeros(n+1)
    for i in range(len(a_u)):
        a_u[i] = a_u[i] + i*h_u
    return a_u

def sumI(j, n, kappa, s_0, S, u, c1):
    z = np.zeros(n)
    for i in range(len(z)):
        z[i] = kappa*s_0*S[j, i]*(u[i])**4
    return sum(z) + c1[j]

#radiation
def createI(n, kappa, s_0, S, u, c1):
    I = np.zeros(n)
    for i in range(len(I)):
        I[i] = sumI(i, n, kappa, s_0, S, u, c1)

```

```

    return I

#Newton method
def solver(L, n, kappa, s_0, s, lam, I_0):
    varTh = kappa + s
    u = np.ones(n)
    u = 280*u
    A = createA(L, n)
    e = createe(n, varTh, x_0, L)
    E = createE(L, n, varTh)
    K = createK(L, n, varTh)
    B = createB(n, A, E, s)
    c1 = createC1(B, e, I_0)
    S = createS(n, K, B)
    eps = 10**(-8)
    while True:
        nablaF = createJac(L, n, kappa, s_0, S, lam, u)
        F = createF(L, n, kappa, lam, s_0, S, u, c1)
        delta_u = gauss(nablaF, -F)
        print(np.linalg.norm(delta_u, 2))
        u = u + delta_u
        if (np.linalg.norm(delta_u, 2) < eps):
            break
    I = createI(n, kappa, s_0, S, u1, c1)
    return u, I

#different graphs
def graph(L, u1, u2, u3, u4, u5, u6, I1, I2, I3, I4, I5, I6, k):
    a = grid(L, n)

    fig, axs = plt.subplots(2, 1, figsize = (7, 10))
    axs[0].grid(True)
    axs[1].grid(True)
    axs[0].set_xlabel('X')
    axs[0].set_ylabel('u')

    axs[1].set_xlabel('X')
    axs[1].set_ylabel('I')

```

```

col = ['purple', 'cyan', 'fuchsia', 'blue', 'red', 'coral']

if k == 1:
    for i in range(2):
        axs[i].plot([], [], color=col[0], label = '\u03F0=0.2, s=0.15')
        axs[i].plot([], [], color=col[1], label = '\u03F0=0.56, s=0.15')
        axs[i].plot([], [], color=col[2], label = '\u03F0=0.92, s=0.15')
        axs[i].plot([], [], color=col[3], label = '\u03F0=1.28, s=0.15')
        axs[i].plot([], [], color=col[4], label = '\u03F0=1.64, s=0.15')
        axs[i].plot([], [], color=col[5], label = '\u03F0=2, s=0.15')
if k == 2:
    for i in range(2):
        axs[i].plot([], [], color=col[0], label = '\u03F0=0.3, s=0.1')
        axs[i].plot([], [], color=col[1], label = '\u03F0=0.3, s=1.08')
        axs[i].plot([], [], color=col[2], label = '\u03F0=0.3, s=2.06')
        axs[i].plot([], [], color=col[3], label = '\u03F0=0.3, s=3.04')
        axs[i].plot([], [], color=col[4], label = '\u03F0=0.3, s=4.02')
        axs[i].plot([], [], color=col[5], label = '\u03F0=0.3, s=5')

if k == 3:
    for i in range(2):
        axs[i].plot([], [], color=col[0], label = 'I_0 = 500')
        axs[i].plot([], [], color=col[1], label = 'I_0 = 600')
        axs[i].plot([], [], color=col[2], label = 'I_0 = 700')
        axs[i].plot([], [], color=col[3], label = 'I_0 = 800')
        axs[i].plot([], [], color=col[4], label = 'I_0 = 900')
        axs[i].plot([], [], color=col[5], label = 'I_0 = 1000')

if k == 4:
    for i in range(2):
        axs[i].plot([], [], color=col[0], label = '\u03bb = 10')
        axs[i].plot([], [], color=col[1], label = '\u03bb = 30')
        axs[i].plot([], [], color=col[2], label = '\u03bb = 50')
        axs[i].plot([], [], color=col[3], label = '\u03bb = 70')
        axs[i].plot([], [], color=col[4], label = '\u03bb = 80')
        axs[i].plot([], [], color=col[5], label = '\u03bb = 100')

if k == 5:

```

```

for i in range(2):
    axs[i].plot([], [], color=col[0], label = '\u03bb = 0')
    axs[i].plot([], [], color=col[1], label = '\u03bb = 0.1')
    axs[i].plot([], [], color=col[2], label = '\u03bb = 0.05')
    axs[i].plot([], [], color=col[3], label = '\u03bb = 0.01')
    axs[i].plot([], [], color=col[4], label = '\u03bb = 0.005')
    axs[i].plot([], [], color=col[5], label = '\u03bb = 0.001')

for i in range(len(u1)):
    axs[0].plot([a[i], a[i+1]], [u1[i], u1[i]], color = col[0])
    axs[0].plot([a[i], a[i+1]], [u2[i], u2[i]], color = col[1])
    axs[0].plot([a[i], a[i+1]], [u3[i], u3[i]], color = col[2])
    axs[0].plot([a[i], a[i+1]], [u4[i], u4[i]], color = col[3])
    axs[0].plot([a[i], a[i+1]], [u5[i], u5[i]], color = col[4])
    axs[0].plot([a[i], a[i+1]], [u6[i], u6[i]], color = col[5])

for i in range(1, len(u1)-2):
    axs[1].plot([a[i], a[i+1]], [I1[i], I1[i]], color = col[0])
    axs[1].plot([a[i], a[i+1]], [I2[i], I2[i]], color = col[1])
    axs[1].plot([a[i], a[i+1]], [I3[i], I3[i]], color = col[2])
    axs[1].plot([a[i], a[i+1]], [I4[i], I4[i]], color = col[3])
    axs[1].plot([a[i], a[i+1]], [I5[i], I5[i]], color = col[4])
    axs[1].plot([a[i], a[i+1]], [I6[i], I6[i]], color = col[5])
axs[0].legend()
axs[1].legend()

plt.show()

```

Библиографический список

1. Оцисик М. Н. Сложный теплообмен. - Москва.: Мир, 1976. - 615 с.
2. Зигель Р. Теплообмен излучением / Зигель Р., Хауэлл ДЖ. - Москва.: Мир, 1975. - 935 с.
3. Абрамовиц М. Справочник по специальным функциям с формулами графиками и математическими таблицами / Абрамовиц М, Стиган И. - М.: Наука, 1979 - 832 с.
4. Амосов А. А. Вычислительные методы для инженеров: учебное пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. - СПб.: Лань, 2014. - 672 с.
5. Юссеф Я. Э. Некоторые методы проекционного типа численного решения одного класса слабо сингулярных уравнений: дис. ... канд. физ.-мат. наук: 01.01.07. - НИУ «МЭИ», Москва, 2015 - 112 с.

Содержание

1	Постановка задачи	7
1.1	Физическая постановка	7
1.2	Математическая постановка	8
2	Построение разностной аппроксимации	9
2.1	Сведение интегро-дифференциального уравнения к интегральному уравнению	9
2.2	Решение интегрального уравнения переноса излучения	11
2.3	Решение уравнения теплопроводности	19
2.4	Метод Ньютона	21
2.5	Форма матрицы Якоби	23
3	Результаты вычислительных экспериментов	24
3.1	Слой воды толщиной 1 м	24
3.2	Слой воды толщиной 4 м	27
3.3	Различные экстримальные ситуации для слоя воды толщиной 1 м	29
3.4	Другие вещества	32
	Приложение 1. Интегро-экспоненциальная функция	35
	Приложение 2. Программный код	36
	Библиографический список	46