ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Факультет информатики, математики и компьютерных наук**

**Программа подготовки бакалавров по направлению**

**01.03.02 Прикладная математика и информатика**

*Торопова Екатерина Андреевна*

# КУРСОВАЯ РАБОТА

«Численное решение уравнений в частных производных
с использованием метода конечных элементов»

**Научный руководитель:**

*ученая степень, ученое звание, должность*

*Пеплин Федор Сергеевич*

**Нижний Новгород, 2024**

# Содержание

# 1. Введение

В данной работе рассматривается численное решение эллиптических уравнений с использованием метода конечных элементов. Современные инженерные и физические исследования не обходятся без точного анализа уравнений в частных производных (УЧП), которые описывают широкий спектр физических процессов. Метод конечных элементов (МКЭ) представляет собой один из ключевых численных методов, позволяющих с высокой точностью решать УЧП для различных задач. Целью данной курсовой работы является демонстрация эффективности МКЭ на примере решения двумерной и трехмерной задачи стационарной теплопроводности, являющиеся примером эллиптических дифференциальных уравнений. В работе будет подробно рассмотрен метод конечных элементов и проведено численное вычисление решения эллиптического УЧП.

**Объект исследования**: дифференциальные уравнения в частных производных второго порядка

**Предмет исследования**: метод конечных элементов

**Актуальность задачи:** В современной науке и инженерии уравнения в частных производных играют ключевую роль в моделировании физических процессов. Они описывают широкий спектр явлений, включая динамику жидкостей, электромагнетизм, теплопередачу, а также распространение звука и света. Поскольку аналитическое решение УЧП возможно лишь для весьма ограниченного класса задач, для решения данных уравнений используют численные методы. Одним из мощнейших инструментов для численного решения является метод конечных элементов, позволяющий аппроксимировать решения в сложных геометрических областях и при нестандартных граничных условиях. Благодаря своей универсальности, МКЭ нашел применение в самых разнообразных областях, от аэрокосмической инженерии до биомедицины.

Метод конечных элементов имеет ряд преимуществ перед другими численными методами: гибкость в геометрии (МКЭ позволяет точно моделировать сложные геометрии и неоднородные материалы), локальное уточнение (возможность локального уточнения сетки в областях с высоким градиентом решения без необходимости изменения всей сетки), многофункциональность (подходит для широкого спектра физических и инженерных задач, включая механику, теплопередачу и электромагнетизм).

Изучение МКЭ остается актуальным, так как это мощный и универсальный инструмент для решения уравнений в частных производных, который продолжает находить новые применения в научных и инженерных исследованиях.

**Цель:** Решение уравнения эллиптического типа на примере уравнения теплопроводности методом конечных элементов с помощью библиотеки deal.II.

**Задачи:**

1. Теоретическое изучение УЧП: рассмотрение основных видов УЧП второго порядка и методов их решения.
2. Теоретическое изучение МКЭ: рассмотрение математического аппарата, используемого для метода конечных элементов.
3. Математическое моделирование задачи теплопроводности: формулировка математической модели.
4. Реализация МКЭ для решения задачи: программирование алгоритма МКЭ и проведение численного эксперимента.

# 2. Методические основы решения дифференциальных уравнений в частных производных второго порядка с использованием МКЭ

## 2.1. Классификация линейных дифференциальных уравнений с частными производными второго порядка

Рассмотрим общий вид линейных дифференциальных уравнений с частными производными второго порядка с n переменными:

$$-\sum_{i,k=1}^{n} a_{ik}(x)u_{x_i x_k} + \sum_{i=1}^{n} b_i(x)u_{x_i} + c(x)u = f(x), \tag{2.1}$$

где коэффициенты $a_{ik}$, $b_i$, $c$ и $f$ в правой части уравнения - заданные функции независимых переменных $x_1 ... x_n$, а $u$ - искомая функция тех же аргументов. Метод решения заданного уравнения зависит от корней характеристического уравнения матрицы $A(x) := (a_{ik}(x))$:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & a_{1n} \\ a_{21} & a_{22} - \lambda & a_{2n} \\ \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{nn} - \lambda \end{vmatrix} = 0 \tag{2.2}$$

Классификация линейных дифференциальных уравнений с частными производными второго порядка проводится в соответствие знакам корней уравнения (2.2). Уравнение принадлежит эллиптическому типу, если все n корней (собственные значения) положительны, а значит матрица $A(x)$ положительно определена; к гиперболическому, если существует один отрицательный корень и $n - 1$ положительных; или к параболическому, если все собственные значения неотрицательны (матрица $A(x)$ положительно полуопределена) и ранг $(A(x), b(x))$ равен $n$.

## 2.2. Типы уравнений математической физики

Существует три основных типа уравнений в частных производных второго порядка (т.н. уравнений "математической физики"), решаемых с помощью метода конечных элементов:

1. Эллиптические уравнения

Данные уравнения, как правило, описывают стационарные процессы, где решения не изменяются со временем. Примером может служить стационарная теплопроводность, где распределение температуры в теле остается постоянным после достижения теплового равновесия.

Пусть $\Omega$ - ограниченная область в $R^n$. Тогда общий вид уравнения стационарной теплопроводности выглядит следующим образом:

$$\triangle u = 0\,, \tag{2.3}$$

где $\triangle$ - оператор Лапласа:

$$\triangle = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}$$

Уравнение (2.3) так же называется уравнением Лапласа и является частным случаем уравнения Пуассона при $f(x) = 0$. Общий вид уравнения Пуассона:

$$\triangle u = f(x) \tag{2.4}$$

2. Параболические уравнения

Данные уравнения моделируют процессы, которые изменяются со временем, но не имеют волнового характера. Так, например, параболические уравнения могут описывать нестационарную теплопроводность, где температура изменяется со временем.

Пусть $T(x,t)$ - распределение температуры в теле, где t $\in [t_0, t_{end}]$. Изменение энергии в элементе объема определяется тепловым потоком через

поверхность тела, а также источником тепла внутри тела Q. Тогда можно построить уравнение теплопроводности:

$$T_t = \sigma \triangle T + Q \tag{2.5}$$

3. Гиперболические уравнения

Гиперболические уравнения описывают волновые процессы и динамические системы. Примерами могут быть колебание струны в одномерном случае, колебание мембраны в двумерном случае и распространение звуковых колебаний в трехмерном случае. Классический пример - волновое уравнение распределения давления газа.

Пусть $p(t, x)$ - распределение давления газа. В случае постоянной температуры движение идеального газа описывается следующим уравнением:

$$p_{tt} = c^2 \triangle p \tag{2.6}$$

при задании следующих начальных условий:

$$p(x, 0) = f(x)$$

$$p_t(x, 0) = g(x)$$

Метод конечных элементов позволяет решать уравнения математической физики с высокой точностью, что особенно важно для инженерных задач. Именно поэтому МКЭ является незаменимым инструментом в проектировании систем, подверженных различным тепловым и механическим нагрузкам.

## 2.3. Основы МКЭ на примере одномерных задач

Изучим математические основы МКЭ в рассмотрении одномерных линейных дифференциальных уравнений эллиптического типа. К таким урав-

нениям можно отнести одномерные теплопроводность, стационарную диффузию и упругость.

### 2.3.1. Граничные условия

Граничные условия определяют поведение уравнений в частных производных на границах рассматриваемой области. Два основных типа граничных условий - это условия Дирихле и Неймана.

Граничные условия Дирихле, являющиеся граничными условиями первого рода, задают точные значения искомой функции на границе области. Так, например, при решении задачи теплопроводности, условия Дирихле могут задавать температуру на поверхности объекта. Введем условные обозначения:

$$u(x) = u_0, x \in \partial\Omega \tag{2.7}$$

где $u(x)$ - искомая функция, $u_0$ - заданное значение на границе $\partial\Omega$

Граничные условия Неймана, являющиеся условиями второго рода, задают производную искомой функции по нормали к границе области. В контексте той же задачи теплопроводности, условия Неймана могут определять тепловой поток через границу. Условие Неймана можно записать следующим образом:

$$\frac{\partial u}{\partial n} = g(x), x \in \partial\Omega \tag{2.8}$$

где $\frac{\partial u}{\partial n}$ - производная по нормали, а $g(x)$ - заданная функция на границе $\partial\Omega$
В случае теплопроводности, $g(x)$ - тепловой поток.

### 2.3.2. Сильная постановка задачи

Для метода конечных элементов существует два типа формулировок задач: сильная и слабая. Сильная постановка задачи в МКЭ - это прямое применение дифференциальных уравнений, которое часто бывает сложно

решить из-за непрерывности и дифференцируемости требований. Сильная формулировка требует найти определенную функцию, удовлетворяющую некоторому дифференциальному уравнению в частных производных, чтобы были соблюдены заданные граничные условия.

Слабая же постановка, также известная как вариационная постановка, задает искомую функцию в уравнении в интегральном виде, что обычно приводит к решению системы алгебраических уравнений. Это позволяет использовать функции, которые могут не быть непрерывно дифференцируемыми, упрощая решение и повышая его точность. Именно поэтому слабая постановка является предпочтительным выбором для численного решения задач.

Рассмотрим сильную постановку задачи одномерной линейной упругости. Возьмем в рассмотрение уравнение, описывающее смещение упругого стрежня под напряжением:

$$u_{,xx} + f = 0 \tag{2.9}$$

Предполагается, что $f$ - гладкая функция, $f : [0, L] \to \mathbb{R}$ . Обозначим граничные условия для нашей задачи: $u(L) = u_g; -u_{,x}(0) = j_n$ . Тогда сильная постановка для нашей задачи будет выглядеть следующим образом:

$$\begin{cases} \text{По данным } f : [0, L] \to \mathbb{R}, u_g, j_n, \text{ найти такую функцию} \\ u : [0, L] \to \mathbb{R}, \text{ что} \\ u_{,xx} + f = 0 \text{ на } [0, L] \\ u(L) = u_g \\ -u_{,x}(0) = j_n \end{cases} \tag{2.10}$$

Требуется, чтобы $u(x)$ была гладкой, поскольку заданное УЧП содержит две частных производных, а также необходимо гарантированное поточечное вычисление УЧП в интервале $(0, L)$.

Точное решение для нашей задачи будет выглядеть следующим обра-

зом:

$$u(x) = \int_x^1 (\int_0^y f(z)dz)dy + q + (1-x)k \tag{2.11}$$

### 2.3.3. Слабая постановка задачи

Для постановки задачи в слабой форме введем два класса функций: класс пробных решений $S$ и множество весовых функций $V$. Предполагаем, что функции класса $S$ удовлетворяют граничному условию Дирихле $u(L) = u_g$ и для них верно следующее:

$$\int_0^1 (u_{,x})^2 dx < \infty \tag{2.12}$$

То есть, $u \in H^1([0,1])$, где $H^1$ - пример Гильбертова пространства, включающее себя квадратично интегрируемые функции, чьи первые производные тоже квадратично интегрируемы. В контексте МКЭ, пространство $H^1$ часто используется для определения пространства приближенных решений, особенно для задач упругости и теплопроводности.

Обозначим множество пробных решений и множество весовых функций:

$$S = \{u \in H^1([0,1])|\ u(1) = q\}$$

$$V = \{w \in H^1([0,1])|\ w(1) = 0\}$$

Принимая во внимание введенные нами функции, рассмотрим слабую постановку задачи:

$$\begin{cases} \text{По данным } f : [0, L] \to \mathbb{R}, u_g, j_n, \text{ найти такую} \\ u \in S, \text{ что } \forall w \in V \\ \int\limits_0^1 u_{,x} w_{,x} dx = \int\limits_0^1 w f dx + w(0) j_n \end{cases} \tag{2.13}$$

Для метода метода конечных элементов характерно использование сла-

бой формы постановки задач. Для краткости часто используются следующие сокращения:

$$a(u,w) = \int_0^1 u_{,x} w_{,x} dx \tag{2.14}$$

$$(u,w) = \int_0^1 uw dx$$

Тогда уравнение в слабой форме можно переписать в виде

$$a(w,u) = (w,f) + w(0)k \tag{2.15}$$

Стоит отметить, что приведенные постановки задач эквивалентны друг другу.

### 2.3.4. Метод приближений Галеркина

Метод Галеркина — это численный метод для решения дифференциальных уравнений, который использует систему взвешенных остатков. В этом методе тестовые функции выбираются таким образом, чтобы они совпадали с функциями базиса, используемыми для приближения решения. Это означает, что если решение приближается с помощью некоторого набора функций $\varphi_i$, то те же самые функции $\varphi_i$ будут использоваться в качестве тестовых функций.

Построим конечномерные аппроксимации пространств $V, S$ и обозначим их как $S^h \subset S$ и $V^h \subset V$, где $h$ - размер конечного элемента (диаметр). Считаем, что $u^h(L) = u_g, w^h(L) = 0$. Рассмотрим метод Галеркина-Петрова, являющийся вариацией метода Галеркина.

Метод Галеркина-Петрова — это вариация метода Галеркина, в которой тестовые функции отличаются от функций базиса. В этом методе тестовые функции $\varphi_i$ выбираются так, чтобы они были не обязательно равны базисным функциям $\xi_i$ , используемым для приближения решения. Это позволяет

улучшить стабильность и сходимость численного решения за счет более гибкого выбора тестовых функций.

Итак, пусть $V^h$ дано. Тогда $\forall v^h \in V^h$ рассмотрим $u^h \in S^h$:

$$u^h = v^h + q^h \tag{2.16}$$

Здесь $q^h$ - некоторая функция, удовлетворяющая $q^h(L) = u_g$. Запишем теперь уравнение из слабой формулировки в терминах $u^h \in S^h, w^h \in V^h$:

$$a(w^h, u^h) = (w^h, f) + w^h(0)k \tag{2.17}$$

Подставим определение $u^h \in S^h$ из (2.16) в (2.17) и получим:

$$a(w^h, v^h) = (w^h, f) + w^h(0)k - a(w^h, q^h) \tag{2.18}$$

Пользуясь результатом, сформулируем постановку задачи Галеркина-Петрова:

$$\begin{cases} \text{По данным } f : [0, L] \to \mathbb{R}, u_g, j_n, \text{ найти такую} \\ u^h = v^h + q^h, v^h \in V^h, \text{ что } \forall w^h \in V^h \\ a(w^h, v^h) = (w^h, f) + w^h(0)j_n - a(w^h, q^h) \end{cases} \tag{2.19}$$

### 2.3.5. Матричная постановка задачи Галеркина-Бубнова

Метод Галеркина предполагает решение системы линейных уравнений. Рассмотрим систему более подробно.

Пусть $V^h$ состоит из линейных комбинаций семейства функций $N_A : \omega \to \mathbb{R}, A = 1, ..n$, которые называются базисными, или интерполяционными. То есть для $w \in V$ :

$$w^h = \sum_{A=1}^{n} c_A N_A \tag{2.20}$$

Требуем, чтобы $N_A(1) = 0, A = 1, ...n$. Зададим $q^h$ для более полного определения $S^h$: $q^h = qN_{n+1}$, где $N_{n+1} : \omega \to \mathbb{R}, N_{n+1}(1) = 1$ - еще одна базисная функция. Тогда для $u^h \in S^h$ получим:

$$u^h = v^h + q^h = \sum_{A=1}^{n} N_A d_A + qN_{n+1} \tag{2.21}$$

Подставив данный результат в уравнение Галеркина, получим

$$a(\sum_{A=1}^{n} c_A N_A, \sum_{s=1}^{n} d_s N_s) = (\sum_{A+1}^{n} c_A N_A, f) + \sum_{A=1}^{n} c_A N_A(0)k - a(\sum_{A=1}^{n} c_A N_A, qN_{n+1}) \tag{2.22}$$

Пусть

$$G_A = \sum_{s=1}^{n} a(N_A, N_S)d_s - (N_A, f) - N_A(0)k + a(N_A, N_{n-1})q \tag{2.23}$$

тогда в силу билинейности операций

$$\sum_{A=1}^{n} c_A G_A = 0 \tag{2.24}$$

Так как $c_A$ - произвольные константы, то $G_A = 0, A = 1, ..n$. Значит, из (2.23) мы получим

$$\sum_{s=1}^{n} a(N_A, N_S)d_s = (N_A, f) + N_A(0)k - a(N_A, N_{n-1})q \text{ для } A = 1, ..n \tag{2.25}$$

Полученное тождество - система из $n$ линейных уравнений с $n$ неизвестными $d_s$. Положим

$$K_{AB} = a(N_A, N_B), \text{тогда } K = \sum_{B=1}^{n} K_{AB}, \tag{2.26}$$

$$F_A = (N_A, f) + N_A(0)k - a(N_A, N_{n+1})q, \text{тогда } F = \sum_{A=1}^{n} F_A. \tag{2.27}$$

Из (2.25), (2.26) и (2.27) наша система примет вид

$$Kd = F \tag{2.28}$$

В (2.28) $d$ - искомый вектор смещений, а $K$ и $F$ - заданы. Матрица $K$ также называется матрицей жесткости, а вектор $F$ - вектор сил. Данное уравнение представляет собой матричное представление задачи Галеркина-Бубнова.

Подробнее рассмотрим матрицу жесткости. Из определения $a(u, w)$ в (2.14) получаем, что

$$K_{AB} = \int_0^1 N_{A,x} N_{B,x} dx \tag{2.29}$$

При $B > A + 1$ элемент матрицы $K_{AB}$ обращается в 0, а значит, достаточно большое количество элементов матрица $K$ - нулевые. На самом деле, матрица $K$ имеет трехдиагональный вид, и, кроме того, является положительно определенной.

### 2.3.6. Пространство конечных элементов. Глобальное и локальное описания элементов

Построим $n$ мерное пространство $V^h$. Разобьем отрезок $[0, 1]$ на $n$ интервалов $[x_A, x_{A+1}]$ длиной $h_A = x_{A+1} - x_A$. Пусть параметр разбиения $h = max(x_{A+1} - x_A)$ для $A = 1, ..n - 1$

Для метода конечных элементов характерно локальное и глобальное описания элемента, связь между которыми устанавливается во время сборки глобальной матрицы жесткости системы, где локальные матрицы жесткости отдельных элементов объединяются для формирования общей системы уравнений, которая описывает поведение всей конструкции. Локальное описание элемента можно рассматривать как поведение и свойства в рамках са-

мого элемента, например, распределение напряжений или деформаций внутри элемента. Глобальное описание же связано с поведением всей конструкции или системы в целом, куда входит данный элемент. Рассмотрим математические обозначения, часто использующиеся для описаний конечных элементов.

| Глобальное описание элемента | Локальное описание элемента |
|---|---|
| 1. Область определения $[x_A, x_{A+1}]$<br>2. Узлы $\{x_A, x_{A+1}\}$<br>3. Степени свободы $\{d_A, d_{A+1}\}$<br>4. Базисные функции $\{N_A, N_{A+1}\}$<br>5. Вид интерполирующей функции<br>$u^h = d_A N_A(x) + d_{A+1} N_{A+1}(x)$,<br>$x \in [x_A, x_{A+1}]$ | 1. Область определения $[\xi_1, \xi_2]$<br>2. Узлы $\{\xi_A, \xi_{A+1}\}$<br>3. Степени свободы $\{d_A, d_{A+1}\}$<br>4. Базисные функции $\{N_1, N_2\}$<br>5. Вид интерполирующей функции<br>$u^h = d_A N_1(x) + d_{A+1} N_2(x)$,<br>$x \in [\xi_1, \xi_2]$ |

Таблица 2.1

Для аппроксимации неизвестной функции используются интерполяционные многочлены Лагранжа внутри каждого конечного элемента. Такие многочлены так же называют базисными функциями. Общий вид:

$$N^A(\xi_1) = \sum_{B=1, B\neq A}^{n_{n1D}} \frac{\xi_1 + \xi_1^A}{\xi_1^A - \xi_1^B} \tag{2.30}$$

## 2.4. Постановка задачи стационарной теплопроводности

### 2.4.1. Сильная и слабая формулировки

Пусть $j_i$ - компоненты вектора потока тепла, $u$ - температура, $f$ - приток тепла в единичном объеме. Определим вектор теплового потока следующим образом:

$$j_i = -K_{ij} u_{,j} \tag{2.31}$$

, где $K$ - компоненты тензора теплопередачи и $K_{ij} = K_{ji}$ являются некоторыми известными функциями от $x$. Тензор теплопередачи может быть представлен в виде матрицы, если свойства материала зависят от направления мате-

риала (то есть, материал анизотропен). Если же материал изотропен, т.е. его свойства одинаковы во всех направлениях, тензор теплопроводности упрощается до скаляра. Считается, что тело является однородным, если коэффициенты $K_{ij}$ постоянны. Также предполагается, что матрица $K$ положительно определена.

Рассмотрим сильную постановку задачи для задачи теплопроводности:

$$\begin{cases} \text{По данным } f(x), u_g, j_n, \quad j_i = -K_{ij} \cdot u_{,j}, (i,j = 1,2,3) \\ \text{ найти такую функцию } u, \text{ что} \\ -j_{i,i}\big|_\Omega = f \\ u\big|_{\partial\Omega_u} = u_g \\ -j \cdot n\big|_{\partial\Omega_j} = j_n \end{cases} \tag{2.32}$$

В (2.32) функция $u_g$ устанавливает температурные граничные условия (что фактически накладывается граничным условием Дирихле), а $j_n$ - на поток тепла, заданный через границу области (накладывается условием Неймана). $\partial\Omega_u$ и $\partial\Omega_j$ - открытые подмножества кусочно-гладкой границы $\partial\Omega$ для области $\Omega$. Здесь же рассмотрим и слабую конечно-мерную постановку задачи:

Определим пространство $S^h \in V^h$ как конечное подмножество открытых подпространств $\Omega^e, e = 1, ..n_{el}$ пространства $\Omega$, тогда

$$\begin{cases} \text{По данным } f, u_g, j_n, \quad j_i = -K_{ij} \cdot u_{,j}, \\ \text{ найти такие функции } u^h \in S^h = \{u^h \in H^1(\Omega) |\ u^h\big|_{\partial\Omega_u} = u_g\}, \text{ что} \\ \forall w^h \in V^h \subset V, \quad V^h = \{w^h \in H_1(\Omega) |\ w^h\big|_{\partial\Omega_u} = 0\} \\ \int\limits_\Omega w^h_{,i} j^n_i dV = \int\limits_\Omega w^h f dV - \int\limits_{\partial\Omega_j} w^h j_n dS \end{cases} \tag{2.33}$$

### 2.4.2. Гексаэдральный элемент

Конечные элементы могут иметь различные геометрические формы, в зависимости от требований точности моделирования и анализа конкретной задачи. Так, для двумерных задач часто используются треугольные или квадратичные элементы, а для трехмерных тетраэдральные и гексаэдральные. Рассмотрим гексаэдральные конечные элементы, они же восьмиузловые элементы, которыми мы в дальнешем будет пользоваться.

В контексте пространства гексаэдрального элемента $\Omega^e$ мы рассматриваем трилинейные базисные функции и узлы $x_e^A,$ где $A = 1, ..., 8$ - локальная нумерация каждого отдельного элемента. Рассмотрим вид $u^h$ и $w^h$ для элемента $e$:

$$u_e^h = \sum_{A=1}^{n_{n_{el}}=8} N^A(x) d_e^A \qquad w_e^h = \sum_{A=1}^{n_{n_{el}}=8} N^A(x) c_e^A \tag{2.34}$$

Рассмотрим отображение пространства локального элемента $\Omega_\xi$ в пространство глобального $\Omega_e$, т.е. такое $x(\xi) : \Omega_\xi \to \Omega_e$ (см. рис. 2.1). Тогда

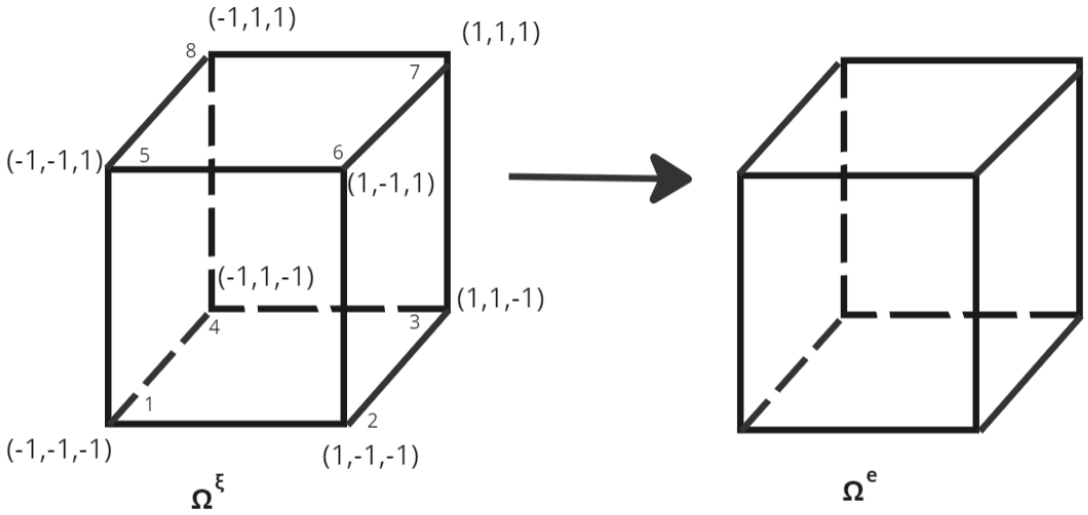



Рис. 2.1. Отображение $x(\xi) : \Omega_\xi \to \Omega_e$

базисные функции могут быть записаны в виде:

$$N^A(\xi, \eta, \zeta) = \frac{1}{8}(1 + \xi\xi^A)(1 + \eta\eta^A)(1 + \zeta\zeta^A) \tag{2.35}$$

В представленной формуле $A$ - локальный номер узла, а $\xi_A, \eta_A, \zeta_A$ - координаты узла $x^A$. Так, например, $N^1(\xi,\eta,\zeta) = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta)$, а $N^8(\xi,\eta,\zeta) = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)$ (см. рис. 2.1).

### 2.4.3. Выведение матрично-векторной постановки

Вернемся к слабой постановке задачи стационарной теплопроводности, обратив внимание на то, что $\overline{\Omega} = \overline{\bigcup \Omega^e},\ \forall e$:

$$\sum_e \int\limits_\Omega w^h_{,i} \cdot j^n_i dV = \sum_e \int\limits_\Omega w^h f dV - \sum_{e\in\varepsilon_N} \int\limits_{\partial\Omega_j} w^h j_n dS, \tag{2.36}$$

или, учитывая, что$j_i = -K_{ij} \cdot u^h_{,j}$ :

$$\sum_e \int\limits_\Omega w^h_{,i} \cdot (-K_{ij} \cdot u^h_{,j}) dV = \sum_e \int\limits_\Omega w^h f dV - \sum_{e\in\varepsilon_N} \int\limits_{\partial\Omega_j} w^h j_n dS, \tag{2.37}$$

где $e \in \varepsilon_N$, если $\partial\Omega^e \bigcap \partial\Omega_j \neq \emptyset$, то есть если граница элемента $e$ не пересекает границу области, на которой задано граничное условие Неймана. Итак, рассматривая нашу интегральную постановку задачи, нам необходимо понимать, как вычислять градиенты $w^h_{e,i}$ и $u^h_{e,i}$:

$$u^h_{e,i} = \sum_{A=1}^{n_{n_{el}}} N^A_{,i}(x) d^A_e \qquad w^h_{e.i} = \sum_{A=1}^{n_{n_{el}}} N^A_{,i}(x) c^A_e \tag{2.38}$$

Рассматривая компоненты $\xi_I$ вектора $\xi \in \Omega^\xi, \xi = \{\xi,\eta,\zeta\}^T$, найдем $N^A_{,i}$:

$$N^A_{,i} = \frac{\partial N^A}{\partial x_i} = \frac{\partial N^A}{\partial \xi_I} \cdot \frac{\partial \xi_I}{\partial x_i} \tag{2.39}$$

Обратимся к отображению:

$$x_i(\xi) = \sum_{A=1}^{n_{el}} N^A(\xi) \cdot x^A_{e_i} \quad \Longrightarrow$$

$$J_{ix} = \frac{\partial x_i}{\partial \xi_I} = \sum_{A=1}^{n_{el}} N^A_{,I}(\xi) \cdot x^A_{e_i}, \tag{2.40}$$

где $J_{ix}$ - элементы якобиана $J := \frac{\partial x}{\partial \xi}$, являющийся тензором. В механике сплошных сред якобиан также используется для описания относительного изменения объема элемента сплошной среды при деформации. В матричном виде:

$$J = \begin{pmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_1}{\partial \xi_3} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \\ \frac{\partial x_3}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_3} \end{pmatrix} \quad \Rightarrow \quad \exists \quad J^{-1} = \begin{pmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} & \frac{\partial \xi_1}{\partial x_3} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} & \frac{\partial \xi_2}{\partial x_3} \\ \frac{\partial \xi_3}{\partial x_1} & \frac{\partial \xi_3}{\partial x_2} & \frac{\partial \xi_3}{\partial x_3} \end{pmatrix}$$

Далее, понимая природу $w^h_{,i}$ и $u^h_{,i}$, вновь обратимся к интегральной постановке:

$$\int\limits_{\Omega} w^h_{,i} \cdot j^n_i dV = -\int\limits_{\Omega^e} w^h_{,i} \cdot K_{ij} \cdot u^h_{,j} dV = -\int\limits_{\Omega^e} (\sum_{A=1}^{n_{el}} N^A_{,i} c^A_e) \cdot K_{ij} \cdot (\sum_{B=1}^{n_{el}} N^B_{,J} d^B_e) dV \tag{2.41}$$

Подставим (2.39) под интеграл и воспользуемся $dV = det(J(\xi)) \cdot dV^{\xi}$:

$$-\int\limits_{\Omega^e} (\sum_{A=1}^{n_{el}} N^A_{,I} \xi_{I,i} c^A_e) \cdot K_{ij} \cdot (\sum_{B=1}^{n_{el}} N^b_{,J} \xi_{J,j} d^B_e) dV =$$

$$= -\sum_{A,B} c^A_e \Bigg( \int\limits_{\xi_1=-1}^{1} \int\limits_{\xi_2=-1}^{1} \int\limits_{\xi_3=-1}^{1} N^A_{,I} \xi_{I,i} \cdot K_{ij} \cdot N^b_{,J} \xi_{J,j} \cdot det[J(\xi)] d\xi_1 d\xi_2 d\xi_3 \Bigg) d^B_e =$$

$$= -\sum_{A,B} c^A_e \cdot \kappa^{AB}_e d^B_e = - < c^1_e ... c^{n_{el}}_e > \cdot \begin{pmatrix} \kappa^{11}_e & \dots & \kappa^{1n_{nel}}_e \\ \dots & \dots & \dots \\ \dots & \dots & \kappa^{n_{nel}n_{nel}}_e \end{pmatrix} \cdot \begin{pmatrix} d^1_e \\ \dots \\ \dots \\ d^{n_{el}}_e \end{pmatrix} =$$

$$= -c^T_e \kappa_e d_e \tag{2.42}$$

Похожим образом преобразуем остальные интегралы из слабой конеч-

номерной формулировки:

$$\int\limits_{\Omega^e} w^h f dV = \int\limits_{\Omega^e} (\sum_A N^A c_e^A) f dV = \sum_A c_e^A \int\limits_{\Omega^\xi} N^A f(x(\xi)) \cdot det[J(\xi)] dV^\xi =$$

$$=< c_e^1 ... c_e^{n_{el}} > \cdot \int\limits_{\xi_1=-1}^{1} \int\limits_{\xi_2=-1}^{1} \int\limits_{\xi_3=-1}^{1} \begin{pmatrix} N^1 \\ \dots \\ N_{n_{el}} \end{pmatrix} f(\xi) \cdot det[I(\xi)] \cdot d\xi_1 \cdot d\xi_2 \cdot d\xi_{3=}$$

$$=< c_e^1 ... c_e^{n_{el}} > \cdot \begin{pmatrix} F_e^{int_1} \\ \dots \\ F_e^{int_{n_{el}}} \end{pmatrix} = c_e^T \cdot F_e^{int} \tag{2.43}$$

Для преобразования последнего интеграла введем $\mathscr{A}_N = \{A | x_e^A \in \partial\Omega_j^e\}$:

$$-\int\limits_{\partial\Omega_j^e} w^h j_n dS = -\int\limits_{\partial\Omega_j^e} (\sum_{A=1}^{n_{el}} N^A c_e^A) j_n dS = -\sum_{A\in\mathscr{A}_j} \int\limits_{\partial\Omega_j^\xi} N^A j_n det[J_s] dS^\xi =$$

$$= -\sum_{A\in\mathscr{A}_j} \int\limits_{\xi_{i=-1}}^{1} \int\limits_{\xi_{j=-1}}^{1} N^A j_n det[J_S] d\xi_i d\xi_j =< c_e^{A_1} ... c_e^{A_4} > \cdot \begin{pmatrix} F^{j_{A_1}} \\ \dots \\ F^{j_{A_4}} \end{pmatrix} = -c_e^T F_e^j \tag{2.44}$$

В интегральной постановке из (2.41), (2.42) и (2.43) получаем:

$$\sum_e c_e^T \kappa_e d_e = -\sum_e^{n_{el}} c_e^T F_e^{int} + \sum_{e\in\xi_N} c_e^T F_e^j \tag{2.45}$$

Получаем матрично-векторную постановку:

$$c^T K \vec{d} = c^T F^{int} + c_e^T F^j \tag{2.46}$$

### 2.4.4. Численное интегрирование

Для определения матриц жесткости и векторов нагрузок необходимо вычислять интегралы на каждом конечном элементе. При таких условиях получение точного аналитического решения является затруднительным, поэтому часто используют численные методы интегрирования для их вычисления. К таким численным методам относят квадратурные формулы, которыми мы и будем пользоваться.

Чтобы вычислить некоторый определенный интеграл в приближении, его заменяют конечной суммой:

$$\int_{-1}^{1} g(\xi)d\xi = \sum_{l=1}^{n_{int}} g(\xi_l)w_l \tag{2.47}$$

Здесь $n_{int}$ - количество точек интегрирования, $\xi_l$ - значение узла, а $w_l$ - вес, приписываемый узлу. $\xi_l$ и $w_l$ задаются в зависимости от квадратурного правила, которое мы используем. Упростим интегрирование для двумерной задачи:

$$\int_{\xi_2=-1}^{1}\int_{\xi_1=-1}^{1} g(\xi_1,\xi_2)d\xi_1 d\xi_2 = \sum_{l_2=1}^{n_{int}^2}\sum_{l_1=1}^{n_{int}^3} g(\xi_1^{l_1},\xi_2^{l_2})\cdot w_{l_1}w_{l_2} \tag{2.48}$$

И для трехмерной:

$$\int_{\xi_3=-1}^{1}\int_{\xi_2=-1}^{1}\int_{\xi_1=-1}^{1} g(\xi_1,\xi_2,\xi_3)d\xi_1 d\xi_2 d\xi_3 = \sum_{l_3=1}^{n_{int}^3}\sum_{l_2=1}^{n_{int}^2}\sum_{l_1=1}^{n_{int}^3} g(\xi_1^{l_1},\xi_2^{l_2},\xi_3^{l_3})\cdot w_{l_1}w_{l_2}w_{l_3} \tag{2.49}$$

Численные методы интегрирования различаются видами используемых аппроксимирующих функций. Так, существуют методы прямоугольников, трапеций, парабол (т.н. метод Симпсона). В сравнении с перечис-

ленными методами, метод Гаусса имеет более высокий порядок точности интегрирования полиномиальной функции засчет подбора узлов и их весов. Значения узлов метода Гаусса по $n$ точкам подбираются с помощью корней полинома Лежандра степени $n$. Далее рассмотрим правило квадратуры Гаусса по 3 и 4 точкам:

$$\int_{-1}^{1} f(t)dt = \frac{5}{9}f(-\sqrt{\frac{3}{5}}) + \frac{8}{9}f(0) + \frac{5}{9}f(-\sqrt{\frac{3}{5}}) \tag{2.50}$$

$$\int_{-1}^{1} f(t)dt = \frac{18+\sqrt{30}}{36}f(-\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}) + \frac{18+\sqrt{30}}{36}f(\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}) +$$

$$+\frac{18-\sqrt{30}}{36}f(-\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}) + \frac{18-\sqrt{30}}{36}f(\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}) \tag{2.51}$$

# 3. Численное решение эллиптического уравнения на примере задачи стационарной теплопроводности методом конечных элементов с помощью библиотеки deal.II

В данном разделе рассмотрим реализацию численного расчета решения уравнения теплопроводности в стационарном состоянии с помощью библиотеки с открытым исходным кодом deal.II. В частности, рассмотрим двумерную и трехмерную постановку задач. Полную реализацию решения можно посмотреть, перейдя по ссылке https://github.com/prodoInaya/fem.

Сформулируем точные условия для нашей задачи:

$$\text{УЧП} \quad -\nabla \cdot j = f$$

Условие для теплового потока $j = -K \cdot \nabla u$

Граничное условие Неймана $-j \cdot n = h$ на $\partial\Omega_j$

Граничное условие Дирихле $u = g$ на $\partial\Omega_u$

В качестве начальных условий для обеих задач

$$K = 385 \text{ Вт} \cdot \text{м}^{-1} \cdot \text{К}^{-1}$$

Конкретизируем начальные условия для двух постановок:

1) Для двумерной сетки возьмем $15 \times 40, x \in [0, 0.03], y \in [0, 0.08]$. Положим $u(x) = 300(1 + \frac{1}{3}K.m^{-1} \cdot x)$ вдоль $y = 0$ для нижнего узла и $u(x) = 310(1 + 8K.m^{-2} \cdot x^2)$ вдоль $y = 0.08$ для верхнего узла.

2) Для трехмерной задачи рассмотрим сетку $8 \times 16 \times 4, x \in [0, 0.04], y \in [0, 0.08], z \in [0, 0.02]$. Положим $u(y, z) = 300(1 + \frac{1}{3}K.m^{-1} \cdot (y + z))$ вдоль $x = 0$ для левого узла и $u(y, z) = 310(1 + 8K.m^{-2} \cdot (y + z))$ вдоль $x = 0.04$ для правого узла.

Разделим решения для двумерной и трехмерной постановки задач на два раздела: CA2a для двумерной и CA2b для трехмерной сетки. Рассмотрим план работы программы для решения задач теплопроводности для двух постановок:

В файле main2a.cc или main2b.cc задаем размерность задачи в константе $dimension$, задающая размерность сетки с помощью вектора $num\_of\_elements$, чья размерность равна значению $dimension$. Далее создаем объект класса $FEM$, для которого

1. Генерируем сетку в функции $generate\_mesh$

2. С помощью функции $setup\_system$ заполняем таблицу $nodeLocations$ координатами для каждого глобального номера узла, задаем граничные условие через $define\_boundary\_coords$, задаем размеры глобальных матриц $K$ и $F$, искомого вектора $D$, а также задаем квадратурное правило, которым мы будем пользоваться (будем пользоваться квадратурой Гаусса для 3 и 4 точек).

3. Обращаемся к функции $assemble\_system$, в которой происходит сборка $Flocal$ и $Klocal$ в глобальную матрицу жесткости и глобальный вектор воздействия.

4. Получаем искомый вектор решения $D$ в функции $solve$ с помощью метода вычисления перемножения матриц $vmult$, реализованного в deal.II. Выводим решения с помощью функции $output\_results$, возвращающий $.h5$ файл, содержащий искомый вектор решения.

Все вышеперечисленные функции реализованы в файлах $FEM2a.h$ и $FEM2b.h$. Рассмотрим реализацию всех перечисленных функций более подробно:

1. Определим базисные функции в $basis\_function$. Так, на вход подается $node-$ локальный номер узла, для которого необходимо рассчитать приписываемую ему базисную функцию с помощью двух заданных точек в двумерном случае и с помощью трех в трехмерном:

```
switch(node) {
    case 0: value = 0.25 * (1 - xi_1) * (1 - xi_2); break;
    case 1: value = 0.25 * (1 + xi_1) * (1 - xi_2); break;
    case 2: value = 0.25 * (1 - xi_1) * (1 + xi_2); break;
    case 3: value = 0.25 * (1 + xi_1) * (1 + xi_2); break;}
```

   Трехмерные базисные функции определяются аналогичным образом, с помощью полиномов Лагранжа:

```
switch(node) {
    case 0: value = 0.125 * (1 - xi_1) * (1 - xi_2) * (1 - xi_3); break;
    case 1: value = 0.125 * (1 + xi_1) * (1 - xi_2) * (1 - xi_3); break;
    case 2: value = 0.125 * (1 - xi_1) * (1 + xi_2) * (1 - xi_3); break;
    case 3: value = 0.125 * (1 + xi_1) * (1 + xi_2) * (1 - xi_3); break;
    case 4: value = 0.125 * (1 - xi_1) * (1 - xi_2) * (1 + xi_3); break;
    case 5: value = 0.125 * (1 + xi_1) * (1 - xi_2) * (1 + xi_3); break;
    case 6: value = 0.125 * (1 - xi_1) * (1 + xi_2) * (1 + xi_3); break;
    case 7: value = 0.125 * (1 + xi_1) * (1 + xi_2) * (1 + xi_3); break;}
```

2. В функции $basis\_gradient$ вычисляем градиент базисной функции в конкретном узле. На вход подаются те же аргументы, что в $basis\_function$. Так как в этой функции нам необходимо вывести вычисленные значения градиента, создадим массив $values$ той же размерности, что и рассматриваемая задача, которую можно определить в конструкторе метода. Из-за громоздкости решения, рассмотрим данную функцию только для двумерной задачи:

```
std::vector<double> values(dim, 0.0); // Здесь dim=2
switch(node) {
    case 0:
      values[0] = 0.25 * (-(1 - xi_2));
      values[1] = 0.25 * (-(1 - xi_1)); break;
    case 1:
      values[0] = 0.25 * (1 - xi_2);
      values[1] = 0.25 * (-(1 + xi_1)); break;
    case 2:
      values[0] = 0.25 *(-(1 + xi_2));
      values[1] = 0.25 * (1 - xi_1); break;
    case 3:
      values[0] = 0.25 * (1 + xi_2);
      values[1] = 0.25 * (1 + xi_1); break;}
```

3. Генерируем сетку в $generate\_mesh$ с помощью встроенного метода $deal.II\ GridGenerator : subdivided\_hyper\_rectangle()$, устанавливаем границы сетки исходя из заданных условий. Рассмотрим на примере трехмерной сетки:

```
template <int dim>
void FEM<dim>::generate_mesh(std::vector<unsigned int> numberOfElements){
  double x_min = 0.0, x_max = 0.04, y_min = 0.0, y_max = 0.08,
    z_min = 0.0, z_max = 0.02;
  Point<dim,double> min(x_min,y_min,z_min), max(x_max,y_max,z_max);
  GridGenerator::subdivided_hyper_rectangle (triangulation,
  numberOfElements, min, max);}
```

4. Задаем граничные условия в $define\_boundary\_conds$. Для этого необходимо понимать способ хранения глобальной нумерации узлов для двумерной сетки в deal.II в таблице (структура table) NodeLocations:

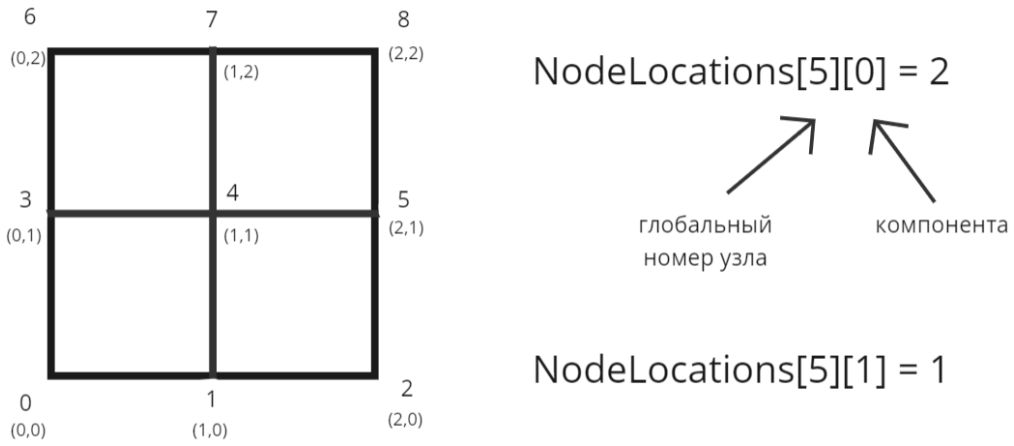



Рис. 3.1. Способ хранения глобальной нумерации узлов в deal.II

Для трехмерной сетки $nodeLocations$ имеет ту же структуру с тем отличием, что компонент у каждого узла 3. Рассмотрим задание граничных условий для трехмерной задачи:

```
template <int dim>
void FEM<dim>::define_boundary_conds(){
  const unsigned int totalNodes = dof_handler.n_dofs(); // количество узлов
  for(unsigned int globalNode=0; globalNode<totalNodes; globalNode++){
    if(nodeLocation[globalNode][0]== 0.){
     boundary_values[globalNode] = 300.0*(1.0 + (nodeLocation[globalNode][1]
       + nodeLocation[globalNode][2])/3.0);}
    if(nodeLocation[globalNode][0]== 0.04){
     boundary_values[globalNode] = 310.0*(1.0 + (nodeLocation[globalNode][1]
       + nodeLocation[globalNode][2])/3.0);}}}
```

5. В функции $setup\_system$ создадим структуры и переменные, необходимые для решения задач: в первую очередь, матрицу жесткости $K$, вектор воздействия $F$ и искомый вектор $D$, вызываем функцию $define\_boundary\_conditions$. Заполняем $nodeLocations$ координатами для каждого глобального индекса:

```
dof_handler.distribute_dofs (fe);
  MappingQ1<dim,dim> mapping;
  std::vector< Point<dim,double> > dof_coords(dof_handler.n_dofs());
  nodeLocation.reinit(dof_handler.n_dofs(),dim);
  DoFTools::map_dofs_to_support_points<dim,dim(mapping,
  dof_handler,dof_coords);
```

```
  for(unsigned int i=0; i<dof_coords.size(); i++){
    for(unsigned int j=0; j<dim; j++){
      nodeLocation[i][j] = dof_coords[i][j];}}
```

Задаем размеры глобальных матриц и векторов:

```
sparsity_pattern.reinit (dof_handler.n_dofs(), dof_handler.n_dofs(),
dof_handler.max_couplings_between_dofs());
DoFTools::make_sparsity_pattern (dof_handler, sparsity_pattern);
sparsity_pattern.compress();
K.reinit (sparsity_pattern);
F.reinit (dof_handler.n_dofs());
D.reinit (dof_handler.n_dofs());
```

Задаем квадратурное правило для численного интегрирования. В функции реализованы квадратурные формулы Гаусса для 3 и 4 точек:

```
quad_points.resize(quadRule);
quad_weight.resize(quadRule);
if (quadRule == 3) {
    quad_points[0] = -sqrt(3./5.); quad_points[1] = 0.0;
    quad_points[2] = sqrt(3./5.);
    quad_weight[0] = 5 / 9.; quad_weight[1] = 8 / 9.;
    quad_weight[2] = 5 / 9.;}
  else if (quadRule == 4) {
    quad_points[0] = -sqrt((3./7.) - (2 / 7.) * sqrt(6 / 5.));
    quad_points[1] = sqrt((3./7.) - (2 / 7.) * sqrt(6 / 5.));
    quad_points[2] = -sqrt((3./7.) + (2 / 7.) * sqrt(6 / 5.));
    quad_points[3] = sqrt((3./7.) + (2 / 7.) * sqrt(6 / 5.));
    quad_weight[0] = (18. + sqrt(30.)) / 36.;
    quad_weight[1] = (18. + sqrt(30.)) / 36.;
    quad_weight[2] = (18. - sqrt(30.)) / 36.;
    quad_weight[3] = (18. - sqrt(30.)) / 36.;}
```

6. Реализация сборки локальных и глобальных матриц $F$ и $K$ осуществлена в функции $assemble\_system$. Для начала поймем, как собирается локальная матрица жесткости:

$$K^{local}[A][B] = -\int\limits_{\Omega_\xi}\int \frac{dN^A}{dx_I}\kappa_{IJ}\frac{dN^B}{dx_J}det(J)d\xi_1 d\xi_2, \quad I,J = 0,..,dim-1 \tag{3.1}$$

$$\frac{dN^A}{dx_I} = \frac{dN^A}{d\xi_1}\frac{d\xi_1}{dx_I} + \frac{dN^A}{d\xi_2}\frac{d\xi_2}{dx_I}, \quad \text{где } \frac{d\xi_2}{dx_I} = J^{-1}[2][I] \qquad (3.2)$$

Значит, нам необходимо рассчитать якобиан по определению и найти обратную для него матрицу методом $.invert()$. Вновь рассмотрим для трехмерной задачи:

```
FullMatrix<double> Jacobian(dim,dim);
FullMatrix<double> invJacob(dim,dim), kappa(dim,dim); \\ J^{-1}
kappa = 0.; kappa[0][0] = 385.; kappa[1][1] = 385.; kappa[2][2] = 385.;
double detJ;
for(unsigned int q1=0; q1<quadRule; q1++){
  for(unsigned int q2=0; q2<quadRule; q2++){
    for(unsigned int q3=0; q3<quadRule; q3++){
      Jacobian = 0.;
      for(unsigned int i=0;i<dim;i++){
        for(unsigned int j=0;j<dim;j++){
          for(unsigned int A=0; A<dofs_per_elem; A++){
            Jacobian[i][j] += nodeLocation[local_dof_indices[A]][i]
^^I^^I          * basis_gradient(A, quad_points[q1], quad_points[q2],
              quad_points[q3])[j];}}}
       detJ = Jacobian.determinant();
       invJacob.invert(Jacobian);} ... }}
```

Не выходя из цикла, собираем $K^{local}$:

```
for(unsigned int A=0; A<dofs_per_elem; A++){
  for(unsigned int B=0; B<dofs_per_elem; B++){
    for(unsigned int i=0;i<dim;i++){
      for(unsigned int j=0;j<dim;j++){
        for(unsigned int I=0;I<dim;I++){
          for(unsigned int J=0;J<dim;J++){
            Klocal[A][B] += basis_gradient(A, quad_points[q1],
              quad_points[q2], quad_points[q3])[I]
              * invJacob[I][i] * kappa[i][j] * invJacob[J][j]
              * basis_gradient(B, quad_points[q1], quad_points[q2],
                                  quad_points[q3])[J]
              * detJ * quad_weight[q1] * quad_weight[q2]
              * quad_weight[q3]; }}}}}}
```

$F^{local}$ в данных задачах мы приняли за нулевой вектор, однако, при изменении условия возможно переобозначить его в зависимости от заданных условий. Собираем локальные матрицы в глобальные:

```
for(unsigned int A=0; A<dofs_per_elem; A++){
    for(unsigned int B=0; B<dofs_per_elem; B++){
       K.add(local_dof_indices[A], local_dof_indices[B], Klocal[A][B]);}}
```

7. Находим искомый вектор в функции $solve$:

```
template <int dim>
void FEM<dim>::solve(){
  SparseDirectUMFPACK  A;
  A.initialize(K);
  A.vmult (D, F); //D=K^{-1}*F }
```

8. Выводим результат:

```
template <int dim>
void FEM<dim>::output_results(){
  std::ofstream output1("solution.vtk");
  DataOut<dim> data_out;
  data_out.attach_dof_handler(dof_handler);
  data_out.add_data_vector(D, nodal_solution_names,
                           DataOut<dim>::type_dof_data,
                           nodal_data_component_interpretation);
  data_out.build_patches();
  data_out.write_vtk(output1);
  output1.close(); }}
```

Искомый нами вектор решения представляет собой распределение температуры в узлах сетки. Посмотрим на векторы решения для двумерной и трехмерной задач с помощью платформы графической визуализации $ParaView$ (Рис. 3.2.; Рис.3.3):

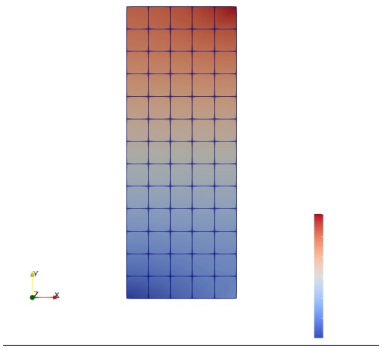
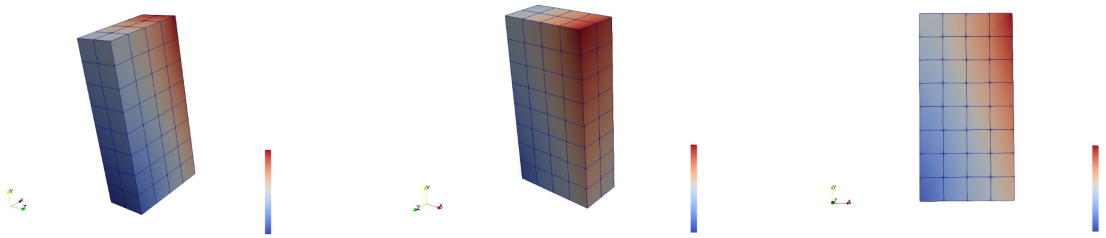


Рис. 3.2. Визуализация 2D модели решения стационарной задачи.



(a) (b) (c)

Рис. 3.3. Визуализация 3D модели решения стационарной задачи.

Посчитанный нами вектор решения дает нам вполне корректное представление о температурном распределении в заданных узлах. Полученные результаты в дальнейшем применимы для более глубокого анализа термодинамического поля, могут быть использованы для прогнозирования поведения материала при других начальных тепловых потоках и граничных условиях.

# 4. Заключение

В ходе исследования темы курсовой работы были рассмотрены типы дифференциальных уравнений в частных производных второго порядка (так называемые уравнения математической физики) и теоретические принципы метода конечных элементов, исследован теоретический подход к решению эллиптических уравнений заданным методом, успешно реализован численный расчёт стационарной задачи теплопроводности на языке $C++$ с помощью библиотеки с открытым исходным кодом $deal.II$, была рассмотрена графическая визуализация посчитанного решения. Проведенное исследование демонстрирует эффективность применения метода конечных элементов для решения задач математической физики с помощью современных инструментов программирования и методов визуализации. Проведенное исследование и полученные результаты могут быть полезны в дальнейшем при проектировании теплообменников, анализе энергоэффективности при проектировании систем отопления для минимизации потерь тепла, а также для многих других инженерных задач.

# 5. Список литературы

The finite element method for problems in physics. URL https://www.coursera.org/learn/finite-element-method. — University of Michigan, 2014.

Иваньшин П.Н. Метод конечных элементов. — Казанский Федеральный Университет, 2013.

O.C. Zienkiewicz, R.L. Taylor and J.Z. Zhu. The Finite Element Method: Its Basis and Fundamentals. — Butterworth-Heinemann, 2013.

Wolfgang Bangerth's Lectures. URL https://www.math.colostate.edu/ bangerth /videos.html. — deal.II manual, 2014

Ю.А.Сагдеева, С.П. Копысов и А.К.Новиков. Введение в метод конечных элементов. — Удмуртский Университет, 2011.