

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт

Факультет

Кафедра

вычислительной техники
(наименование)

Высшая и прикладная математика
(наименование)

Направление подготовки 01.03.04 «Прикладная математика»
(код и наименование)

Профиль Математическое моделирование в экономике и технике
(код и наименование)

БАКАЛАВРСКАЯ РАБОТА

на тему

РАЗНОСТНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ТЕРМОУПРУГОСТИ

Студент 17ВБ1
(группа)

Нф 17.06.2021 Новоженова А. А.
(подпись, дата) (фамилия, инициалы)

Руководитель к.ф.-м.н., доцент

Куд 17.06.21 Кудряшова Н. Ю.
(подпись, дата) (фамилия, инициалы)

Нормоконтролер к.т.н., доцент

Тар 17.06.2021 Тарасов Д. В.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от 04.06.21г. № 10)

Заведующий кафедрой д.ф.-м.н., проф.

Бой 18.06.21 Бойков И. В.
(подпись, дата) (фамилия, инициалы)

Работа защищена с отметкой хор. (протокол заседания ГЭК от 22.06.21г. № 11)

Секретарь ГЭК к.пед.н., доцент

Звер 22.06.21г. Зверовщикова Н. В.
(подпись, дата) (фамилия, инициалы)

Пенза 2021

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Политехнический институт

Факультет

Вычислительной техники
(наименование)

Кафедра

Высшая и прикладная математика
(наименование)

«УТВЕРЖДАЮ»

Заведующий кафедрой

 И. В. Бойков
(подпись, инициалы, фамилия)

«30» ноября 2020 г.

ЗАДАНИЕ

на бакалаврскую работу

1. Студент Новоженова Анастасия Геннадьевна гр. 17ВБ1
(фамилия, имя, отчество)

факультета вычислительной техники

2. Тема работы Разностные методы решения задач термоупругости

Тема утверждена приказом по ПГУ № 0903 от « 30 » ноября 2020 г.

3. Руководитель работы к.ф.-м.н., доцент Кудряшова Н. Ю.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

1. Построить и обосновать разностные схемы для решения задач термоупругости

2. Установить порядок аппроксимации и условия устойчивости разностных схем

3. Построить алгоритм решения стационарных и нестационарных задач термоупругости

4. Разработать программную реализацию построенных разностных схем

5. С помощью разработанных программ решить модельные задачи, произвести сравнение полученных результатов, сделать выводы

5. Перечень подлежащих разработке вопросов

1. Обзор литературы. Постановка задачи

2. Разработка и обоснование вычислительных схем

3. Построение алгоритмов решения стационарных и нестационарных задач термоупругости

4. Решение модельных примеров

6. Календарный график по выполнению проекта

Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1. Обзор литературы. Постановка задачи	20 %	28.05.2021	
2. Построение алгоритмов решения стационарных и нестационарных задач термоупругости	25 %	02.06.2021	
3. Разработка численных алгоритмов решения	20 %	07.06.2021	
4. Разработка программы	25 %	10.06.2021	
5. Оформление пояснительной записки	10 %	14.06.2021	

Дата выдачи задания «30» ноября 2020 г.

Руководитель бакалаврской работы  Н. Ю. Кудряшова
(подпись, инициалы, фамилия)

Задание к исполнению принял студент  А. Г. Новоженова
(подпись, инициалы, фамилия)

Работу к защите допустить
Декан факультета  Л. Р. Фионова
(подпись, инициалы, фамилия)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. РЕШЕНИЕ СТАЦИОНАРНЫХ ЗАДАЧ ТЕРМОУПРУГОСТИ.....	6
1.1. Основные понятия и определения термоупругости	6
1.1.1. Разностные уравнения	6
1.1.2. Разностная аппроксимация простейших дифференциальных операторов	7
1.1.3. Устойчивые и неустойчивые разностные схемы.....	7
1.1.4. Аппроксимация	8
1.1.5. Устойчивость.....	8
1.1.6. Сходимость	9
1.2. Постановка задачи.....	9
1.3. Разностная схема	11
1.4. Порядок аппроксимации, условия сходимости	14
1.5. Способ реализации.....	15
2. РЕШЕНИЕ НЕСТАЦИОНАРНЫХ ЗАДАЧ ТЕРМОУПРУГОСТИ	17
2.1. Постановка задачи.....	17
2.3. Решение многомерных уравнений	19
2.3.1. Разностные схемы расщепления	19
2.3.2. Экономичные разностные схемы	23
2.3.3. Схема переменных направлений.....	28
2.4. Порядок аппроксимации, условия сходимости.	30
2.5. Способ реализации.....	31
3. ПРАКТИЧЕСКАЯ ЧАСТЬ	35
3.1. Модельная задача.....	35
3.2. Модельная задача.....	35
3.3. Анализ полученных результатов	36
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	40

ПРИЛОЖЕНИЕ А. Листинг программы для решения стационарного уравнения термоупругости при помощи разностной схемы	41
ПРИЛОЖЕНИЕ Б. Листинг программы для решения нестационарного уравнения термоупругости при помощи разностной схемы	46
ПРИЛОЖЕНИЕ В. Графические результаты зависимости погрешности от количества итераций для задач термоупругости	51

ВВЕДЕНИЕ

В настоящее время теория термоупругости достигла значимых успехов в развитии, в связи с серьёзными проблемами, возникающими при разработке конструкций, если элементы работают в нерегулярных и необычных условиях нагрева. При этом изменяются физико-механические свойства материалов и возникают градиенты температуры с неравномерным тепловым увеличением компонентов.

Стимулом к изучению термической упругости стали проблемы термоупругости конструктивных элементов. Исследования проводились на основе теории, развитой Дюамелем и Нейманом, которые вытекали из следующей гипотезы: Полная деформация – это сумма упругих деформаций, сопровождаемых давлением с физиологическими отношениями и чистым тепловым расширением, где температурное поле известно из классической теории теплопроводности.

Согласно основным положениям теории Дюамеля-Неймана, было показано, что нестационарные механические и тепловые устройства ограничены: они не точно описывают движение тела. Динамические эффекты возникают в конструкциях при определенных нестатистических тепловых условиях.

В целом изменения температуры тела происходят не только в результате поступления тепла от внешних источников, но и в результате самого процесса деформации. Когда тело деформируется в результате высокоскоростных механических или тепловых воздействий, так называемый эффект связи вызывается взаимодействием полей деформации и температуры.

Теория термоупругости частно накладывает ограничения на размер термического возмущения: повышение температуры относительно невелико по сравнению с начальной температурой. Это устранение ограничения не нарушает предположение о малой деформации, но приводит к нелинейным выражениям термоупругости. Развитие общей теории термоупругости возможно без определенных ограничений теории малых деформаций с учетом температурной

характеристики упругости и теплоты. Обычно это нелинейная теория комбинированной термоупругости, которая в особых случаях включает, как линейную теорию термоупругости в сочетании с небольшим тепловым возмущением, так и независимую термоупругость в сочетании с большим тепловым возмущением, использующую уравнения линейного движения и нелинейное уравнение теплопроводности.

При исследовании задач динамической термоупругости с учетом взаимосвязи полей деформации и температуры становится возможным выявление новых качественных характеристик процесса деформации. Анализ относительно простого решения задачи об одновременном распределении гармонических термоупругих волн в бесконечном теле позволяет точно понять основные свойства термоупругих явлений на основе различных частот волн и параметров связи материалов. Двумерные задачи о распространении упругих тепловых волн по поверхности длинного цилиндра следует упомянуть, как проблемы с основными ограничениями теплоизоляции.

Решение задач термоупругости с телами конечных размеров создает серьезные математические трудности. Поэтому большой интерес представляют колебательные принципы комбинированной термоупругости, особенно принцип биоколебаний, позволяющих разработать приближенные методы решения комбинированных задач упругой теории динамической теплопроводности.

Всё вышесказанное доказывает важность и ценность проблемы термоупругости и изучения ее моделей.

В данной выпускной квалифицированной работе исследуются и реализуются разностные методы решения стационарных и нестационарных уравнений, в частных производных, к которым приводят задачи термоупругости.

1. РЕШЕНИЕ СТАЦИОНАРНЫХ ЗАДАЧ ТЕРМОУПРУГОСТИ

1.1. Основные понятия и определения термоупругости

1.1.1. Разностные уравнения

Решение дифференциальных уравнений приближенными методами приводит к системам линейных алгебраических уравнений [1]

$$Au = f,$$

где $A = (a_{ij})$ – квадратная матрица порядка N , $u = (u_1, u_2, \dots, u_N)$ – искомый вектор, $f = (f_1, f_2, \dots, f_N)$ – заданный вектор.

Существует два метода решения СЛАУ:

- а) прямые методы,
- б) итерационные методы.

Рассмотрим разностные уравнения, для которых матрица имеет специальный вид. Отметить, что на практике чаще всего встречаются системы разностных уравнений очень высокого порядка.

Разностные уравнения появляются при аппроксимации дифференциальных уравнений математической физики. При этом приходится искать функции двух или трех переменных, заданных на сетке, т. е. на дискретном множестве точек, число которых достигает десятков и даже сотен тысяч. Для определения сеточных функций получаются системы линейных алгебраических уравнений (разностных уравнений), для которых характерны два обстоятельства:

- 1) матрица A имеет специальный вид (имеет много нулевых элементов),
- 2) число уравнений очень велико ($10^4 - 10^5$).

В этой работе мы проведем изучение разностных уравнений термоупругости. В частности, будут рассмотрены методы решения стационарных и нестационарных задач термоупругости.

1.1.2. Разностная аппроксимация простейших дифференциальных операторов

Для того чтобы написать разностную схему, описывающую дифференциальное уравнение, нужно сделать два шага.

1. Необходимо заменить область непрерывного изменения аргумента областью дискретного его изменения.

2. Необходимо заменить дифференциальный оператор некоторым разностным оператором, а также сформулировать разностный аналог для краевых условий и для начальных данных.

После таких процедур мы приходим к алгебраической системе уравнений. Таким образом, задача о численном решении линейного дифференциального уравнения сводится к решению алгебраической системы.

При численном решении той или иной математической задачи мы не можем воспроизвести разностное решение для всех значений аргумента, изменяющегося внутри некоторой области евклидова пространства.

Естественно поэтому выбрать в этой области некоторое конечное множество точек и приближенное решение искать только в этих точках. Такое множество точек называется сеткой. Отдельные точки называют узлами сетки.

Функция, определенная в узлах сетки, называется сеточной функцией. Таким образом, мы заменили область непрерывного изменения аргумента сеткой, т. е. областью дискретного изменения аргумента; иными словами, мы осуществили аппроксимацию пространства решений дифференциального уравнения пространством сеточных функций.

Свойства разностного решения и, в частности, его близость к точному решению зависят от выбора сетки.

1.1.3. Устойчивые и неустойчивые разностные схемы

Использование разностных схем позволяет свести решение задачи для дифференциального уравнения к решению системы линейных алгебраических уравнений. При этом правые части уравнений, краевые и начальные данные за-

даются с определенной погрешностью. В процессе самого численного решения системы также возможны ошибки, связанные с округлением. Естественно потребовать от разностной схемы, чтобы малые ошибки, допущенные во входных данных, не нарастали в процессе вычислений и не приводили к искажению решения.

Схемы, которые в процессе счета усиливают начальные погрешности, называются неустойчивыми и не могут быть использованы на практике.

1.1.4. Аппроксимация

Пусть дифференциальный оператор $L(u)$, определен на функциях u , заданных в области $D \subset R^N$, аппроксимируется на некотором классе функций $u \in U$ конечно-разностным оператором $R_h(u_h)$, определенным на функциях u_h заданных на сетки с шагом h , если выполняются условие сходимости[2]

$$\|L(u) - R_h(u_h)\| \rightarrow 0, \quad h \rightarrow 0 \quad (\forall u \in U).$$

Аппроксимация имеет порядок точности k , если

$$\|L(u) - R_h(u_h)\| \leq h^k M, \quad h \rightarrow 0 \quad (\forall u \in U),$$

где $M = \text{const}$, которая зависит от функции $u \in U$, но не зависит от шага h . Аппроксимация зависит от выбора нормы. Достаточно часто используется дискретный аналог нормы равномерной непрерывности:

$$\|u_h\| = \max_n |u_h(x_n)|,$$

иногда используются дискретные аналоги интегральных норм.

1.1.5. Устойчивость

Недостаточно будет условий аппроксимации для того, чтобы результат разностной схемы приближался к точному ответу при $h \rightarrow 0$. Если в схеме коэффициенты не зависят от решения дифференциального уравнения, то необходимо выполнение условия устойчивости. Данные схемы можно представить, как линейный оператор, который преобразует значение функции в момент t в значения функции в момент $t+h$. Условия устойчивости требует, чтобы собствен-

ные числа оператора не превосходили по модулю $1+ch$, где $c=const$, при $h \rightarrow 0$. Погрешности схемы быстро возрастают и результаты становятся хуже, если это условие не выполняется.

1.1.6. Сходимость

Под сходимостью численного решения понимают его сходимость к точному решению при уменьшении шага сетки h .

$$\lim_{h \rightarrow 0^+} \|\tilde{y}_h(t) - y(t_n)\| = 0.$$

Если выполнены условия аппроксимации и условия устойчивости, то результат разностной схемы сходится к решению дифференциального уравнения.

1.2. Постановка задачи

Рассмотрим напряженное состояние твердого упругого изотропного тела с прямоугольным сечением Ω . Для перемещения $v = (v_1(x, y); v_2(x, y))$.

Выполнены уравнения Ламе[1]:

$$\mu \Delta v + (\lambda + \mu) \text{grad div } v - \gamma \text{grad } u + f = 0, x \in \Omega \quad (1.2.1)$$

где λ и μ – постоянные Ламе, характеризующие упругие свойства среды, $\gamma = (3\lambda + 2\mu)\alpha$, α – коэффициент линейного расширения, f – вектор объемных сил. В уравнении (1.2.1) Δ – оператор Лапласа, т.е.

$$\Delta v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2},$$

Стационарное тепловое поле однородного тела описывается уравнением

$$\Delta u = 0, (x, y) \in \Omega \quad (1.2.2)$$

Дополним уравнения (1.2.1) и (1.2.2) граничными условиями

$$v(x, y) = q(x, y) \quad (1.2.3)$$

$$u(x, y) = g(x, y) \quad (1.2.4)$$

$$(x, y) \in \partial\Omega$$

Получим систему уравнений:

$$(\lambda + \mu) \frac{\partial^2 v_1}{\partial x^2} + \mu \frac{\partial^2 v_1}{\partial y^2} + (\lambda + \mu) \frac{\partial^2 v_2}{\partial x \partial y} - \gamma \frac{\partial u}{\partial x} + f_1 = 0, \quad (1.2.5)$$

$$\mu \frac{\partial^2 v_2}{\partial x^2} + (\lambda + 2\mu) \frac{\partial^2 v_2}{\partial y^2} + (\lambda + \mu) \frac{\partial^2 v_1}{\partial x \partial y} - \gamma \frac{\partial u}{\partial y} + f_2 = 0, \quad (1.2.6)$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0. \quad (1.2.7)$$

Система (1.2.5) и (1.2.6), характеризуется тем, что неизвестные $v_\alpha(x)$, $\alpha=1, 2$ входят в оба уравнения.

Определим вектор-функции w , равный нулю на $\partial\Omega$, оператор

$$Lw = -\mu\Delta w - (\lambda + \mu) \text{grad div } w, x \in \Omega. \quad (1.2.8)$$

Пусть $H_2 = H \oplus H$, $H = L_2(\Omega)$, тогда непосредственными выкладками убеждаемся, что

$$(Lw, v) = (w, Lv),$$

т.е. оператор L самосопряжен. Имеем[1]

$$(1.2.9) (Lw, v) = \int_{\Omega} \left[2\mu \left(\left(\frac{\partial \omega_1}{\partial x} \right)^2 + \left(\frac{\partial \omega_2}{\partial y} \right)^2 \right) + \lambda \left(\frac{\partial \omega_1}{\partial x} + \frac{\partial \omega_2}{\partial y} \right)^2 \right] dx,$$

поэтому $L \geq 0$ в силу того, что λ и μ положительны.

Покажем справедливость оценки

$$c_1(-Vw, w) \leq (Lw, w) \leq c_2(-Vw, w). \quad (1.2.10)$$

Учитывая, что

$$(-Vw, w) = \sum_{a=1}^2 \int_{\Omega} \left[\left(\frac{\partial \omega_a}{\partial x} \right)^2 + \left(\frac{\partial \omega_a}{\partial y} \right)^2 \right] dx,$$

Имеет место неравенство[1]

$$\int_{\Omega} \frac{\partial \omega_1}{\partial y} \frac{\partial \omega_2}{\partial x} dx = \int_{\Omega} \frac{\partial \omega_1}{\partial x} \frac{\partial \omega_2}{\partial y} dx,$$

из уравнения (1.2.9) получаем[1]

$$(Lw, v) = \sum_{a=1}^2 \mu \int_{\Omega} \left[\left(\frac{\partial \omega_a}{\partial x} \right)^2 + \left(\frac{\partial \omega_a}{\partial y} \right)^2 \right] dx + \\ + (\lambda + \mu) \int_{\Omega} \left(\frac{\partial \omega_1}{\partial x} + \frac{\partial \omega_2}{\partial y} \right)^2 dx \geq \mu (-Vw, w),$$

В уравнении (1.2.10) $c_1 = \mu$.

Аналогично[1]

$$(Lw, v) = \sum_{a=1}^2 \mu \int_{\Omega} \left[\left(\frac{\partial \omega_a}{\partial x} \right)^2 + \left(\frac{\partial \omega_a}{\partial y} \right)^2 \right] dx + (\lambda + \mu) \int_{\Omega} \left(\frac{\partial \omega_1}{\partial x} + \frac{\partial \omega_2}{\partial y} \right)^2 dx \leq \\ \leq \sum_{a=1}^2 \mu \int_{\Omega} \left[\left(\frac{\partial \omega_a}{\partial x} \right)^2 + \left(\frac{\partial \omega_a}{\partial y} \right)^2 \right] dx + (\lambda + \mu) \int_{\Omega} \left(\frac{\partial \omega_1}{\partial x} + \frac{\partial \omega_2}{\partial y} \right)^2 dx + \\ + (\lambda + \mu) \int_{\Omega} \left(\frac{\partial \omega_2}{\partial x} + \frac{\partial \omega_1}{\partial y} \right)^2 dx \leq (\lambda + \mu) \sum_{a=1}^2 \int_{\Omega} \left[\left(\frac{\partial \omega_a}{\partial x} \right)^2 + \left(\frac{\partial \omega_a}{\partial y} \right)^2 \right] dx,$$

поэтому $c_2 = \lambda + 2\mu$ в неравенстве (1.2.10).

Для дискретных аналогов оценки типа (1.2.10) позволяют строить регуляризованные итерационные методы на основе разностного оператора Лапласа.

1.3. Разностная схема

Для того чтобы построить разностные схемы для (1.2.1)-(1.2.4) будем исходить из уравнений (1.2.5),(1.2.6). В прямоугольнике Ω вводится равномерная разностная сетка по каждому направлению ω . В результате получаются узлы (x_i, y_j) , которые образуют шаблон (рис.1)

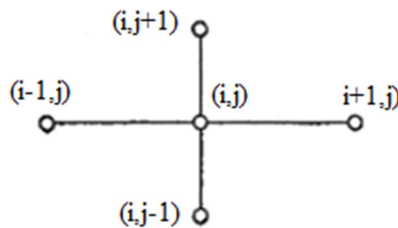


Рисунок 1 – Шаблон «крест»

Эта сетка имеет шаги h и l по направлениям x и y

$$\begin{aligned}x_i &= ih, i = 0, \dots, M, & y_j &= jl, j = 0, \dots, N, \\x_M &= Mh = a, & y_N &= Nl = b.\end{aligned}$$

Используем аппроксимацию второго порядка для смешанных производных в (1.2.5), (1.2.6)

Заменим производные разностными отношениями:

$$\begin{aligned}\left. \frac{\partial^2 v_1}{\partial x^2} \right|_{(i,j)} &= \frac{v_{1i+1,j} - 2v_{1i,j} + v_{1i-1,j}}{h^2} + O(h^2), \\ \left. \frac{\partial^2 v_1}{\partial y^2} \right|_{(i,j)} &= \frac{v_{1i,j+1} - 2v_{1i,j} + v_{1i,j-1}}{l^2} + O(l^2), \\ \left. \frac{\partial^2 v_2}{\partial x \partial y} \right|_{(i,j)} &= \frac{v_{2i+1,j+1} + 2v_{i,j+1} - v_{2i+1,j} - v_{2i,j}}{4hl} + O(h^2 + l^2), \\ \left. \frac{\partial u}{\partial x} \right|_{(i,j)} &= \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2), \\ \left. \frac{\partial^2 v_2}{\partial x^2} \right|_{(i,j)} &= \frac{v_{2i+1,j} - 2v_{2i,j} + v_{2i-1,j}}{h^2} + O(h^2), \\ \left. \frac{\partial^2 v_2}{\partial y^2} \right|_{(i,j)} &= \frac{v_{2i,j+1} - 2v_{2i,j} + v_{2i,j-1}}{l^2} + O(l^2), \\ \left. \frac{\partial^2 v_1}{\partial x \partial y} \right|_{(i,j)} &= \frac{v_{1i+1,j+1} + v_{1i,j+1} - v_{1i+1,j} - v_{1i,j}}{4hl} + O(h^2 + l^2), \\ \left. \frac{\partial u}{\partial y} \right|_{(i,j)} &= \frac{u_{i,j+1} - u_{i,j-1}}{2l} + O(l^2)\end{aligned}$$

Построим разностную схему для уравнений (1.2.5) и (1.2.6).

$$\begin{aligned}(\lambda + 2\mu) \frac{v_{1i+1,j} - 2v_{1i,j} + v_{1i-1,j}}{h^2} + \mu \frac{v_{1i,j+1} - 2v_{1i,j} + v_{1i,j-1}}{l^2} + \\ + (\lambda + 2\mu) \frac{v_{2i+1,j+1} + 2v_{2i,j+1} - v_{2i+1,j} - v_{2i,j}}{4hl} - \gamma \frac{u_{i+1,j} - u_{i-1,j}}{2h} + f_{i,j} = 0\end{aligned} \quad (1.3.1)$$

$$\begin{aligned} & \mu \frac{v_{2i+1,j} - 2v_{2i,j} + v_{2i-1,j}}{h^2} + (\lambda + \mu) \frac{v_{2i,j+1} - 2v_{2i,j} + v_{2i,j-1}}{l^2} + \\ & + (\lambda + 2\mu) \frac{v_{1i+1,j+1} + v_{1i,j+1} - v_{1i+1,j} - v_{1i,j}}{4hl} - \gamma \frac{u_{i,j+1} - u_{i,j-1}}{2l} + f_{2i,j} = 0 \end{aligned} \quad (1.3.2)$$

Для уравнения Лапласа (1.2.2) введем замену производных разностными отношениями:

$$\begin{aligned} \left. \frac{\partial^2 u}{\partial x^2} \right|_{(i,j)} &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2), \\ \left. \frac{\partial^2 u}{\partial y^2} \right|_{(i,j)} &= \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{l^2} + O(l^2). \end{aligned}$$

получим

$$\begin{aligned} \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{l^2} &= 0 \\ i = \overline{1, N-1}, \quad j = \overline{1, M-1}. \end{aligned} \quad (1.3.3)$$

Простейшим образом аппроксимируем граничные условия (1.2.3) и (1.2.4)[3]:

$$v(x_i, y_j) = q(x_i, y_j), (x_i, y_j) \in \partial\Omega, \quad (1.3.4)$$

$$u(x_i, y_j) = g(x_i, y_j), (x_i, y_j) \in \partial\Omega. \quad (1.3.5)$$

Вместо разностной задачи (1.3.2)-(1.3.5) будем рассматривать:

$$(\wedge v + Gu) = F, \quad x \in w, \quad (1.3.6)$$

$$\overset{\circ}{A}u = \varphi, \quad x \in w. \quad (1.3.7)$$

на множестве сеточных функций, обращающихся в нуль на $\partial\omega$. Согласно с (1.3.1) и (1.3.2) имеем[4]

$$Gu = \left\{ \gamma \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \gamma \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{l^2} \right\} \quad (1.3.8)$$

а $\overset{\circ}{A}$ - разностный оператор Лапласа.

1.4. Порядок аппроксимации, условия сходимости

Рассмотрим детально свойства разностного оператора Λ . Пусть скалярное произведение $H_2 = H \oplus H$ в H определяется:[2]

$$(u, v) = \sum_{x \in \omega} u(x)v(x)hl.$$

При условии (1.3.1) и (1.3.2) получим:

$$\begin{aligned} -(\Lambda v)_1 &= (\lambda + 2\mu) \frac{v_{1i+1,j} - 2v_{1i,j} + v_{1i-1,j}}{h^2} + \mu \frac{v_{1i,j+1} - 2v_{1i,j} + v_{1i,j-1}}{l^2} + \\ &+ (\lambda + 2\mu) \frac{v_{2i+1,j+1} + 2v_{2i,j+1} - v_{2i+1,j} - v_{2i,j}}{4hl} \end{aligned} \quad (1.4.1)$$

$$\begin{aligned} -(\Lambda v)_2 &= \mu \frac{v_{2i+1,j} - 2v_{2i,j} + v_{2i-1,j}}{h^2} + (\lambda + \mu) \frac{v_{2i,j+1} - 2v_{2i,j} + v_{2i,j-1}}{l^2} + \\ &+ (\lambda + \mu) \frac{v_{1i+1,j+1} + v_{1i,j+1} - v_{1i+1,j} - v_{1i,j}}{4hl} \end{aligned} \quad (1.4.2)$$

Убедимся в самосопряженности оператора Λ :

$$(\Lambda v, w) = (v, \Lambda w). \quad (1.4.3)$$

Получим оценку энергетической эквивалентности из формул (1.4.1)-(1.4.2)

$$\mu(\overset{o}{A}v, v) \leq (\Lambda v, v) \leq (\lambda + 2\mu)(\overset{o}{A}v, v). \quad (1.4.4)$$

Оператор Λ наследует свойства дифференциального оператора упругости L .

Рассмотрим систему (1.3.7), как задачу для погрешности, притом

$$f = O(|h|^2), \quad \varphi = O(|h|^2), \quad |h|^2 = h^2 + l^2 \quad (1.4.5)$$

Умножим скалярно уравнение (1.3.6) на v , а уравнение (1.3.7) на u . Из (1.3.7) получаем

$$\|u\|_1 \leq \|\varphi\|_{-1}, \quad (1.4.6)$$

где обозначим

$$\|u\|_1^2 \equiv (\overset{o}{A}u, u), \quad \|u\|_{-1}^2 \equiv ((\overset{o}{A})^{-1}u, u). \quad (1.4.7)$$

Из уравнения (1.3.6) при условии (1.3.8) и (1.4.6) следует

$$\|v\|_1 \leq M\tau\|\varphi\|_{-1} + \|F\|_{-1} \quad (1.4.8)$$

Из (1.4.6) и (1.4.8) следует, что разностная схема (1.3.1) и (1.3.2) имеет порядок аппроксимации $O(h^2 + l^2)$.

1.5. Способ реализации

Для реализации разностной схемы (1.3.3) применим метод Гаусса - Зейделя.

Перепишем уравнение (1.3.3) в виде:

$$u_{ij} = \frac{1}{2} \left[\frac{l^2}{l^2 + h^2} (u_{i-1,j} + u_{i+1,j}) + \frac{h^2}{l^2 + h^2} (u_{i,j+1} + u_{i,j-1}) \right], \quad (1.5.1)$$

$$u_{i0} = g(x_i, y_0), \quad u_{iN} = g(x_i, y_N), \quad i = 1, 2, \dots, M-1.$$

$$u_{0j} = g(x_0, y_j), \quad u_{Mi} = g(x_M, y_j), \quad j = 1, 2, \dots, N-1.$$

Для (1.5.1) используем разностный шаблон:

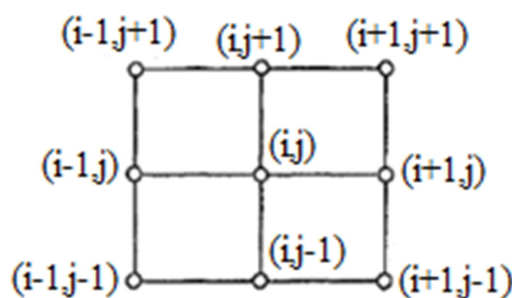


Рисунок 2 – Разностный шаблон

Численное решение уравнения состоит в нахождении приближенных значений u_{ij} . Выберем начальное приближение $u_{ij}^{(0)}$. Последующие приближения будем находить по формулам:

$$u_{ij}^{(s+1)} = \frac{1}{2} \left[\frac{l^2}{l^2 + h^2} (u_{i-1,j}^{(s+1)} + u_{i+1,j}^{(s)}) + \frac{h^2}{l^2 + h^2} (u_{i,j+1}^{(s)} + u_{i,j-1}^{(s+1)}) \right], \quad (1.5.2)$$

где s – номер итерации. При $s \rightarrow \infty$ последовательность $u_{ij}^{(s)}$ сходится к точному решению системы (1.5.1).

В результате определим значения сеточной функции u_{ij} во всех узлах сетки.

Для решения уравнений (1.3.1) и (1.3.2) применим итерационные методы.

Выберем начальные приближения $v_{1ij}^{(0)}$ и $v_{2ij}^{(0)}$. Приближенные значения сеточных функций v_{1ij} и v_{2ij} будем находить по формулам:

$$v_{1ij}^{(k+1)} = v_{1ij}^{(k)} + \gamma \tau_k \frac{u_{i+1,j} - u_{i-1,j}}{2h} - \tau_k f_{1ij} - \tau_k (\lambda + 2\mu) \frac{v_{1i+1,j}^{(k)} - 2v_{1ij}^{(k)} + v_{1i-1,j}^{(k)}}{h^2} - \mu \tau_k \frac{v_{1i,j+1}^{(k)} - 2v_{1ij}^{(k)} + v_{1i,j-1}^{(k)}}{l^2} - \tau_k (\lambda + \mu) \frac{v_{2i+1,j+1}^{(k)} + v_{2i,j+1}^{(k)} - v_{2i+1,j}^{(k)} + v_{2i,j}^{(k)}}{4hl}. \quad (1.5.3)$$

$$v_{2ij}^{(k+1)} = v_{2ij}^{(k)} + \gamma \tau_k \frac{u_{i,j+1} - u_{i,j-1}}{2l} - \tau_k f_{2ij} - \tau_k \mu \frac{v_{2i+1,j}^{(k)} - 2v_{2ij}^{(k)} + v_{2i-1,j}^{(k)}}{h^2} - (\lambda + 2\mu) \tau_k \frac{v_{2i,j+1}^{(k)} - 2v_{2ij}^{(k)} + v_{2i,j-1}^{(k)}}{l^2} - \tau_k (\lambda + \mu) \frac{v_{1i+1,j+1}^{(k)} + v_{1i,j+1}^{(k)} - v_{1i+1,j}^{(k)} + v_{1i,j}^{(k)}}{4hl}. \quad (1.5.4)$$

$$i = \overline{0, N-1}, j = \overline{0, M-1}.$$

где τ_k - некоторый положительный числовой параметр.

2. РЕШЕНИЕ НЕСТАЦИОНАРНЫХ ЗАДАЧ ТЕРМОУПРУГОСТИ

2.1. Постановка задачи

Рассмотрим твердое упругое изотропное тело с прямоугольным сечением Ω . Нестационарные термоупругие деформации описываются уравнением [1]

$$\rho \frac{\partial^2 v}{\partial t^2} - \mu \Delta v - (\lambda + \mu) \text{grad div } v + \gamma \text{grad } u = f, \quad (2.1.1)$$

$$x \in \Omega, \quad 0 < t \leq T.$$

где $\rho = \text{const}$ – плотность. Для моделирования температурного состояния используется нестационарное уравнение теплопроводности, которое запишем в виде [1]

$$\frac{\partial u}{\partial t} - k \Delta u = 0, \quad (x, y) \in \Omega, \quad 0 < t \leq T. \quad (2.1.2)$$

Для системы уравнений термоупругости (2.1.1) и (2.1.2) граничные условия зададим в простейшем виде:

$$v(x, y, 0) = w^0(x, y), \quad (x, y) \in \Omega, \quad (2.1.3)$$

$$\frac{\partial v}{\partial t}(x, 0) = w^1(x), \quad x \in \Omega, \quad (2.1.4)$$

$$u(x, 0) = u^0(x), \quad x \in \Omega. \quad (2.1.5)$$

Запишем уравнение (2.1.1) и (2.1.2) для отдельных компонентов перемещения в следующем виде:

$$\rho \frac{\partial^2 v_1}{\partial t^2} + (\lambda + \mu) \frac{\partial^2 v_1}{\partial x_1^2} - \mu \frac{\partial^2 v_1}{\partial x_2^2} - (\lambda + \mu) \frac{\partial^2 v_2}{\partial x_1 \partial x_2} + \gamma \frac{\partial u}{\partial x_1} = f_1, \quad (2.1.6)$$

$$\rho \frac{\partial^2 v_2}{\partial t^2} - \mu \frac{\partial^2 v_2}{\partial x_1^2} - (\lambda + \mu) \frac{\partial^2 v_2}{\partial x_2^2} - (\lambda + \mu) \frac{\partial^2 v_1}{\partial x_1 \partial x_2} + \gamma \frac{\partial u}{\partial x_2} = f_2, \quad (2.1.7)$$

$$\frac{\partial u}{\partial t} - k \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0 \quad (2.1.8)$$

$$x \in \Omega, \quad 0 < t \leq T.$$

Из гиперболической системы определим вектор перемещения $v = (v_1, v_2)$

2.2. Разностная схема

Для того чтобы построить разностную схему введем сетку узлов времени t с шагом τ [5]:

$$\omega_\tau = \{t_n = n\tau, n = 0, 1, \dots, K-1, K\tau = T\}$$

и пространственную сетку по переменным x и y с шагом h

$$\Omega_h = \{(x_i, y_j), x_i = ih, y_j = jh, i, j = 0, 1, \dots, M, Mh = 1\}$$

Заменим частные производные во внутренних узлах сетки разностными отношениями:

$$\left. \frac{\partial^2 v_1}{\partial t^2} \right|_{(x_i, y_j, t_p)} = \frac{v_{1i,j}^{p+1} - v_{1ij}^{p-1} - 2v_{1i,j}^p}{\tau^2} + O(\tau^2),$$

$$\left. \frac{\partial^2 v_1}{\partial x_1^2} \right|_{(x_i, y_j, t_p)} = \frac{v_{1i+1,j}^p - 2v_{1ij}^p + v_{1i-1,j}^p}{h^2} + O(h^2),$$

$$\left. \frac{\partial^2 v_1}{\partial x_2^2} \right|_{(x_i, y_j, t_p)} = \frac{v_{1i,j+1}^p - 2v_{1ij}^p + v_{1i,j-1}^p}{l^2} + O(l^2),$$

$$\left. \frac{\partial^2 v_2}{\partial x_1 \partial x_2} \right|_{(x_i, y_j, t_p)} = \frac{v_{2i+1,j+1}^p + v_{2i,j+1}^p - v_{2i+1,j}^p + v_{2i,j}^p}{4hl} + O(l^2 + h^2)$$

$$\left. \frac{\partial u}{\partial x_1} \right|_{(x_i, y_j, t_p)} = \frac{u_{i+1,j}^p - u_{i-1,j}^p}{2h} + O(h^2).$$

$$\left. \frac{\partial^2 v_2}{\partial t^2} \right|_{(x_i, y_j, t_p)} = \frac{v_{2i,j}^{p+1} - v_{2ij}^{p-1} - 2v_{2i,j}^p}{\tau^2} + O(\tau^2),$$

$$\left. \frac{\partial^2 v_2}{\partial x_1^2} \right|_{(x_i, y_j, t_p)} = \frac{v_{2i+1,j}^p - 2v_{2ij}^p + v_{2i-1,j}^p}{h^2} + O(h^2),$$

$$\left. \frac{\partial^2 v_2}{\partial x_2^2} \right|_{(x_i, y_j, t_p)} = \frac{v_{2i,j+1}^p - 2v_{2ij}^p + v_{2,j+1}^p}{l^2} + O(l^2),$$

$$\left. \frac{\partial^2 v_1}{\partial x_1 \partial x_2} \right|_{(x_i, y_j, t_p)} = \frac{v_{1i+1,j+1}^p + v_{1i,j+1}^p - v_{1i+1,j}^p + v_{1i,j}^p}{4hl} + O(h^2 + l^2),$$

$$\left. \frac{\partial u}{\partial x_2} \right|_{(x_i, y_j, t_p)} = \frac{u_{i,j+1}^p - u_{i,j-1}^p}{2l} + O(l^2).$$

Составим для уравнений (2.1.6) - (2.1.8) разностные схемы

$$\begin{aligned} & p \frac{v_{1i,j}^{p+1} - v_{1ij}^{p-1} - 2v_{1i,j}^p}{\tau^2} - (\lambda + 2\mu) \frac{v_{1i+1,j}^p - 2v_{1ij}^p + 2v_{1i-1,j}^p}{h^2} - \\ & - \mu \frac{v_{1i,j+1}^p - 2v_{1ij}^p + v_{1,j+1}^p}{l^2} - (\lambda + \mu) \frac{v_{2i+1,j+1}^p + v_{2i,j+1}^p - v_{2i+1,j}^p + v_{2i,j}^p}{4hl} = \\ & = \gamma \frac{u_{i+1,j}^p - u_{i-1,j}^p}{2h} + f_1(x_i, y_j, t_p), \end{aligned} \quad (2.2.1)$$

$$\begin{aligned} & p \frac{v_{2i,j}^{p+1} - v_{2ij}^{p-1} - 2v_{2i,j}^p}{\tau^2} - \mu \frac{v_{2i+1,j}^p - 2v_{2ij}^p + 2v_{2i-1,j}^p}{h^2} - \\ & - (\lambda + \mu) \frac{v_{2i,j+1}^p - 2v_{2ij}^p + v_{2,j+1}^p}{l^2} - (\lambda + \mu) \frac{v_{1i-1,j+1}^p + v_{1i,j+1}^p - v_{1i+1,j}^p + v_{1i,j}^p}{4hl} = \\ & = \gamma \frac{u_{i,j+1}^p - u_{i,j-1}^p}{2l} + f_2(x_i, y_j, t_p), \end{aligned} \quad (2.2.2)$$

$$\frac{u_{i,j}^{p+1} - u_{i,j}^p}{\tau} = k \frac{u_{i-1,j}^{p+1} - 2u_{ij}^{p+1} + u_{i+1,j}^{p+1}}{h^2} + k \frac{u_{i,j+1}^{p+1} - 2u_{ij}^{p+1} + u_{i,j-1}^{p+1}}{l^2}. \quad (2.2.3)$$

2.3. Решение многомерных уравнений

2.3.1. Разностные схемы расщепления

Разностные схемы расщепления – важное средство при расчете решений нестационарных задач математической физики. Рассмотрим правило построения схемы расщепления в двумерном уравнении теплопроводности.

Рассмотрим задачу Коши:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, 0 < t \leq T, \quad (2.3.1)$$

$$u(x, y, 0) = \varphi(x, y), -\infty < x, y < \infty,$$

где $\varphi(x, y)$ - известная функция.

Обозначим

$$Au = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

По направлению x и y выберем квадратичную сетку с шагом h . Выберем шаг τ из условия

$$N\tau \leq T \leq (N+1)\tau.$$

Координаты некоторого узла (x_m, y_n, t_p) будут определены по формулам:

$$x_m = mh, y_n = nh, t_p = p\tau,$$

$$m, n = 0, \pm 1, \pm 2, \dots, p = 0, 1, \dots, N.$$

Предположим, что нам известно решение исходной задачи в момент времени $t = t_p$, т.е. известно значение $u(x, y, t_p)$.

Обозначим u^p значение функции $u(x, y, t)$ при $t = t_p$.

Выразим значение $u^{p+1} = u(x, y, t_{p+1})$ через $u^p = u(x, y, t_p)$ и значение оператора A .

Разложим в ряд Тейлора[2]:

$$\begin{aligned} u(x, y, t_{p+1}) &= u(x, y, t_p + \tau) = u(x, y, t_p) + \tau \frac{\partial u(x, y, t_p)}{\partial \tau} + O(\tau^2) = \\ &= u(x, y, t_p) + \tau u(x, y, t_p) + O(\tau^2) = (E + \tau A)u(x, y, t_p) + O(\tau^2). \end{aligned}$$

Положим $A_1 \equiv \frac{\partial^2}{\partial x^2}$, $A_2 \equiv \frac{\partial^2}{\partial y^2}$, тогда $A = A_1 + A_2$.

Вместе с задачей (2.3.1.) рассмотрим вспомогательные задачи[3]

$$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2}, t_p < t \leq t_{p+1},$$

$$v(x, y, t_p) = u(x, y, t_p), -\infty < x, y < \infty; \quad (2.3.2)$$

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial y^2}, t_p < t \leq t_{p+1},$$

$$w(x, y, t_p) = v(x, y, t_p), -\infty < x, y < \infty; \quad (2.3.3)$$

Решим данные задачи последовательно. Вычислим задачу (2.3.2), затем задачу (2.3.3). Выясним связь между значениями $u(x, y, t_{p+1})$ и $w(x, y, t_{p+1})$.

Имеем[4]

$$v(x, y, t_{p+1}) = (E + \tau A_1)u(x, y, t_p) + O(\tau^2),$$

$$w(x, y, t_{p+1}) = (E + \tau A_2)w(x, y, t_p) + O(\tau^2) =$$

$$= (E + \tau A_2)(x, y, t_{p+1}) + O(\tau^2) = (E + \tau A_2) \left[(E + \tau A_1)u(x, y, t_p) + O(\tau^2) \right] =$$

$$= (E + \tau A)u(x, y, t_p) + O(\tau^2) = (x, y, t_{p+1}) + O(\tau^2).$$

Решив уравнения (2.3.2) и (2.3.3) получим значения функции $w(x, y, t_{p+1})$, которые отличаются от истинных значений $u(x, y, t_{p+1})$ на величину $O(\tau^2)$. Нам удалось заменить процесс решения двумерной задачи (2.3.1) процессом решения двух одномерных задач (2.3.2) и (2.3.3). В объёме вычислений второй процесс имеет преимущества и это делает изложенную здесь операцию расщепления выгодной.

Построим схемы расщепления (2.3.1).

1) Явная схема

Воспользуемся двумя явными шаблонами:

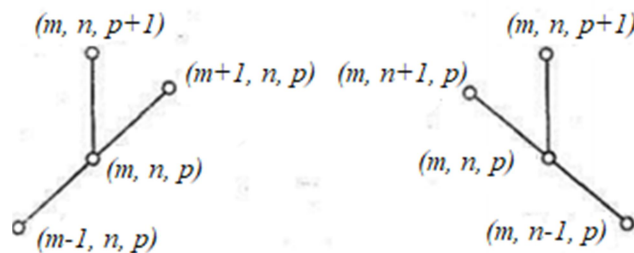


Рисунок 3 – Шаблон явной схемы

Используя данные схемы для задачи (2.3.2) и (2.3.3) получим:

$$\frac{v_{mn}^{p+1} - v_{mn}^p}{\tau} = \wedge_{xx}(v_{mn}^p), \quad v_{mn}^p = u_{mn}^p, m, n = 0, \pm 1, \dots, p = 0, \dots, N-1,$$

$$\frac{w_{mn}^{p+1} - w_{mn}^p}{\tau} = \wedge_{yy}(w_{mn}^p), \quad w_{mn}^p = v_{mn}^{p+1}, m, n = 0, \pm 1, \dots, p = 0, \dots, N-1,$$

обозначим:

$$\wedge_{xx}(v_{mn}^p) = \frac{v_{m+1,n}^p - 2v_{mn}^p + v_{m-1,n}^p}{h^2},$$

$$\wedge_{yy}(w_{mn}^p) = \frac{w_{m+1,n}^p - 2w_{mn}^p + w_{m-1,n}^p}{l^2}.$$

Формулы (2.3.2) и (2.3.3) образуют явную разностную схему расщепления для (2.3.1). Значения v_{mn}^{p+1} вычисляем по формуле:

$$v_{mn}^1 = \Phi(x_m, y_n) + \tau \wedge_{xx}(v_{mn}^0).$$

Вычисляем w_{mn}^1 :

$$w_{mn}^1 = v_{mn}^1 + \tau \wedge_{yy}(v_{mn}^1).$$

Таким образом, вычисляем значение w_{mn}^1 на первом временном слое. Полагаем, что $u_{mn}^1 \approx w_{mn}^1$, это означает, что определены приближенные значения решения задачи (2.3.1) на первом временном слое. Значение на последнем временном слое вычисляем аналогично.

Для устойчивости схемы (2.3.4) и (2.3.5) требуются ограничения на $\tau \leq \frac{1}{2}h^2$.

2) Неявная схема

Для задач (2.3.2) и (2.3.3) запишем разностные схемы:

$$\frac{v_{mn}^{p+1} - v_{mn}^p}{\tau} = \wedge_{xx}(v_{mn}^{p+1}),$$

$$v_{mn}^p = u_{mn}^p, m, n = 0, \pm 1, \dots, p = 0, \dots, N-1, \quad (2.3.6)$$

$$\frac{w_{mn}^{p+1} - w_{mn}^p}{\tau} = \wedge_{yy}(w_{mn}^{p+1}),$$

$$w_{mn}^p = v_{mn}^{p+1}, m, n = 0, \pm 1, \dots, p = 0, \dots, N - 1. \quad (2.3.7)$$

Разностные схемы были составлены по неявным шаблонам.

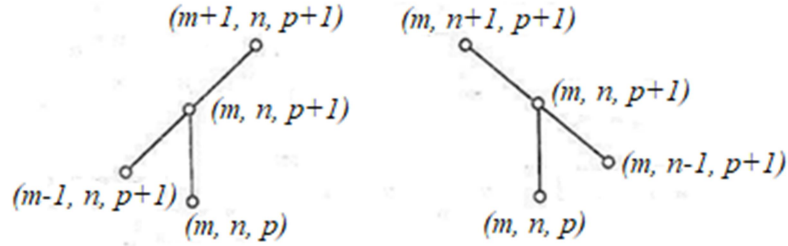


Рисунок 4 – Шаблон неявной схемы

Формулы (2.3.6) и (2.3.7) образуют неявную разностную схему расщепления для (2.3.1). Вычислим значение v_{mn}^1 , как решение СЛАУ

$$v_{mn}^1 + \tau \wedge_{xx}(v_{mn}^1) = \varphi(x_m, y_n).$$

w_{mn}^1 будем находить как решение СЛАУ:

$$w_{mn}^1 + \tau \wedge_{yy}(w_{mn}^1) = v_{mn}^1.$$

Где v_{mn}^1 уже получено на первом этапе. Полагаем, что $u_{mn}^1 \approx w_{mn}^1$. Значит, мы вычислили приближенные значения решения задачи (2.3.1). Решение систем линейных алгебраических уравнений можно осуществить по алгоритму разностной прогонки.

2.3.2. Экономичные разностные схемы

Построим разностную схему для смешанной граничной задачи уравнения теплопроводности:[6]

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, 0 < t \leq T, 0 \leq x, y \leq 1,$$

$$u(x, y, 0) = \psi(x, y),$$

$$u(x, y, t)|_{\Gamma} = 0, \quad (2.3.1)$$

где Γ -боковые грани прямоугольного параллелепипеда $D = \{0 \leq x, y \leq 1, 0 < t \leq T\}$. Предположим, что (2.3.1) имеет единственное решение $u(x, y, t)$ и это решение непрерывно в $\bar{D} = D + \Gamma$ вместе со своими производными $\frac{\partial^i u}{\partial t^i}$, $i = 1, 2$, $\frac{\partial^k u}{\partial x^k}$, $\frac{\partial^k u}{\partial y^k}$, $k = 3, 4$.

Выберем квадратную сетку по пространству и координаты узлов (x_m, y_n, t_p) определим по правилу

$$(x_m, y_n, t_p) = (mh, nh, p\tau), m, n = 0, 1, \dots, M, h = \frac{1}{M},$$

$$\tau > 0, p = 0, 1, \dots, N, N\tau \leq T < (N+1)\tau.$$

1) Явная схема.

Запишем разностную схему для (2.3.1) используя разностный шаблон (Рис.5).

$$L_h^{(1)} u^{(h)} = f^{(h)}, \quad (2.3.2)$$

где

$$L_h^{(1)} u^{(h)} = \begin{cases} \frac{u_{mn}^{p+1} - u_{mn}^p}{\tau} - \Delta_{xx}(u_{mn}^p) - \Delta_{yy}(u_{mn}^p), \\ m, n = 1, 2, \dots, M-1, p = 0, 1, \dots, N-1, \\ u_{mn}^0, m, n = 0, 1, \dots, M, \\ u_{mn}^{p+1}, (mh, nh, (p+1)\tau) \in \Gamma_h. \end{cases}$$

Обозначим узлы сетки Γ_h ,

$$f^{(h)} \equiv \begin{cases} 0, \\ \psi(x_m, y_n), m, n = 0, 1, \dots, M, \\ 0. \end{cases}$$

Разностная схема (2.3.2) явная и аппроксимирует задачу (2.3.1) с погрешностью порядка $O(\tau + h^2)$. С помощью принципа максимума установим, что при $\tau \leq \frac{1}{4}h^2$ разностная схема (2.3.2) устойчива.

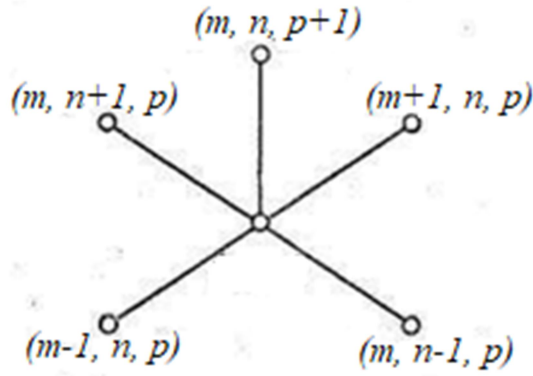


Рисунок 6 – Шаблон явной схемы

2) Неявная схема

Используя неявный шаблон, запишем разностную схему для (2.3.1)[6]:

$$L_h^{(2)} u^{(h)} = f^{(h)}, \quad (2.3.3)$$

где

$$L_h u^{(h)} = \begin{cases} \frac{u_{mn}^{p+1} - u_{mn}^p}{\tau} - \Delta_{xx}(u_{mn}^{p+1}) - \Delta_{yy}(u_{mn}^{p+1}), \\ m, n = 1, 2, \dots, M-1, p = 0, 1, \dots, N-1, \\ u_{mn}^0, m, n = 0, 1, \dots, M, \\ u_{mn}^{p+1}, (mh, nh, (p+1)\tau) \in \Gamma_h, \end{cases}$$

$$f^{(h)} \equiv \begin{cases} 0, \\ \psi(x_m, y_n), m, n = 0, 1, \dots, M, \\ 0. \end{cases}$$

Разностная схема (2.3.3) неявная и аппроксимирует задачу (2.3.1) с погрешностью порядка $O(\tau + h)$. Устойчива при любых значениях $r > 0, r = \frac{\tau}{h^2}$.

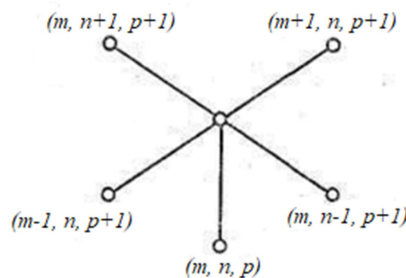


Рисунок 7 – Шаблон неявной схемы

Обозначим u^k совокупность значений сеточной функции на слое k , т.е.

$$u^k = \{u_{mn}^k, m, n = 0, 1, \dots, M\}, k = 0, 1, \dots, N. [7]$$

Пусть известно значение u^p . Для того чтобы получить значение u^{p+1} по разностной схеме (2.3.2), мы выполним число арифметических операций, пропорциональное числу неизвестных u_{mn}^{p+1} . Неизвестных будет $(M-1)^2$, т.к. значения на Γ_h известны: $u_{mn}^{p+1} = 0$, если узел принадлежит Γ_h .

Введем понятие экономичной разностной схемы. Двумерная разностная схема экономична, если для вычисления значений u^{p+1} по значениям u^p эта схема требует выполнения числа арифметических операций, пропорционального числу неизвестных значений на слое $p+1$.

Разностная схема (2.3.2) экономична, вычисления по этой схеме будут устойчивыми при выполнении условия $\tau \leq \frac{1}{4}h^2$, которое является жестким ограничением на шаг τ .

Если использовать разностную схему (2.3.3), то для вычисления u^p и u^{p+1} нужно решить СЛАУ, в которой число уравнений $(M-1)^2$, количество этих уравнений равно количеству неизвестных. Даже при условии, что эта система будет решаться по методу Гаусса, число операций, необходимых для вычисления u^p и u^{p+1} , будет приблизительно пропорционально $[(M-1)^2]^3$. Следовательно, схему (2.3.3) нельзя назвать экономичной. Но отметим, то, что важным положительным свойством этой схемы является ее абсолютная устойчивость.

Построим разностную схему, которая аппроксимирует задачу (2.3.1) и обладает свойствами:

- 1) экономичности;
- 2) абсолютной устойчивости;

Запишем разностную схему, используя идею расщепления и формулы (2.3.2),(2.3.3).

Получим

$$L_h u^{(h)} = f^{(h)}, \quad (2.3.4)$$

где

$$L_h u^{(h)} = \begin{cases} \frac{u_{mn}^{p+1} - u_{mn}^p}{\tau} + \wedge_{yy}(u_{mn}^{p+1}), \\ m, n = 1, 2, \dots, M-1, p = 0, 1, \dots, N-1, \\ u_{mn}^0, m, n = 0, 1, \dots, M, p = 0, 1, \dots, N-1, \\ u_{mn}^{p+1}, (mh, nh, (p+1)\tau) \in \Gamma_h. \end{cases}$$

$$f^{(h)} \equiv \begin{cases} 0, m, n = 1, 2, \dots, M-1, p = 0, 1, \dots, N-1, \\ 0, (mh, nh, (p+1)\tau) \in \Gamma_h, \\ \tilde{u}_{mn}^{p+1}, m, n = 0, 1, \dots, M-1, p = 0, 1, \dots, N-1, \end{cases}$$

Сеточная функция $\tilde{u}_{mn}^{p+1}$ вычисляется, как решение вспомогательной разностной схемы[8]

$$\begin{aligned} & \frac{\tilde{u}_{mn}^{p+1} - u_{mn}^p}{\tau} + \wedge_{xx}(\tilde{u}_{mn}^{p+1}), \\ & m, n = 1, 2, \dots, M-1, p = 0, 1, \dots, N-1, \\ & u_{mn}^0 = \psi_{mn}, m, n = 0, 1, \dots, M, \\ & \tilde{u}_{mn}^{p+1} = 0, (mh, nh, (p+1)\tau) \in \Gamma_h. \end{aligned} \quad (2.3.5)$$

Разностная схема (2.3.4) аппроксимирует задачу (2.3.1) с погрешностью порядка $O(\tau + h^2)$, абсолютно устойчива и экономична. Здесь мы уделим большее внимание вопросу реализации разностной схемы (2.3.4) и установлены свойства ее экономичности. Посмотрим, как по значениям $u_{mn}^0 = \psi_{mn}$ вычислим u_{mn}^1 на первом слое и затем по значениям u_{mn}^1 значения u_{mn}^2 и так далее.

В (2.3.5) положим $p = 0$, получим

$$\begin{aligned}\frac{\tilde{u}_{mn}^1 - u_{mn}^0}{\tau} &= \frac{\tilde{u}_{m+1,n}^1 - 2\tilde{u}_{m,n}^1 + \tilde{u}_{m-1,n}^1}{h^2}, \\ m &= 1, 2, \dots, M-1, \\ \tilde{u}_{0n}^1 &= 0, \tilde{u}_{Mn}^1, m, n = 1, 2, \dots, M-1.\end{aligned}\tag{2.3.6}$$

При фиксированном n формула (2.3.6) представляет граничную задачу для разностного уравнения второго порядка. Для решения данной задачи используем метод разностной прогонки.

Положим $p = 0$ в задачу (2.3.4):

$$\begin{aligned}\frac{\tilde{u}_{mn}^1 - u_{mn}^1}{\tau} &= \frac{\tilde{u}_{m+1,n}^1 - 2\tilde{u}_{m,n}^1 + \tilde{u}_{m-1,n}^1}{h^2}, \\ u_{m0}^1 &= 0, u_{mM}^1 = 0, n = 1, 2, \dots, M-1, \\ m &= 1, 2, \dots, M-1.\end{aligned}\tag{2.3.7}$$

Данную задачу (2.3.7) можно решить с помощью метода разностной прогонки, если значение m фиксировано.

2.3.3. Схема переменных направлений

Рассмотрим уравнение теплопроводности [9]

$$\begin{cases} \frac{\partial u}{\partial t} = Lu + f(x, t), t \in (0, t_0], \\ u|_{\Gamma} = \mu(x, t), u(x, 0) = u_0(x), \\ Lu = \Delta u = (L_1 + L_2)u, L_\alpha u = \frac{\partial^2 u}{\partial x_\alpha^2}, \alpha = 1, 2. \end{cases}\tag{2.3.8}$$

Для уравнения (2.3.8) построим равномерную сетку с шагами h, l . Пусть γ_h - граница сеточной области, содержащая все узлы на сторонах прямоугольника, кроме вершин. Оператор L_α заменим разностным оператором Λ_α :

$$\Lambda = \Lambda_1 + \Lambda_2$$

Неявная схема на каждом слое приводит к разностной краевой задаче [10]:

$$\begin{cases} A_i y_{i-1} - C_i y_i + B_i y_{i+1} = -F, \\ y_0 = \mu_1, y_N = \mu_2, A_i > 0, B_i > 0, \\ C_i \geq A_i + B_i, \end{cases}\tag{2.3.9}$$

Данная задача решается методом прогонки.

Вернемся к двумерной задаче в прямоугольнике. Сетку можно представить, как совокупность узлов. Число узлов в каждой строке равно $N_1 + 1$, а в каждом столбце имеется $N_2 + 1$ узлов.

Если на каждом столбце или строке решать задачу (2.3.9) методом прогонки при фиксированном i , то для отыскания решения во всех узлах сетки, понадобится число $O(N_1, N_2)$ арифметических действий, пропорциональное числу узлов двумерной сетки. Идея большинства экономических методов состоит в сведении перехода со слоя на слой к последовательному решению одномерных задач.

Данную алгоритмическую идею четко выражает неявная схема переменных направлений. Переход от слоя n к слою $n+1$ совершается в два этапа с шагами $0,5\tau$ [11]:

$$\frac{y^{n+\frac{1}{2}} - y^n}{0,5\tau} = \Lambda_1 y^{n+\frac{1}{2}} + \Lambda_2 y^n + \varphi^n, \quad (2.3.10)$$

$$\frac{y^{n+1} - y^{n+\frac{1}{2}}}{0,5\tau} = \Lambda_1 y^{n+\frac{1}{2}} + \Lambda_2 y^{n+1} + \varphi^n. \quad (2.3.11)$$

Уравнения (2.3.10) и (2.3.11) пишутся во всех внутренних узлах $x = x_i$ сетки и для всех $t = t_n > 0$. Первая схема неявная по направлению x и явная по y . Вторая схема явная по x и неявная по y . Добавим начальные условия к уравнению (2.3.10) и (2.3.11): $y(x, 0) = u_0(x)$

Перепишем уравнения в виде:

$$\begin{cases} \frac{2}{\tau} \bar{y} - \Lambda_1 \bar{y} = F, F = \frac{2}{\tau} y + \Lambda_2 y + \varphi, \\ \frac{2}{\tau} \bar{y} - \Lambda_2 \bar{y} = F, F = \frac{2}{\tau} y + \Lambda_1 y + \varphi. \end{cases} \quad (2.3.12)$$

Перепишем уравнение (2.3.12) в виде[8]:

$$\begin{cases} \frac{1}{h^2} y_{i-1} - 2\left(\frac{1}{h^2} + \frac{1}{\tau}\right) \bar{y}_i + \frac{1}{h^2} y_{i+1} = -F \\ i = 1, \dots, N-1 \end{cases} \quad (2.3.13)$$

$$\begin{cases} \frac{1}{l^2} y_{i-1} - 2\left(\frac{1}{l^2} + \frac{1}{\tau}\right) \bar{y}_i + \frac{1}{l^2} y_{i+1} = -F \\ i = 1, \dots, N-1 \end{cases} \quad (2.3.14)$$

Методом прогонки решаем (2.3.13) и определяем $\bar{y}$ во всех узлах сетки, после вычисляем F и решаем задачу (2.3.14). При переходе от слоя $n+1$ к слою $n+2$ процедура счёта повторяется, происходит чередование направлений.

2.4. Порядок аппроксимации, условия сходимости

Каждое уравнение (2.4.1) и (2.4.2)

$$\frac{u_{i,j}^{p+\frac{1}{2}} - u_{i,j}^p}{0,5\tau} = k(\Lambda_1 u_{i,j}^{p+\frac{1}{2}} + \Lambda_2 u_{i,j}^p), \quad (x_i, y_j) \in \omega_h, \quad (2.4.1)$$

$$\frac{u_{i,j}^{p+1} - u_{i,j}^{p+\frac{1}{2}}}{0,5\tau} = k(\Lambda_1 u_{i,j}^{p+\frac{1}{2}} + \Lambda_2 u_{i,j}^{p+1}), \quad (x_i, y_j) \in \omega_h. \quad (2.4.2)$$

аппроксимирует исходное уравнение (2.1.8) по τ с первым порядком аппроксимации и по h, l вторым порядком. Сумма погрешности аппроксимации этих уравнений имеет второй порядок

Разностная схема имеет вид:

$$\frac{u_{i,j}^{p+1} - u_{i,j}^p}{\tau} = k \frac{u_{i-1,j}^{p+1} - 2u_{ij}^{p+1} + u_{i+1,j}^{p+1}}{h^2} + k \frac{u_{i,j+1}^{p+1} - 2u_{ij}^{p+1} + u_{i,j-1}^{p+1}}{l^2}.$$

Данная схема абсолютно устойчива по начальным данным и по граничным условиям. Шаги τ, h, l можно выбирать по правилам Рунге.

Основная идея Рунга состоит в вычислении решения при шаге h , потом вычисляют решение при шаге $2h$ и сравнивают значения $u_h(x, y)$ и $u_{2h}(x, y)$ в одинаковых узлах. Если эти значения совпадают на заданном числе знаков, то решение $u_h(x, y)$ принимают за искоемое. Если не совпадают, то шаг h делят по-

полам и вычисляют решение $u_h(x, y)$. Далее контроль точности в вычислении значений осуществляют аналогично $u_h(x, y)$.

Решение разностной схемы u_{ij}^n при $\tau, h, l \rightarrow \infty$ сходятся к точному решению в среднеквадратичной норме на временном слое.

2.5. Способ реализации

Заменим частные производные разностными отношениями

$$\begin{aligned}\left. \frac{\partial u}{\partial t} \right|_{(x_i, y_j, t_p)} &= \frac{u_{i,j}^{p+1} - u_{i,j}^p}{\tau} + O(\tau), \\ \left. \frac{\partial^2 u}{\partial x^2} \right|_{(x_i, y_j, t_p)} &= \frac{u_{i-1,j}^{p+1} - 2u_{ij}^{p+1} + u_{i+1,j}^{p+1}}{h^2} + O(h^2), \\ \left. \frac{\partial^2 u}{\partial y^2} \right|_{(x_i, y_j, t_p)} &= \frac{u_{i,j+1}^{p+1} - 2u_{ij}^{p+1} + u_{i,j-1}^{p+1}}{l^2} + O(l^2).\end{aligned}$$

подставим в уравнение (2.1.2) получим СЛАУ

$$\frac{u_{i,j}^{p+1} - u_{i,j}^p}{\tau} = k \frac{u_{i-1,j}^{p+1} - 2u_{ij}^{p+1} + u_{i+1,j}^{p+1}}{h^2} + k \frac{u_{i,j+1}^{p+1} - 2u_{ij}^{p+1} + u_{i,j-1}^{p+1}}{l^2}. \quad (2.5.1)$$

Для решения уравнения (2.5.1) применяем продольно поперечную разностную схему, в данной схеме разбивается переход с одного временного слоя на другой на два этапа:

1. Переход на промежуточный временной слой $(p + \frac{1}{2})$ с шагом $0,5\tau$, осуществляется методом разностной прогонки системы в направлении оси x .
2. Переход на $(p + 1)$ -й временной слой с промежуточного $(p + \frac{1}{2})$ -го с шагом $0,5\tau$, производится методом разностной прогонки в направлении оси y .

Разностная краевая задача представляет собой СЛАУ с трехдиагональной матрицей размера $(N + 1) \times (N + 1)$. [12]

$$\begin{aligned} A_i u_{i-1} - C_i u_i + B_i u_{i+1} &= -F_i, \\ A_i \neq 0, B_i \neq 0, i &= \overline{1, N-1}. \end{aligned} \quad (2.5.2)$$

$$u_0 = \lambda_1 u_1 + \mu_1, u_N = \lambda_2 u_{N-1} + \mu_2$$

Для решения задачи (2.5.2) используем метод прогонки. Предположим, что соотношение

$$u_i = \alpha_{i+1} u_{i+1} + \beta u_{i+1} \quad (2.5.3)$$

С неопределенными коэффициентами α_{i+1} и β_{i+1} , и подставим $u_{i-1} = \alpha_i u_i + \beta_i$ в

$$(A_i \alpha_i - C_i) u_i + b_i u_{i+1} = -(F_i + A_i B_i), \quad (2.5.4)$$

Сравним с (2.5.3), находим[13]:

$$\begin{aligned} \alpha_{i+1} &= \frac{B_i}{C_i - A_i \alpha_i}, i = \overline{1, N-1}. \\ \beta_{i+1} &= \frac{A_i \beta_i + F_i}{C_i - A_i \alpha_i}, i = \overline{1, N-1}. \end{aligned}$$

Вычисления по формуле (2.5.3) ведутся путем перехода от $i+1$ к i , и для начала этих вычислений надо знать u_N . Определим u_N из краевого условия и условия (2.5.3) при $i = N-1$:

$$u_{N-1} = \alpha_N u_N + \beta_N$$

отсюда найдём

$$u_N = \frac{\mu_2 + \lambda_2 \beta_N}{1 - \alpha_N \lambda_2}$$

Шаблон метода переменных направлений (рис.8).

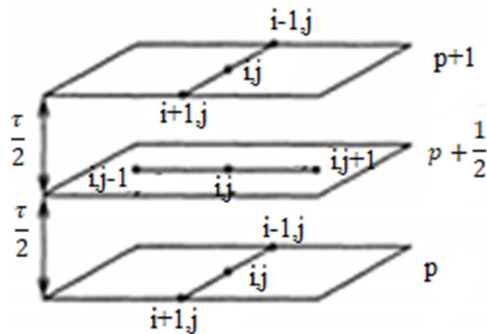


Рисунок 8 – Шаблон метода переменных направлений

На первом этапе находим вспомогательную функцию $u_{i,j}^{p+\frac{1}{2}}$ из уравнения:

$$\frac{u_{i,j}^{p+\frac{1}{2}} - u_{i,j}^p}{0,5\tau} = k(\Lambda_1 u_{i,j}^{p+\frac{1}{2}} + \Lambda_2 u_{i,j}^p), \quad (x_i, y_j) \in \omega_h, \quad (2.5.5)$$

Затем $u_{i,j}^{p+1}$ найдём из уравнения:

$$\frac{u_{i,j}^{p+1} - u_{i,j}^{p+\frac{1}{2}}}{0,5\tau} = k(\Lambda_1 u_{i,j}^{p+\frac{1}{2}} + \Lambda_2 u_{i,j}^{p+1}), \quad (x_i, y_j) \in \omega_h. \quad (2.5.6)$$

Обозначим разностные операторы:

$$\Lambda_1 u_{ij} = \frac{u_{i-1,j} - u_{ij} + u_{i+1,j}}{h^2},$$

$$\Lambda_2 u_{ij} = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{l^2},$$

Запишем уравнения (2.5.4) и (2.5.5) в виде:

$$\frac{u_{i,j}^{p+\frac{1}{2}} - u_{i,j}^p}{0,5\tau} = k\left(\frac{u_{i-1,j}^{p+\frac{1}{2}} - 2u_{i,j}^{p+\frac{1}{2}} + u_{i+1,j}^{p+\frac{1}{2}}}{h^2} + \frac{u_{i,j+1}^p - 2u_{i,j}^p + u_{i,j-1}^p}{l^2}\right), \quad (2.5.7)$$

$$\frac{u_{i,j}^{p+1} - u_{i,j}^{p+\frac{1}{2}}}{0,5\tau} = k\left(\frac{u_{i-1,j}^{p+\frac{1}{2}} - 2u_{i,j}^{p+\frac{1}{2}} + u_{i+1,j}^{p+\frac{1}{2}}}{h^2} + \frac{u_{i,j+1}^{p+1} - 2u_{i,j}^{p+1} + u_{i,j-1}^{p+1}}{l^2}\right). \quad (2.5.8)$$

Уравнения (2.5.6) и (2.5.7) можно решать последовательно методом разностной прогонки. Значения сеточной функции u_{ij}^p будут определены.

Из разностных схем (2.2.1) и (2.2.2) определим:

$$\begin{aligned} v_{1i,j}^{p+1} = & -v_{1i,j}^{p-1} + 2v_{1i,j}^p - \gamma\tau^2 \frac{u_{i+1,j}^p - u_{i-1,j}^p}{2h} - \tau^2 f_1(x_i, y_j, t_p) + \\ & + \tau^2(\lambda + 2\mu) \frac{v_{1i+1,j}^p - 2v_{1ij}^p + v_{1i-1,j}^p}{h^2} + \tau^2\mu \frac{v_{1i,j+1}^p - 2v_{1ij}^p + v_{1i,j-1}^p}{l^2} - \\ & - \tau^2(\lambda + \mu) \frac{v_{2i+1,j+1}^p + v_{2i,j+1}^p - v_{2i+1,j}^p + v_{2i,j}^p}{4hl} \end{aligned} \quad (2.5.9)$$

$$\begin{aligned}
v_{2i,j}^{p+1} = & -v_{2i,j}^{p-1} + 2v_{1i,j}^p - \gamma\tau^2 \frac{u_{i,j+1}^p - u_{i,j-1}^p}{2l} - \tau^2 f_2(x_i, y_j, t_p) + \\
& + \tau^2 \mu \frac{v_{2i+1,j}^p - 2v_{2ij}^p + 2v_{2i-1,j}^p}{h^2} + \tau^2 (\lambda + \mu) \frac{v_{2i,j+1}^p - 2v_{2ij}^p + v_{2,j+1}^p}{l^2} - \\
& - \tau^2 (\lambda + \mu) \frac{v_{1i+1,j+1}^p + v_{1i,j+1}^p - v_{1i+1,j}^p + v_{1i,j}^p}{4hl} \\
& i = \overline{0, N-1}, \quad j = \overline{0, M-1}, \quad p = \overline{0, P-1}.
\end{aligned} \tag{2.5.10}$$

Значение сеточных функций на нулевом и первом временных слоях определим из условий в начальный момент времени:

$$\begin{aligned}
v_1^0(x_i, y_j) &= w_1^0(x_i, y_j), \\
v_2^0(x_i, y_j) &= w_2^0(x_i, y_j), \\
v_{1ij}^1 &= \tau w_{1ij}^1 + v_{2ij}^0, \\
v_{2ij}^1 &= \tau w_{1ij}^2 + v_{2ij}^0.
\end{aligned}$$

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

Рассмотрим применение выше указанных методов к решению следующих задач.

3.1. Модельная задача

Имеем систему

$$\begin{aligned}(\lambda + \mu) \frac{\partial^2 v_1}{\partial x^2} + \mu \frac{\partial^2 v_1}{\partial y^2} + (\lambda + \mu) \frac{\partial^2 v_2}{\partial x \partial y} - \gamma \frac{\partial u}{\partial x} + f_1 &= 0, \\ \mu \frac{\partial^2 v_2}{\partial x^2} + (\lambda + 2\mu) \frac{\partial^2 v_2}{\partial y^2} + (\lambda + \mu) \frac{\partial^2 v_1}{\partial x \partial y} - \gamma \frac{\partial u}{\partial y} + f_2 &= 0,\end{aligned}$$

где числовые коэффициенты заданы:

$$\lambda = 0,1; \quad \mu = 0,2; \quad \gamma = (3\lambda + 2\mu)\alpha; \quad \alpha = 0,1;$$

$$\Delta u = 0 \text{ - область, где } 0 \leq x \leq 1, \quad 1 \leq y \leq 3.$$

Запишем граничные условия:

$$u(0, y) = \cos(y), \quad u(x, 1) = e^x \cos(1),$$

$$u(1, y) = e \cos(y), \quad u(x, 3) = e^x \cos(3).$$

$$v_1(0, y) = y, \quad v_1(1, y) = 1 + y,$$

$$v_1(x, 1) = x + 1, \quad v_1(x, 3) = x + 3.$$

$$v_2(0, y) = -y, \quad v_2(1, y) = 1 - y, \quad v_2(x, 1) = x - 1, \quad v_2(x, 3) = x - 3.$$

Точным решением являются функции:

$$u = e^x \cos(y); \quad v_1 = x + y; \quad v_2 = x - y;$$

3.2. Модельная задача

Имеем систему:

$$\rho \frac{\partial^2 v_1}{\partial t^2} + (\lambda + \mu) \frac{\partial^2 v_1}{\partial x_1^2} - \mu \frac{\partial^2 v_1}{\partial x_2^2} - (\lambda + \mu) \frac{\partial^2 v_2}{\partial x_1 \partial x_2} + \gamma \frac{\partial u}{\partial x_1} = f_1,$$

$$\rho \frac{\partial^2 v_2}{\partial t^2} - \mu \frac{\partial^2 v_2}{\partial x_1^2} - (\lambda + \mu) \frac{\partial^2 v_2}{\partial x_2^2} - (\lambda + \mu) \frac{\partial^2 v_1}{\partial x_1 \partial x_2} + \gamma \frac{\partial u}{\partial x_2} = f_2,$$

где числовые коэффициенты:

$$\rho = 0,3; \lambda = 0,1; \mu = 0,2; \gamma = (3\lambda + 2\mu)\alpha; \alpha = 0,1;$$

$$0 \leq x \leq 1, 1 \leq y \leq 3, 0 \leq t \leq 1.$$

Запишем граничные условия:

$$u(1, y, t) = e^{-2t} \cos(1) \sin(y), \quad u(x, 1, t) = e^{-2t} \cos(x) \sin(1),$$

$$u(0, y, t) = e^{-2t} \sin(y), \quad u(x, 3, t) = e^{-2t} \cos(x) \sin(3).$$

$$v_1(0, y, t) = y + t, \quad v_1(1, y, t) = 1 + y + t,$$

$$v_1(x, 1, t) = x + 1 + t, \quad v_1(x, 3, t) = x + 3 + t, \quad v_1(x, y, 0) = x + y + 1$$

$$v_2(0, y, t) = -y + t, \quad v_2(1, y, t) = 1 - y + t,$$

$$v_2(x, 1, t) = x - 1 + t, \quad v_2(x, 3, t) = x - 3 + t, \quad v_2(x, y, 0) = x - y.$$

Точным решением являются функции:

$$u = e^{-2t} \sin(x) \cos(y); \quad v_1 = x + y + t; \quad v_2 = x - y + t;$$

3.3. Анализ полученных результатов

После составления модельных задач переходим к численному решению. Нахождение неизвестных значений функции будем осуществлять, используя программы, разработанные в среде Microsoft Visual Studio 2010 на языке C/C++, листинг которых приведен в приложениях А и Б. Зависимость погрешности в графическом виде представлены в приложение В.

Результаты численного решения представлены в таблицах 1–6. Данные таблицы содержат сведения о точном и приближенном решении функции, а также погрешность и значения x, y .

Результаты модельной задачи №3.1 представлены в таблицах 1–3.

Таблица 1 – Таблица для функции v_I

x	Y	$v_I(x, y)$	$v_{I_n}(x, y)$	Погрешность
0,05	1,1	1,15	1,14847	0,0015271
0,05	1,2	1,25	1,2475	0,0024963

Окончание таблицы 1 – Таблица для функции v_1

x	y	$v_1(x, y)$	$v_{1_n}(x, y)$	Погрешность
0,1	1,1	1,2	1,19737	0,00262906
0,1	1,2	1,3	1,29543	0,00456721
0,15	1,1	1,25	1,24649	0,00351194
0,15	1,2	1,35	1,34367	0,00633494
0,2	1,1	1,3	1,29573	0,00426905
0,2	1,2	1,4	1,39212	0,00787807
0,25	2,7	2,95	2,92989	0,0201103
0,25	2,8	3,05	3,03484	0,0151646

Таблица 2 – Таблица для функции v_2

x	y	$v_2(x, y)$	$v_{2_n}(x, y)$	Погрешность
0,05	1,1	-1,05	-1,05133	0,00133188
0,05	1,2	-1,15	-1,15202	0,00202259
0,1	1,1	-1	-1,00176	0,00176193
0,1	1,2	-1,1	-1,1028	0,00280142
0,15	1,1	-0,95	-0,951839	0,00183875
0,15	1,1	-0,95	-0,951839	0,00183875
0,2	1,1	-0,9	-0,901745	0,00174515
0,2	1,1	-0,9	-0,901745	0,00174515
0,25	2,9	-2,65	-1,70329	0,00328759
0,3	1,1	-0,8	-0,751326	0,00132551

Таблица 3 – Таблица для функции u

x	y	$u(x, y)$	$u_n(x, y)$	Погрешность
0,05	1,1	0,476854	0,475873	0,0009810
0,05	1,2	0,3809362	0,3809387	0,0000025
0,1	1,2	0,4004673	0,4004718	0,0000045
0,1	1,3	0,2956383	0,2956319	0,0000193
0,15	1,6	0,033925	0,0339101	0,0000149
0,15	2,9	1,128102	1,128088	0,0000014
0,2	1,1	0,5541278	0,5540218	0,000106

Результаты модельной задачи №3.2 представлены в таблицах 4–6.

Таблица 4 – Таблица для функции v_1

x	y	t	$v_1(x, y)$	$v_{1_n}(x, y)$	Погрешность
0,05	1,1	0,05	1,2	1,1987	0,00130179
0,05	2,9	0,95	3,9	3,89944	0,000556794

Окончание таблицы 4 – Таблица для функции v_I

x	y	t	$v_I(x, y)$	$v_{I_n}(x, y)$	Погрешность
0,1	1,2	0,05	1,35	1,34662	0,0033812
0,1	2	0,05	2,15	2,14476	0,00523938
0,15	1,1	0,15	1,4	1,39734	0,00266379
0,15	1,3	0,45	1,9	1,89441	0,00559141
0,2	1,2	0,45	1,85	1,84485	0,00515285
0,2	2	0,75	2,95	2,94064	0,00935967
0,25	1,2	0,15	1,6	1,59433	0,00566706
0,25	1,7	0,25	2,2	2,1893	0,0107039

Таблица 5 – Таблица для функции v_2

x	y	t	$v_2(x, y)$	$v_{2_n}(x, y)$	Погрешность
0,05	1,1	0,05	-1	-1,0013	0,00130016
0,05	1,2	0,05	-1,1	-1,10212	0,00211839
0,1	1,1	0,05	-0,95	-0,95195	0,0019502
0,1	2	0,05	-1,85	-1,85588	0,00588195
0,1	2,9	0,1	-2,7	-2,70063	0,000633814
0,15	1,4	0,45	-0,8	-0,806601	0,00660105
0,15	1,7	0,25	-1,3	-1,30817	0,00817202
0,2	1,4	0,1	-1,1	-1,10765	0,00764799
0,25	2,4	0,3	-1,85	-1,85825	0,00825008
0,3	1,5	0,4	-0,8	-0,810169	0,0101687

Таблица 6 – Таблица для функции u

x	y	t	$u(x, y)$	$u_n(x, y)$	Погрешность
0,05	1,1	0,05	0,80539	0,811639	0,006248881
0,1	1,2	0,25	0,5624861	0,577129	0,01464292
0,15	1,4	0,05	0,8816593	0,9179273	0,03626803
0,2	1,1	0,25	0,5297697	0,542507	0,0127373
0,2	2,8	0,8	0,06628479	0,07322618	0,006941392
0,25	1,4	0,25	0,5791243	0,6190831	0,03995883
0,25	1,6	0,05	0,8763344	0,93438	0,05804565
0,3	1,5	0,1	0,780204	0,8362475	0,05604342
0,35	1,1	0,85	0,1529382	0,1614097	0,008471426
0,4	1,3	0,15	0,6574731	0,7001785	0,04270545

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы были рассмотрены и реализованы разностные схемы, используемые при решении стационарных и нестационарных задач термоупругости. В ходе работы над выпускным квалификационным проектом была исследована основная теория решения уравнений термоупругости при помощи разностных схем, а именно:

- изучены два типа задач термоупругости – стационарные и нестационарные;
- исследованы разностные схемы решения задач термоупругости;
- найден порядок аппроксимации и сходимости используемых в работе разностных схем;
- определены условия устойчивости разностных схем
- произведено математическое моделирование стационарного и нестационарного процесса термоупругости, то есть: составлены модельные задачи; построены алгоритмы решения задач;
- полученные алгоритмы реализованы на ЭВМ;
- численное решение задач было представлено в табличном и графическом виде;
- произведен анализ полученных результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Самарский, А. А. Вычислительная теплопередача / А. А. Самарский, П. Н. Вабищевич. – М. : Едиториал УРСС, 2003. – 784 с
2. Самарский, А. А. Теория разностных схем., учебное пособие, Главная редакция физико-математической литературы изд-ва «Наука», М., 1977. – 577 с.
3. Самарский, А. А. Однородные разностные схемы для нелинейных уравнений параболического типа / А. А. Самарский // Журнал вычислительной математики и математической физики. – Москва, 1962. – Т.2, № 1. – 32 с. 80
4. Крайнов, А. Ю. Численные методы решения задач тепло- и массо переноса: учебное пособие / А. Ю. Крайнов, Л. Л. Миньков. – Томск: STT, 2016.–92 с.
5. Карслоу У. Теплопроводность твердых тел/ Егер Д. – Москва, 1964. – 489 с.
6. Кутателадзе С.С. Основы теории теплообмена. М.: Атомиздат, 1979. – 416 с.
7. Минькова, Р. М. Методы математической физики: учебное пособие / Р. М. Минькова. – Екатеринбург: УрФУ, 2013. – 87 с.
8. Ландау Л. Д, Теория упругости. Лифшиц Е. М. Москва.: Наука, 1965.- 248 с.
9. Новацкий В. Теория упругости. Москва.: Мир, 1975.-872 с.
10. Арсеньев, А. А. Что такое математическая физика / А. А. Арсеньев, А. А. Самарский // Новое в жизни, науке, технике. Подписная научнопопулярная серия Математика, кибернетика. – Москва: Знание, 1983. – № 4. – 64 с.
11. Борковская, И. М. Уравнения математической физики / И. М. Борковская, О. Н. Пыжкова. – Минск, 2010. – 77 с.
12. Рогов, А. А. Уравнения математической физики. Сборник примеров и упражнений / А. А. Рогов, Е. Е. Семенова, В. И. Чернецкий, Л. В. Щеголева. – Петрозаводск: ПетрГУ, 2001. – 220 с.
13. Хватцев, И. А. Дифференциальные уравнения в частных производных: учебное пособие / А. А. Хватцев, И. А. Строчков. – Псков: Псковский государственный университет, 2016. – 80 с.

ПРИЛОЖЕНИЕ А

Листинг программы для решения стационарного уравнения термоупругости при помощи разностной схемы (обязательное)

В приложении приведен листинг программы, разработанной в среде Microsoft Visual Studio 2010 на языке C/C++, которая предназначена для нахождения решения уравнения при помощи разностной схемы.

```
#include <iostream>
#include <fstream>

#define _USE_MATH_DEFINES
#include <cmath>

using namespace std;
#define pi 3.14159265358979323846
#define N 20
#define M 20

double x[N+1], y[M+1]; // массивы хранящие сетки узлов по x и y,
double u_t[N + 1][M + 1]; // массивы хранящие значения функции u на каж-
дой итерации
double v_1[N + 1][M + 1]; // массивы хранящие значения функции v1 на
каждой итерации
double v_2[N + 1][M + 1]; // массивы хранящие значения функции v2 на
каждой итерации

const int a = 0;
const int b = 1;
const int c = 1;
const int d = 3;
const double gamma = 0.1;
const double mu = 0.2;
double lambda = (3*lambda+2*mu);

double u(double x1, double x2) // функция решения
{
    return exp(x1)*cos(x2);
}

double v1(double x1, double x2) // функция решения
{
    return x1+x2;
}

double v2(double x1, double x2) // функция решения
{
    return x1 - x2;
}
```

```

double f1(double x1, double x2)// функция решения
{
    return gamma*exp(x1)*cos(x2);
}

double f2(double x1, double x2)// функция решения
{
    return -gamma*exp(x1)*sin(x2);
}

void setka(double *m1, double a, double b, int n)
{
    double h;
    h = (double)(b - a) / n;
    for (int i = 0; i <= n; i++)    m1[i] = a + i*h;
}

void zero_approx(double(*f)(double, double), double m[N+1][M+1], double
*x, double *y)// построение матрицы начального приближения с учетом гра-
ничных условий
{
    for (int i = 0; i<N + 1; i++) // нулевое начальное приближение
        for (int j = 0; j<M + 1; j++)
            m[i][j] = 0;

    // граничные условия
    for (int i = 0; i < M + 1; i++)
        m[0][i] = f(x[0], y[i]);

    for (int i = 0; i < M + 1; i++)
        m[N][i] = f(x[N], y[i]);

    for (int i = 0; i < N + 1; i++)
        m[i][0] = f(x[i], y[0]);

    for (int i = 0; i < N + 1; i++)
        m[i][M] = f(x[i], y[M]);
}

void print(double *x, double *y, double tz[N+1][M+1], int iter)
{
    double max1 = 0, temp;
    ofstream ff1("res.txt");
    cout << "yi\\xi\\t";
    for (int i = 0; i < N + 1; i++)
        cout << x[i] << '\\t';
    cout << endl;

    for (int i = 0; i<N + 1; i++)
    {
        cout << y[i] << '\\t';
        for (int j = 0; j<M + 1; j++)
        {
            //temp = abs(u(x[i], y[j], t) - tz[i][j]);
            //if (max1<temp)max1 = temp;
            //ff1<<temp<< '\\t';

```

```

        //ff1 << x[i] << '\t' << y[j] << '\t' << tz[i][j];
        cout << tz[i][j] << '\t';
    }
    ff1<<endl;
    cout << endl;
}
//cout << "maksimalnaya pogreshnost:  " << max1 << endl;
ff1.close();
}

void pogresh(int iter)
{
    double max_u = 0, max_v1 = 0, max_v2 = 0;
    ofstream out("res.txt", ios::out | ios::app);

    for (int i = 1; i < N; i++)
    {
        double temp = 0;
        for (int j = 1; j < M; j++)
        {
            temp = abs(u(x[i], y[j]) - u_t[i][j]);
            if (max_u < temp) max_u = temp;

            temp = abs(v1(x[i], y[j]) - v_1[i][j]);
            if (max_v1 < temp) max_v1 = temp;

            temp = abs(v2(x[i], y[j]) - v_2[i][j]);
            if (max_v2 < temp) max_v2 = temp;
        }

        cout << iter << " iteration" << endl;
        cout << "maksimalnaya pogreshnost func u(x,y):  " << max_u <<
endl;
        cout << "maksimalnaya pogreshnost func v1(x,y):  " << max_v1 <<
endl;
        cout << "maksimalnaya pogreshnost func v2(x,y):  " << max_v2 <<
endl;

        out << iter << " iteration" << endl;
        out << "maksimalnaya pogreshnost func u(x,y):  " << max_u << endl;
        out << "maksimalnaya pogreshnost func v1(x,y):  " << max_v1 <<
endl;
        out << "maksimalnaya pogreshnost func v2(x,y):  " << max_v2 <<
endl<<endl;
        out.close();
    }
}

void method_Gausa_Zeidelya()
{
    double l, h;
    l = x[1] - x[2];
    h = y[1] - y[2];
    double tau = 0.01;

    double koef1;

```

```

    koefff_1 = 0.5*((1*1) / (1*1 + h*h));
    double koefff_2;
    koefff_2 = 0.5*((1*1) / (1*1 + h*h));;
    double koefff_3;
    koefff_3 = 2 * ((lambda + 2 * mu) / (h*h) + mu / (1*1));
    double koefff_4;
    koefff_4 = 2 * ((lambda + 2 * mu) / (1*1) + mu / (h*h));

    for (int i = 1; i < N; i++)
    {
        for (int j = 1; j < M; j++)
        {
            u_t[i][j] = koefff_1*(u_t[i - 1][j] + u_t[i + 1][j]) +
            koefff_2*(u_t[i][j + 1] + u_t[i][j - 1]);

            v_1[i][j] = ((lambda + 2 * mu)*(v_1[i + 1][j] + v_1[i -
            1][j]) / (h*h) + mu*(v_1[i][j + 1] + v_1[i][j - 1]) / (1*1)
            + (lambda + mu)*(v_2[i + 1][j + 1] - v_2[i][j + 1]
            - v_2[i + 1][j] + v_2[i][j]) / (h*1)
            - gamma*(u_t[i + 1][j] - u_t[i - 1][j]) / (2 * h) +
            f1(x[i], y[j])) / koefff_3;

            v_2[i][j] = (mu*(v_2[i + 1][j] + v_2[i - 1][j]) / (h*h)
            + (lambda + 2 * mu)*(v_2[i][j + 1] + v_2[i][j - 1]) / (1*1) +
            (lambda + mu)*(v_1[i + 1][j + 1] - v_1[i][j + 1] -
            v_1[i + 1][j] + v_1[i][j]) / (h*1) -
            gamma*(u_t[i][j + 1] - u_t[i][j - 1]) / (2 * 1) +
            f2(x[i], y[j])) / koefff_4;
        }
    }

}

void main()
{
    double(*f)(double, double);
    ofstream out0("res.txt");
    out0.close();

    setka(x, a, b, N);
    setka(y, c, d, M);

    f = u;
    zero_approx(f, u_t, x, y);
    f = v1;
    zero_approx(f, v_1, x, y);
    f = v2;
    zero_approx(f, v_2, x, y);

    pogresh(0);
    cout << endl;

    for (int i = 1; i < 1000; i++)
    {
        method_Gausa_Zeidelya();
    }
}

```

```
        if (i % 100 == 0) {
            pogresh(i);
            cout << endl;
        }
    }

    system("pause");
}
```

ПРИЛОЖЕНИЕ Б

Листинг программы для решения нестационарного уравнения термоупругости при помощи разностной схемы (обязательное)

В данном приложении приведен листинг программы, разработанной в среде Microsoft Visual Studio 2010 на языке C/C++, которая предназначена для нахождения решения уравнения при помощи разностной схемы.

```
#include <iostream>
#include <fstream>

#define _USE_MATH_DEFINES
#include <cmath>

using namespace std;
#define pi 3.14159265358979323846
#define N 20
#define M 20

double x[N+1], y[M+1]; // массивы хранящие сетки узлов по x и y,
double u_t[N + 1][M + 1]; // массивы хранящие значения функции u на каждой итерации
double v_1[N + 1][M + 1]; // массивы хранящие значения функции v1 на каждой итерации
double v_2[N + 1][M + 1]; // массивы хранящие значения функции v2 на каждой итерации

const int a = 0;
const int b = 1;
const int c = 1;
const int d = 3;
const double gamma = 0.1;
const double mu = 0.2;
double lambda = (3*lambda+2*mu);

double u(double x1, double x2) // функция решения
{
    return exp(x1)*cos(x2);
}

double v1(double x1, double x2) // функция решения
{
    return x1+x2;
}

double v2(double x1, double x2) // функция решения
{
    return x1 - x2;
}

double f1(double x1, double x2) // функция решения
```

```

{
    return gamma*exp(x1)*cos(x2);
}

double f2(double x1, double x2)// функция решения
{
    return -gamma*exp(x1)*sin(x2);
}

void setka(double *m1, double a, double b, int n)
{
    double h;
    h = (double)(b - a) / n;
    for (int i = 0; i <= n; i++)    m1[i] = a + i*h;
}

void zero_approx(double(*f)(double, double), double m[N+1][M+1], double
*x, double *y)// построение матрицы начального приближения с учетом гра-
ничных условий
{
    for (int i = 0; i<N + 1; i++) // нулевое начальное приближение
        for (int j = 0; j<M + 1; j++)
            m[i][j] = 0;

    // граничные условия
    for (int i = 0; i < M + 1; i++)
        m[0][i] = f(x[0], y[i]);

    for (int i = 0; i < M + 1; i++)
        m[N][i] = f(x[N], y[i]);

    for (int i = 0; i < N + 1; i++)
        m[i][0] = f(x[i], y[0]);

    for (int i = 0; i < N + 1; i++)
        m[i][M] = f(x[i], y[M]);
}

void print(double *x, double *y, double tz[N+1][M+1], int iter)
{
    double max1 = 0, temp;
    ofstream ff1("res.txt");
    cout << "yi\\xi\\t";
    for (int i = 0; i < N + 1; i++)
        cout << x[i] << '\\t';
    cout << endl;

    for (int i = 0; i<N + 1; i++)
    {
        cout << y[i] << '\\t';
        for (int j = 0; j<M + 1; j++)
        {
            //temp = abs(u(x[i], y[j], t) - tz[i][j]);
            //if (max1<temp)max1 = temp;
            //ff1<<temp<< '\\t';
            //ff1 << x[i] << '\\t' << y[j] << '\\t' << tz[i][j];
            cout << tz[i][j] << '\\t';
        }
    }
}

```

```

        }
        ff1<<endl;
        cout << endl;
    }
    //cout << "maksimalnaya pogreshnost:  " << max1 << endl;
    ff1.close();
}

void pogresh(int iter)
{
    double max_u = 0, max_v1 = 0, max_v2 = 0;
    ofstream out("res.txt", ios::out | ios::app);

    for (int i = 1; i < N; i++)
    {
        double temp = 0;
        for (int j = 1; j < M; j++)
        {
            temp = abs(u(x[i], y[j]) - u_t[i][j]);
            if (max_u < temp) max_u = temp;

            temp = abs(v1(x[i], y[j]) - v_1[i][j]);
            if (max_v1 < temp) max_v1 = temp;

            temp = abs(v2(x[i], y[j]) - v_2[i][j]);
            if (max_v2 < temp) max_v2 = temp;
        }

        cout << iter << " iteration" << endl;
        cout << "maksimalnaya pogreshnost func u(x,y):  " << max_u <<
endl;
        cout << "maksimalnaya pogreshnost func v1(x,y):  " << max_v1 <<
endl;
        cout << "maksimalnaya pogreshnost func v2(x,y):  " << max_v2 <<
endl;

        out << iter << " iteration" << endl;
        out << "maksimalnaya pogreshnost func u(x,y):  " << max_u << endl;
        out << "maksimalnaya pogreshnost func v1(x,y):  " << max_v1 <<
endl;
        out << "maksimalnaya pogreshnost func v2(x,y):  " << max_v2 <<
endl<<endl;
        out.close();
    }
}

void method_Gausa_Zeidelya()
{
    double l, h;
    l = x[1] - x[2];
    h = y[1] - y[2];
    double tau = 0.01;

    double koef1;
    koef1 = 0.5*((1*l) / (1*l + h*h));
    double koef2;

```

```

    koefff_2 = 0.5*((1*1) / (1*1 + h*h));;
    double koefff_3;
    koefff_3 = 2 * ((lambda + 2 * mu) / (h*h) + mu / (1*1));
    double koefff_4;
    koefff_4 = 2 * ((lambda + 2 * mu) / (1*1) + mu / (h*h));

    for (int i = 1; i < N; i++)
    {
        for (int j = 1; j < M; j++)
        {
            u_t[i][j] = koefff_1*(u_t[i - 1][j] + u_t[i + 1][j]) +
            koefff_2*(u_t[i][j + 1] + u_t[i][j - 1]);

            v_1[i][j] = ((lambda + 2 * mu)*(v_1[i + 1][j] + v_1[i -
            1][j]) / (h*h) + mu*(v_1[i][j + 1] + v_1[i][j - 1]) / (1*1)
            + (lambda + mu)*(v_2[i + 1][j + 1] - v_2[i][j + 1]
            - v_2[i + 1][j] + v_2[i][j]) / (h*1)
            - gamma*(u_t[i + 1][j] - u_t[i - 1][j]) / (2 * h) +
            f1(x[i], y[j])) / koefff_3;

            v_2[i][j] = (mu*(v_2[i + 1][j] + v_2[i - 1][j]) / (h*h)
            + (lambda + 2 * mu)*(v_2[i][j + 1] + v_2[i][j - 1]) / (1*1) +
            (lambda + mu)*(v_1[i + 1][j + 1] - v_1[i][j + 1] -
            v_1[i + 1][j] + v_1[i][j]) / (h*1) -
            gamma*(u_t[i][j + 1] - u_t[i][j - 1]) / (2 * 1) +
            f2(x[i], y[j])) / koefff_4;
        }
    }

}

void main()
{

    double(*f)(double, double);
    ofstream out0("res.txt");
    out0.close();

    setka(x, a, b, N);
    setka(y, c, d, M);

    f = u;
    zero_approx(f, u_t, x, y);
    f = v1;
    zero_approx(f, v_1, x, y);
    f = v2;
    zero_approx(f, v_2, x, y);

    pogresh(0);
    cout << endl;

    for (int i = 1; i < 1000; i++)
    {
        method_Gausa_Zeidelya();
        if (i % 100 == 0) {
            pogresh(i);

```

```
        cout << endl;
    }
}

system("pause");
}
```

ПРИЛОЖЕНИЕ В

Графические результаты зависимости погрешности от количества итераций для задач термоупругости (обязательное)

Для наглядности рассмотрим графические представления результатов зависимости погрешности от количества итераций. При этом были использованы те же данные погрешности, которые представлены в таблицах 1-6. Количество итераций равно тысячи.

Модельная задача №3.1:

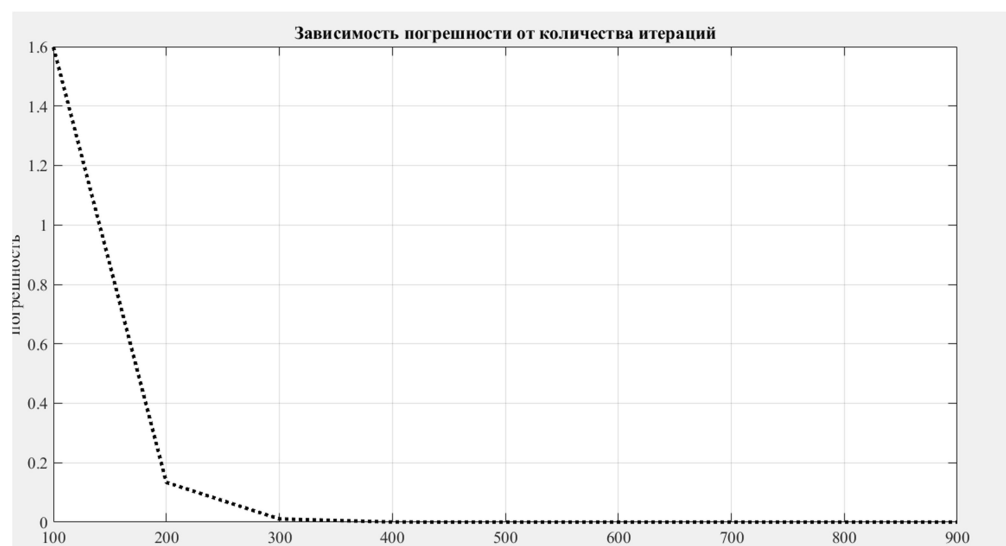


Рисунок В.1 – Зависимость погрешности от количества итераций для u

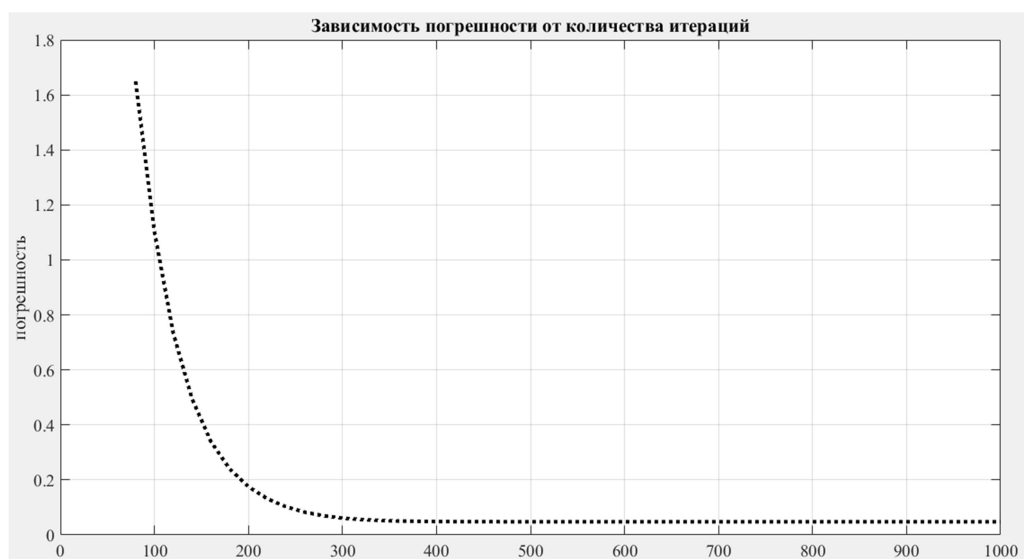


Рисунок В.2 – Зависимость погрешности от количества итераций для v_1 .

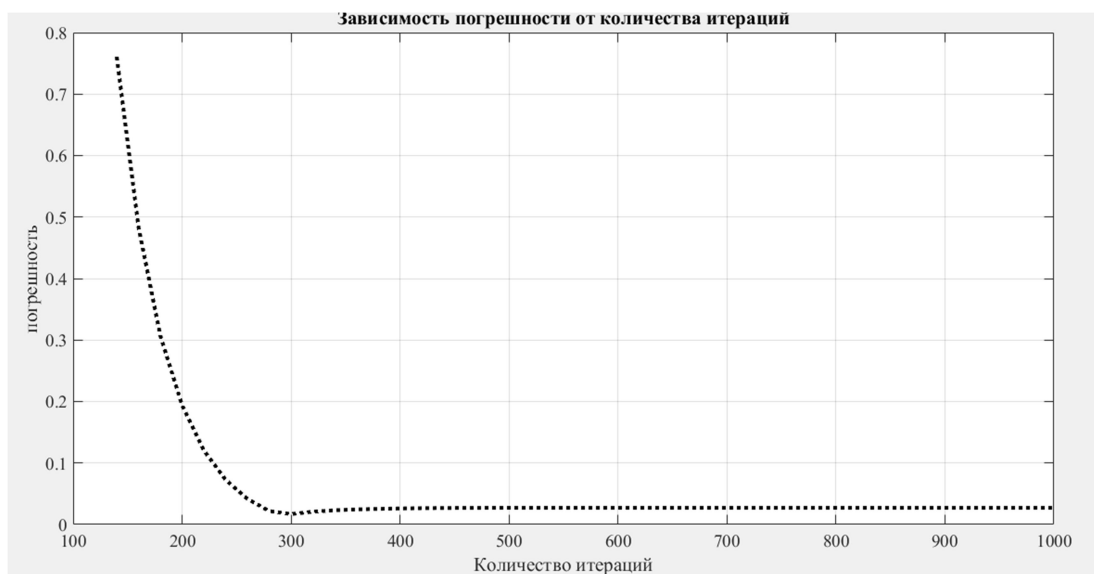


Рисунок В.3 – Зависимость погрешности от количества итераций для V_2 .

Модельная задача №3.2:

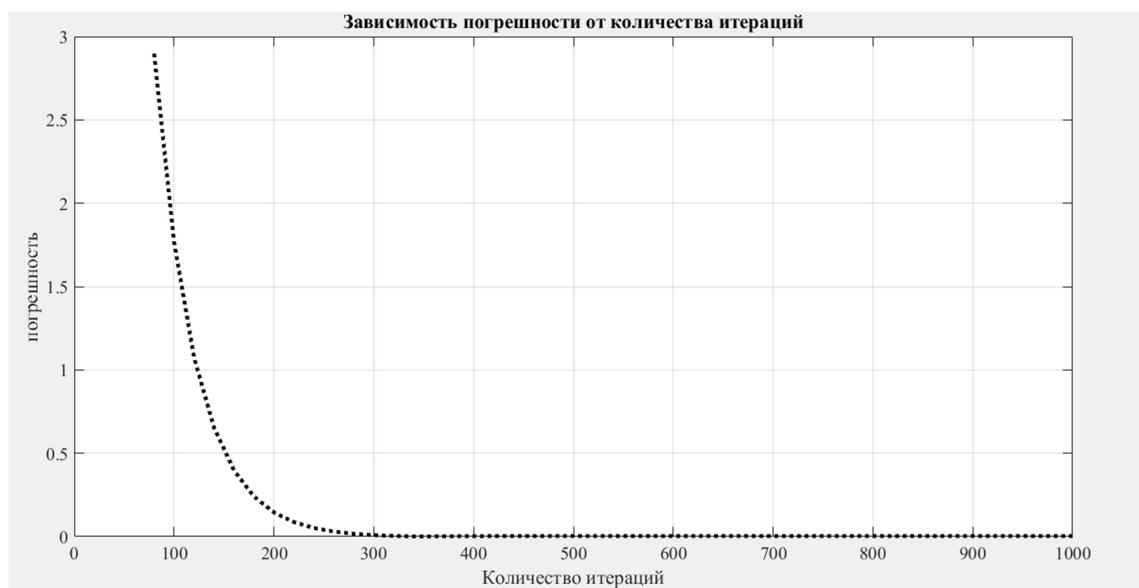


Рисунок В.4 – Зависимость погрешности от количества итераций для u .

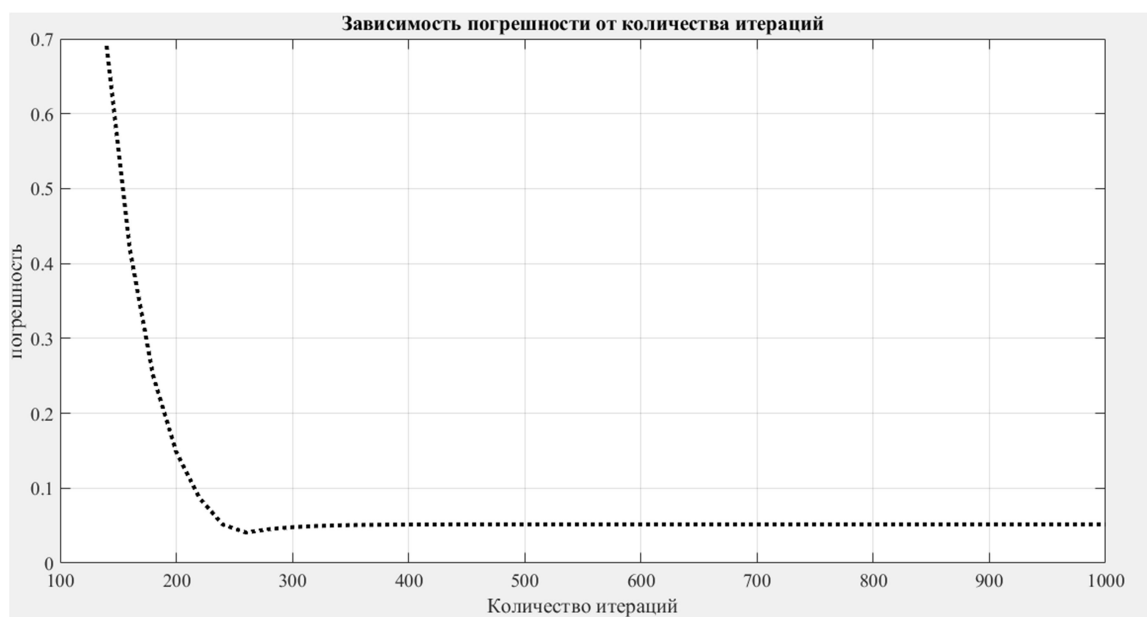


Рисунок В.5 – Зависимость погрешности от количества итераций для V_1 .

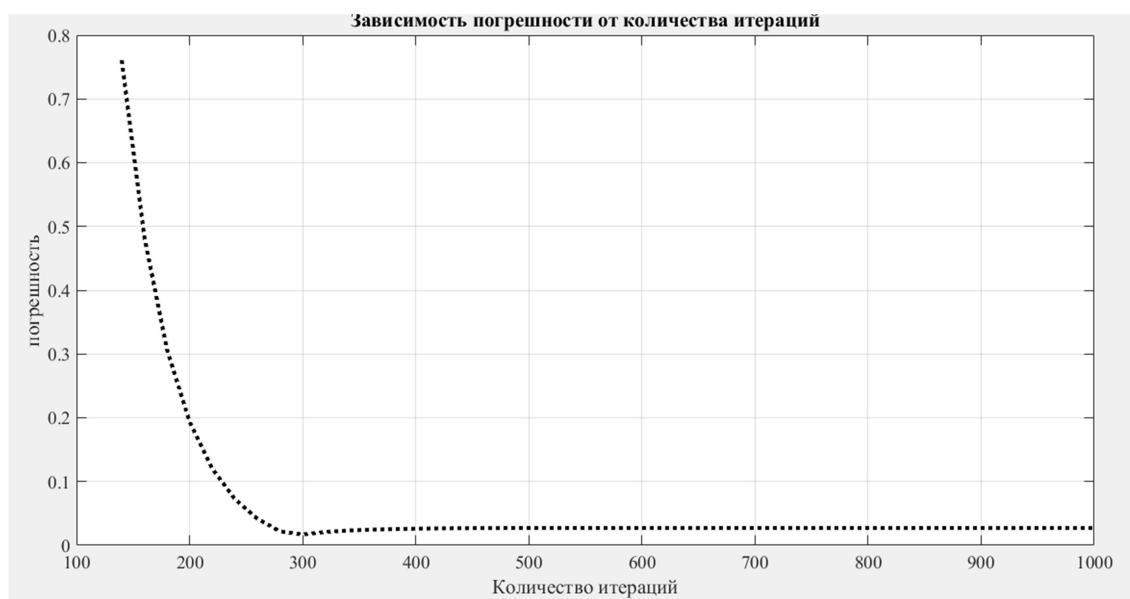


Рисунок В.6 – Зависимость погрешности от количества итераций для V_2 .