

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программной и системной инженерии

РЕКОМЕНДОВАНО К ЗАЩИТЕ
В ГЭК И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

Заведующий кафедрой

А.Г. Ивашко
Д.т.н., профессор

А.Г. Ивашко

20.06. 2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(магистерская диссертация)

ПРОГНОЗИРОВАНИЕ СТРУКТУРЫ ДЕТАЛИ ПОСЛЕ ЛАЗЕРНОЙ СВАРКИ

Прикладная информатика 09.04.03

Магистерская программа: Прикладная информатика в экономике

Выполнил работу
Студент 2 курса
очной формы обучения

Мустафин
(Подпись)

Мустафин
Марсель
Махмутович

Научный руководитель
к.т.н., доцент

Цыганова
(Подпись)

Цыганова
Мария
Сергеевна

Рецензент
д.т.н., профессор каф. Прикладная
и техническая физика ФТИ ТюмГУ

Шабаров
(Подпись)

Шабаров
Александр
Борисович

г. Тюмень, 2019

Содержание

ВВЕДЕНИЕ.....	4
1. ПОСТАНОВКА ЦЕЛИ И ЗАДАЧ	6
2. МОДЕЛЬ ТЕПЛОВЫХ ПОЛЕЙ НА РАВНОМЕРНОЙ СЕТКЕ	8
2.1. Принятые обозначения и допущения.....	8
2.2. Двумерная модель температурных полей	9
2.3. Двумерная модель температурных полей с учетом конвекции	9
2.4. Выбор инструментария для разработки программного обеспечения	20
2.4.1. Анализ существующих программных средств для реализации программного обеспечения.....	21
2.4.2. PYTHON.....	21
2.4.3. C++.....	23
2.4.4. FORTRAN	24
2.4.5. Вывод по выбору инструментария для разработки программного обеспечения	24
2.5. Реализация моделирования тепловых полей.....	25
2.6. Результаты моделирования тепловых полей.....	32
2.7. Выводы по результатам, полученным на равномерной сетке	42
3. МОДЕЛЬ ТЕПЛОВЫХ ПОЛЕЙ НА АДАПТИВНОЙ СЕТКЕ	43
3.1. Схема построения адаптивной сетки	43
3.2. Математическая модель распространения тепла на адаптивной сетке..	48
3.2.1. Описание обозначений, условий и доработок схемы построения адаптивной сетки.	48
3.2.2. Описание алгоритма построения адаптивной сетки	50
3.2.3. Получение разностных уравнений.....	54

3.2.4. Учет граничных условий.....	57
3.2.5. Реализация моделирования тепловых полей.	58
4. МОДЕЛИРОВАНИЕ ФАЗОВЫХ ПРЕВРАЩЕНИЙ	65
4.1. Обзор методов исследования кинетики фазовых превращений	65
4.2. Моделирование фазовых превращений при охлаждении трубы с использованием методологии теории массового обслуживания	67
4.3. Общая схема построения имитационной модели.....	67
4.4. Алгоритм, имитирующий растворение δ -феррита в изотермических условиях	72
4.5. Алгоритм определения значений кинетических параметров превращения в изотермических условиях.....	74
4.6. Алгоритм, имитирующий растворение δ -феррита при непрерывном охлаждении	78
4.7. Реализация моделирования фазовых превращений.	79
4.8. Полученные результаты моделирования фазовых превращений.	81
ЗАКЛЮЧЕНИЕ	82
СПИСОК ЛИТЕРАТУРЫ.....	84

ВВЕДЕНИЕ

Достижения науки и техники, развитие энергетики, строительства, машиностроения, всех отраслей промышленности и сельского хозяйства связаны с производством разнообразных материалов, обладающих определенным комплексом механических, физических, технологических и служебных свойств.

В настоящее время в качестве основных конструкционных материалов металлы и сплавы сохраняют свое доминирующее положение, в первую очередь это относится к металлическим материалам на основе железа, которые применяют во многих отраслях. И, несмотря на то, что выявлено очень много сплавов с требуемыми свойствами, продолжаются эксперименты с термической обработкой имеющихся материалов, которые изменяют их механические свойства [7].

Улучшение качества выпускаемых машин, оборудования, приборов, повышение их технического уровня, производительности, надежности и безопасности в эксплуатации невозможны без улучшения качества применяемых металлов и освоение новых марок сталей и сплавов, обладающих более высокими характеристиками по сравнению с существующими [7].

Главным средством повышения качества металла до желаемого уровня, требуемого современной техникой, является технология металлургического производства. Достижение необходимого сочетания высокой прочности с высокой пластичностью, вязкостью, низкой чувствительностью к концентраторам напряжений и другими характеристиками, требуемыми от современного конструкционного материала, практически невозможно. Металлургическая технология производства сталей, и в первую очередь технология выплавки, является основным фактором получения требуемых механических свойств металла заданного химического состава, достижения необходимых показателей надежности и работоспособности конструкций [7].

Таким образом, возникла необходимость комплексного подхода при прогнозировании совершенствования материалов, учитывающих тенденции развития технологий производства и возможных ее особенностей, а также влияния новых

разработок в области технологии на появление материалов с более высокими характеристиками по сравнению с существующими.

1. ПОСТАНОВКА ЦЕЛИ И ЗАДАЧ

Практически во всех сферах современного производства применяются трубы из нержавеющей стали и жаропрочных сплавов, которые можно обобщенно разделить на бесшовные и сварные. Сварные трубы изготавливаются путем заваривания скатанного в трубу плоского металлического листа. Этот процесс сопровождается образованием шва (в результате плавления материала). Себестоимость сварных труб существенно ниже бесшовных, однако существует ряд ограничений области применения сварных труб. В первую очередь это связано с ограниченной толщиной стенки готового изделия и структурно-фазовой неоднородностью материала в зоне шва [30].

В настоящее время указанные ограничения могут быть существенно уменьшены путем разработки и внедрения новых технологий сварки. Разработка таких технологий должна базироваться на результатах системных и прикладных научных исследований, которые невозможны без применения современных методов моделирования и численных расчетов. Вследствие чего, в рамках данного исследования была поставлена цель: разработка и реализация математических моделей, позволяющих прогнозировать структуру зоны сварного шва нержавеющей стали после применения различных режимов лазерной сварки. В рамках достижения главной цели можно выделить две подцели:

1. построение и реализация математической модели тепловых полей, формирующихся в процессе сварки и последующего остывания трубы;
2. построение и реализация математической модели фазовых превращений, происходящих в процессе изменения температуры трубы в зоне сварного шва и околошовной зоне.

Для достижения поставленных целей определены следующие задачи:

- изучение процессов передачи тепла, проходящих в изучаемом объеме;
- построение математической модели распространения тепла в изучаемом объеме;
- создание программного обеспечения для расчета тепловых полей;

- изучение существующих математических моделей, описывающих кинетику фазовых превращений;
- изучение физических процессов, протекающих при переходе из одной фазы в другую;
- создание программного комплекса для прогнозирования структурного состава стали после термической обработки;
- проведение вычислительных экспериментов с целью проверки адекватности предложенных методов и алгоритмов.

Объектом исследования являются процессы теплопередачи в зоне сварочного шва, и процессы структурных изменений.

Исследование выполнялось при поддержке Министерства науки и высшего образования Российской Федерации, уникальный идентификационный номер ФЦП RFMEFI57817X0252 [29].

2. МОДЕЛЬ ТЕПЛОВЫХ ПОЛЕЙ НА РАВНОМЕРНОЙ СЕТКЕ

Модель предназначена для расчета температурных полей заготовки в процессе сварки и последующего охлаждения.

2.1. Принятые обозначения и допущения

При моделировании температурных полей использовались следующие обозначения:

T – температура;

t – время (от начала обработки);

(r, φ, z) – цилиндрические координаты;

q – плотность мощности источника тепла (лазера) в точке сварки;

T^0 – температура среды, она же – начальная температура трубы;

h – толщина стенки трубы;

D – диаметр трубы.

С учетом того, что толщина трубы может принимать различные значения, могут использоваться модели двух типов:

- двумерная модель (на основе упрощающего предположения о постоянстве температуры по толщине трубы в каждый конкретный момент времени) – для «тонкой» трубы;
- трехмерная модель (с учетом изменения значений температуры по толщине трубы) – для «толстой» трубы.

Выбор первой или второй из указанных моделей для проведения рабочих расчетов должен обосновываться конкретным значением толщины и характеристиками материала трубы. С учетом цилиндрической формы трубы для построения обеих моделей естественно использовать цилиндрическую систему координат (r, φ, z) .

Для имеющихся образцов трубы, где толщина трубы изменяется от 2 до 10 мм, было принято решение пренебречь распространением тепла внутри стержня, и использовать двумерную модель для расчета температурных полей.

2.1.1. Двумерная модель температурных полей

Поскольку распределение тепла по толщине трубы не моделируется, для описания изменения температуры достаточно двух координат – (см. рис. 1).

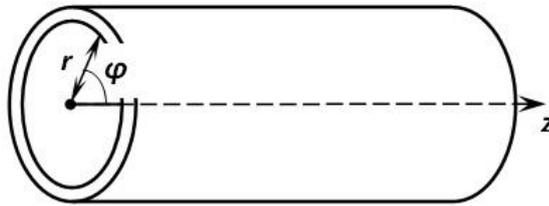


Рисунок 1. Цилиндрические координаты ($r = \text{const}$).

Моделирование выполнялось на основе уравнения теплопроводности, записанного в цилиндрических координатах при постоянном значении r , равном $D/2$ [8].

(1)

где ρ – плотность материала,

c – удельная теплоемкость,

$\lambda = \lambda, T$ – теплопроводность.

Для учета переноса теплоты конвекцией и излучением с двух сторон поверхности трубы в правую часть уравнения введено слагаемое. Такой подход позволяет учесть передачу тепла в окружающую среду, не применяя дифференцирование по координате r , и, тем самым, использовать двумерную (а не трехмерную) модель. Моделирование конвекции и излучения отличается только видом функции

2.2. Двумерная модель температурных полей с учетом конвекции

В случае конвекции

где κ – коэффициент теплоотдачи.

В этом случае распределение температуры описывалось уравнением

(2)

Начальное условие задавалось следующим образом: ; граничные условия по

координате z (граничные условия третьего рода) – в виде:

(3) Кроме того, по координате φ задавались

условия сопряжения [8]:

$$T(0+0) = T(2\pi-0), \quad (4)$$

Решение уравнения (2) выполнялось методом конечных разностей. С этой целью вводились равномерные пространственная и временная сетки.

Пространственная сетка:

по φ φ ,

z j_z ,

φ и z – число узлов по координатам φ и z соответственно;

временная сетка:

Тогда значение температуры в узле (i, j) пространственной сетки на n -м шаге по времени.

ых $, j$ сопряжения (по [8])

$$, j \cdot \quad (5)$$

ахх по координатам выполнялась (явная схема по λ , неявная по T по методике, изложенной в [8]):

где

h_φ – значение шага по φ ,

h_z – значение шага по z .

Аппроксимация функции $q(t, \varphi, z)$:

где q – плотность мощности излучения в точке сварки, $[\]$ – целая часть числа.

Требование $j = n+1$ обусловлено выбором шага по времени (предполагается, что лазер

за один временной шаг смещается на один шаг вдоль оси z); условие $n \leq z - 1$ – прекращением работы лазера после достижения конечного узла по координате z .

Аппроксимация слагаемого, отвечающего за перенос тепла через поверхность трубы:

В результате получено разностное уравнение:

$$\begin{aligned} \rho \cdot c \cdot \frac{T_{ij}^{n+1} - T_{ij}^n}{\tau} = & \frac{1}{r^2} \cdot \frac{1}{h_\varphi} \cdot \left(\lambda_{i+\frac{1}{2},j}^n \frac{T_{i+1,j}^{n+1} - T_{ij}^{n+1}}{h_\varphi} - \lambda_{i-\frac{1}{2},j}^n \frac{T_{ij}^{n+1} - T_{i-1,j}^{n+1}}{h_\varphi} \right) + \\ & + \frac{1}{h_z} \cdot \left(\lambda_{i,j+\frac{1}{2}}^n \frac{T_{i,j+1}^{n+1} - T_{ij}^{n+1}}{h_z} - \lambda_{i,j-\frac{1}{2}}^n \frac{T_{ij}^{n+1} - T_{i,j-1}^{n+1}}{h_z} \right) + \frac{2\kappa}{h} (T^0 - T_{ij}^n) + q_{ij}^n. \end{aligned} \quad (6)$$

После перегруппировки слагаемых:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda_{i-\frac{1}{2},j}^n}{h_\varphi^2} \cdot T_{i-1,j}^{n+1} + \frac{\lambda_{i,j-\frac{1}{2}}^n}{h_z^2} \cdot T_{i,j-1}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda_{i+\frac{1}{2},j}^n + \lambda_{i-\frac{1}{2},j}^n}{r^2 \cdot h_\varphi^2} + \frac{\lambda_{i,j+\frac{1}{2}}^n + \lambda_{i,j-\frac{1}{2}}^n}{h_z^2} \right) \cdot T_{ij}^{n+1} + \\ + \frac{\lambda_{i,j+\frac{1}{2}}^n}{h_z^2} \cdot T_{i,j+1}^{n+1} + \frac{\lambda_{i+\frac{1}{2},j}^n}{r^2 \cdot h_\varphi^2} \cdot T_{i+1,j}^{n+1} = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{ij}^n - \frac{2\kappa}{h} T^0 - q_{ij}^n, \end{aligned} \quad (7)$$

$$23_\varphi - 1, \quad j23_z - 1,$$

Аппроксимация граничных условий по z :

, i

откуда

$$(8)$$

Таким образом, уравнение теплопроводности (2) с граничными условиями (3) и условиями сопряжения (4) аппроксимируется разностными уравнениями (7) с граничными условиями (8) и условиями сопряжения (5).

Для получения поля температуры на n -м шаге по времени необходимо из полученных соотношений определить значения . На шаге $n = 0$ используются также начальные условия.

Рассмотрим подробнее шаг $n = 0$.

На этом шаге $\lambda_{i+\frac{1}{2},j}^0 = \lambda_{i-\frac{1}{2},j}^0 = \lambda_{i,j+\frac{1}{2}}^0 = \lambda_{i,j-\frac{1}{2}}^0 = \lambda(T^0)$. Уравнения (7) примут вид

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{i-1,j}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i,j-1}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{ij}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i,j+1}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{i+1,j}^1 = -\frac{\rho \cdot c}{\tau} \cdot T^0 - q_{ij}^0, \end{aligned} \quad (9)$$

$$23_\varphi - 1, \quad j23_z - 1.$$

Для i, φ, j и j_z соответствующие значения температуры выражаются через температуру в других узлах сетки (из соотношений (5) и (8)). Поэтому неизвестными, подлежащими определению, являются

$$T_{22}^1, T_{23}^1, \dots, T_{2,N_z-1}^1, T_{32}^1, T_{33}^1, \dots, T_{3,N_z-1}^1, \dots, T_{N_\varphi-1,2}^1, T_{N_\varphi-1,3}^1, \dots, T_{N_\varphi-1,N_z-1}^1.$$

Фактически, соотношения (9) представляют собой систему из $(N_\varphi-2) \cdot (N_z-2)$ линейных уравнений с таким же количеством неизвестных. При составлении этих уравнений необходимо отдельно рассмотреть случаи $i=2, \varphi-1, j=2$ и j_z-1 , поскольку для них, помимо соотношений (9), необходимо также учитывать (5) и (8). Для всех прочих значений i и j достаточно просто записать уравнения (9).

- $i=2, \quad j=2$

Соотношение (9) примет вид:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{12}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{21}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{22}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{23}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{32}^1 = -\frac{\rho \cdot c}{\tau} \cdot T^0 - q_{22}^0. \end{aligned}$$

Из (5) из (8)

Окончательно получим:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-1,2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} - \frac{\lambda^2(T^0)}{h_z^2 \cdot (\lambda(T^0) + \kappa \cdot h_z)} \right) \cdot T_{22}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{23}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{32}^1 = -\left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T^0)}{h_z} \cdot \frac{\kappa}{\lambda(T^0) + \kappa \cdot h_z} \right) \cdot T^0 - q_{22}^0. \end{aligned}$$

- $i = 2, j = 3, 4, \dots, N_z - 2$

С учетом (5)

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-1,j}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{2,j-1}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{2j}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{2,j+1}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{3j}^1 = -\frac{\rho \cdot c}{\tau} \cdot T^0 - q_{2j}^0. \end{aligned}$$

- $i = 2, j = N_z - 1$

Соотношение (9) примет вид:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{1,N_z-1}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{2,N_z-2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{2,N_z-1}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{2,N_z}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{3,N_z-1}^1 = -\frac{\rho \cdot c}{\tau} \cdot T^0 - q_{2,N_z-1}^0. \end{aligned}$$

Из (5) из (8)

Окончательно получим:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-1,N_z-1}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{2,N_z-2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} - \frac{\lambda^2(T^0)}{h_z^2 \cdot (\lambda(T^0) + \kappa \cdot h_z)} \right) \cdot T_{2,N_z-1}^1 + \\ + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{3,N_z-1}^1 = -\left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T^0)}{h_z} \cdot \frac{\kappa}{\lambda(T^0) + \kappa \cdot h_z} \right) \cdot T^0 - q_{2,N_z-1}^0. \end{aligned}$$

- $i = N_\varphi - 1, j = 2$

Соотношение (9) примет вид:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-2,2}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,1}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{N_\varphi-1,2}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,3}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{N_\varphi,2}^1 = -\frac{\rho \cdot c}{\tau} \cdot T^0 - q_{N_\varphi-1,2}^0. \end{aligned}$$

Из (5) из (8)

Окончательно получим:

$$\begin{aligned} & \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-2,2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} - \frac{\lambda^2(T^0)}{h_z^2 \cdot (\lambda(T^0) + \kappa \cdot h_z)} \right) \cdot T_{N_\varphi-1,2}^1 + \\ & + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,3}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{22}^1 = - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T^0)}{h_z} \cdot \frac{\kappa}{\lambda(T^0) + \kappa \cdot h_z} \right) \cdot T^0 - q_{N_\varphi-1,2}^0. \end{aligned}$$

- $i = N_\varphi - 1, j = 3, 4, \dots, N_z - 2$

С учетом (5)

$$\begin{aligned} & \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-2,j}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,j-1}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{N_\varphi-1,j}^1 + \\ & + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,j+1}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{2j}^1 = - \frac{\rho \cdot c}{\tau} \cdot T^0 - q_{N_\varphi-1,j}^0. \end{aligned}$$

- $i = N_\varphi - 1, j = N_z - 1$

Соотношение (9) примет вид:

$$\begin{aligned} & \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-2,N_z-1}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,N_z-2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{N_\varphi-1,N_z-1}^1 + \\ & + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,N_z}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{N_\varphi,N_z-1}^1 = - \frac{\rho \cdot c}{\tau} \cdot T^0 - q_{N_\varphi-1,N_z-1}^0. \end{aligned}$$

Из (5) из (8)

Окончательно получим:

$$\begin{aligned} & \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{N_\varphi-2,N_z-1}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{N_\varphi-1,N_z-2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} - \frac{\lambda^2(T^0)}{h_z^2 \cdot (\lambda(T^0) + \kappa \cdot h_z)} \right) \cdot T_{N_\varphi-1,N_z-1}^1 + \\ & + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{2,N_z-1}^1 = - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T^0)}{h_z} \cdot \frac{\kappa}{\lambda(T^0) + \kappa \cdot h_z} \right) \cdot T^0 - q_{N_\varphi-1,N_z-1}^0. \end{aligned}$$

- $i = 3, 4, \dots, N_\varphi - 2, j = 2$

Соотношение (9) примет вид:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{i-1,2}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i1}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{i2}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i3}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{i+1,2}^1 = -\frac{\rho \cdot c}{\tau} \cdot T^0 - q_{i2}^0. \end{aligned}$$

Из (8)

Окончательно получим:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{i-1,2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} - \frac{\lambda^2(T^0)}{h_z^2 \cdot (\lambda(T^0) + \kappa \cdot h_z)} \right) \cdot T_{i2}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i3}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{i+1,2}^1 = -\left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T^0)}{h_z} \cdot \frac{\kappa}{\lambda(T^0) + \kappa \cdot h_z} \right) \cdot T^0 - q_{i2}^0. \end{aligned}$$

- $i = 3, 4, \dots, N_\varphi - 2, j = N_z - 1$

Соотношение (9) примет вид:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{i-1, N_z-1}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i, N_z-2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} \right) \cdot T_{i, N_z-1}^1 + \\ + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i, N_z}^1 + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{i+1, N_z-1}^1 = -\frac{\rho \cdot c}{\tau} \cdot T^0 - q_{i, N_z-1}^0. \end{aligned}$$

Из (8)

Окончательно получим:

$$\begin{aligned} \frac{1}{r^2} \cdot \frac{\lambda(T^0)}{h_\varphi^2} \cdot T_{i-1, N_z-1}^1 + \frac{\lambda(T^0)}{h_z^2} \cdot T_{i, N_z-2}^1 - \left(\frac{\rho \cdot c}{\tau} + \frac{2\lambda(T^0)}{r^2 \cdot h_\varphi^2} + \frac{2\lambda(T^0)}{h_z^2} - \frac{\lambda^2(T^0)}{h_z^2 \cdot (\lambda(T^0) + \kappa \cdot h_z)} \right) \cdot T_{i, N_z-1}^1 + \\ + \frac{\lambda(T^0)}{r^2 \cdot h_\varphi^2} \cdot T_{i+1, N_z-1}^1 = -\left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T^0)}{h_z} \cdot \frac{\kappa}{\lambda(T^0) + \kappa \cdot h_z} \right) \cdot T^0 - q_{i, N_z-1}^0. \end{aligned}$$

Для формирования матрицы системы линейных уравнений необходимо соглашение о нумерации неизвестных. Была принята следующая нумерация: неизвестной присваивался номер

(10)

В соответствии с этим, элемент матрицы, стоящий в строке с номером текущего

разностного уравнения в столбце с номером (10) приравнялся к коэффициенту при в соотношении (9) либо в соотношениях, полученных с учетом (5) и (8) (для значений i и j , рассмотренных особо). Остальные элементы матрицы равны нулю. Аналогичным образом формировался столбец свободных членов (правая часть).

Результатом решения полученного матричного уравнения являлись значения температуры в неграничных узлах сетки. Значения в граничных узлах (т. е. , , и) определялись из соотношений (5) и (8) с использованием найденных неграничных значений. Таким образом, по завершении шага $n = 0$ определялись значения во всех узлах пространственной сетки.

Шаги $n = 1, 2, \dots$ выполнялись аналогично. Отличия состояли в следующем:

- неизвестными, подлежащими определению, являлись значения $T_{22}^{n+1}, T_{23}^{n+1}, \dots, T_{2,N_z-1}^{n+1}, T_{32}^{n+1}, T_{33}^{n+1}, \dots, T_{3,N_z-1}^{n+1}, \dots, T_{N_\varphi-1,2}^{n+1}, T_{N_\varphi-1,3}^{n+1}, \dots, T_{N_\varphi-1,N_z-1}^{n+1}$;
- вместо начальных условий использовались значения температуры, полученные на предыдущем шаге;
- значения теплопроводности уже не предполагались одними и теми же в разных узлах пространственной сетки (в силу различий в значениях температуры), вследствие чего вместо (9) использовались соотношения (7).

Так же, как и на шаге $n = 0$, отдельно рассматривались случаи $2, \varphi-1, j_2$ и j_z-1 :

- $i = 2, j = 2$

С учетом (5) и (8),

$$\frac{1}{r^2} \cdot \frac{\lambda_{1+\frac{1}{2},2}^n}{h_\varphi^2} \cdot T_{N_\varphi-1,2}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda_{2-\frac{1}{2},2}^n + \lambda_{2+\frac{1}{2},2}^n}{r^2 \cdot h_\varphi^2} + \frac{\lambda_{2,2-\frac{1}{2}}^n + \lambda_{2,2+\frac{1}{2}}^n}{h_z^2} - \frac{\left(\lambda_{2,1+\frac{1}{2}}^n \right)^2}{h_z^2 \cdot \left(\lambda_{2,1+\frac{1}{2}}^n + \kappa \cdot h_z \right)} \right) \cdot T_{22}^{n+1} +$$

$$+ \frac{\lambda_{2,2+\frac{1}{2}}^n}{h_z^2} \cdot T_{23}^{n+1} + \frac{\lambda_{2+\frac{1}{2},2}^n}{r^2 \cdot h_\varphi^2} \cdot T_{32}^{n+1} = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{22}^n - \left(\frac{2\kappa}{h} + \frac{\lambda_{2,1+\frac{1}{2}}^n \cdot \kappa}{h_z \cdot \left(\lambda_{2,1+\frac{1}{2}}^n + \kappa \cdot h_z \right)} \right) \cdot T^0 - q_{22}^n.$$

- $i = 2, j = 3, 4, \dots, N_z - 2$

С учетом (5),

$$\begin{aligned} & \frac{1}{r^2} \cdot \frac{\lambda_{1+\frac{1}{2},j}^n}{h_\varphi^2} \cdot T_{N_\varphi-1,j}^{n+1} + \frac{\lambda_{2,j-\frac{1}{2}}^n}{h_z^2} \cdot T_{2,j-1}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda_{2-\frac{1}{2},j}^n + \lambda_{2+\frac{1}{2},j}^n}{r^2 \cdot h_\varphi^2} + \frac{\lambda_{2,j-\frac{1}{2}}^n + \lambda_{2,j+\frac{1}{2}}^n}{h_z^2} \right) \cdot T_{2j}^{n+1} + \\ & + \frac{\lambda_{2,j+\frac{1}{2}}^n}{h_z^2} \cdot T_{2,j+1}^1 + \frac{\lambda_{2+\frac{1}{2},j}^n}{r^2 \cdot h_\varphi^2} \cdot T_{3j}^1 = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{2j}^n - \frac{2\kappa}{h} \cdot T^0 - q_{2j}^n. \end{aligned}$$

- $i = 2, j = N_z - 1$

С учетом (5) и (8),

$$\begin{aligned} & \frac{1}{r^2} \cdot \frac{\lambda_{2-\frac{1}{2},N_z-1}^n}{h_\varphi^2} \cdot T_{N_\varphi-1,N_z-1}^{n+1} + \frac{\lambda_{2,(N_z-1)-\frac{1}{2}}^n}{h_z^2} \cdot T_{2,N_z-2}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda_{2-\frac{1}{2},N_z-1}^n + \lambda_{2+\frac{1}{2},N_z-1}^n}{r^2 \cdot h_\varphi^2} + \right. \\ & \left. + \frac{\lambda_{2,(N_z-1)-\frac{1}{2}}^n + \lambda_{2,(N_z-1)+\frac{1}{2}}^n}{h_z^2} - \frac{\left(\lambda_{2,(N_z-1)+\frac{1}{2}}^n \right)^2}{h_z^2 \cdot \left(\lambda_{2,(N_z-1)+\frac{1}{2}}^n + \kappa \cdot h_z \right)} \right) \cdot T_{2,N_z-1}^{n+1} + \frac{\lambda_{2+\frac{1}{2},N_z-1}^n}{r^2 \cdot h_\varphi^2} \cdot T_{3,N_z-1}^1 = \\ & = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{2,N_z-1}^n - \left(\frac{2\kappa}{h} + \frac{\lambda_{2,(N_z-1)+\frac{1}{2}}^n \cdot \kappa}{h_z \cdot \left(\lambda_{2,(N_z-1)+\frac{1}{2}}^n + \kappa \cdot h_z \right)} \right) \cdot T^0 - q_{2,N_z-1}^n. \end{aligned}$$

- $i = N_\varphi - 1, j = 2$

С учетом (5) и (8),

$$\begin{aligned}
& \frac{1}{r^2} \cdot \frac{\lambda_{(N_\varphi-1)-\frac{1}{2},2}^n}{h_\varphi^2} \cdot T_{N_\varphi-2,2}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda_{(N_\varphi-1)-\frac{1}{2},2}^n + \lambda_{(N_\varphi-1)+\frac{1}{2},2}^n}{r^2 \cdot h_\varphi^2} + \frac{\lambda_{N_\varphi-1,2-\frac{1}{2}}^n + \lambda_{N_\varphi-1,2+\frac{1}{2}}^n}{h_z^2} - \right. \\
& \left. - \frac{\left(\lambda_{N_\varphi-1,1+\frac{1}{2}}^n \right)^2}{h_z^2 \cdot \left(\lambda_{N_\varphi-1,2-\frac{1}{2}}^n + \kappa \cdot h_z \right)} \right) \cdot T_{N_\varphi-1,2}^{n+1} + \frac{\lambda_{N_\varphi-1,2+\frac{1}{2}}^n}{h_z^2} \cdot T_{N_\varphi-1,3}^{n+1} + \frac{\lambda_{(N_\varphi-1)+\frac{1}{2},2}^n}{r^2 \cdot h_\varphi^2} \cdot T_{22}^{n+1} = \\
& = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{N_\varphi-1,2}^n - \left(\frac{2\kappa}{h} + \frac{\lambda_{N_\varphi-1,1+\frac{1}{2}}^n \cdot \kappa}{h_z \cdot \left(\lambda_{N_\varphi-1,1+\frac{1}{2}}^n + \kappa \cdot h_z \right)} \right) \cdot T^0 - q_{N_\varphi-1,2}^n.
\end{aligned}$$

- $i = N_\varphi - 1, j = 3, 4, \dots, N_z - 2$

С учетом (5),

$$\begin{aligned}
& \frac{1}{r^2} \cdot \frac{\lambda_{(N_\varphi-1)-\frac{1}{2},j}^n}{h_\varphi^2} \cdot T_{N_\varphi-2,j}^{n+1} + \frac{\lambda_{N_\varphi-1,j-\frac{1}{2}}^n}{h_z^2} \cdot T_{N_\varphi-1,j-1}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda_{(N_\varphi-1)-\frac{1}{2},j}^n + \lambda_{(N_\varphi-1)+\frac{1}{2},j}^n}{r^2 \cdot h_\varphi^2} + \right. \\
& \left. + \frac{\lambda_{N_\varphi-1,j-\frac{1}{2}}^n + \lambda_{N_\varphi-1,j+\frac{1}{2}}^n}{h_z^2} \right) \cdot T_{N_\varphi-1,j}^{n+1} + \frac{\lambda_{N_\varphi-1,j+\frac{1}{2}}^n}{h_z^2} \cdot T_{N_\varphi-1,j+1}^{n+1} + \frac{\lambda_{(N_\varphi-1)+\frac{1}{2},j}^n}{r^2 \cdot h_\varphi^2} \cdot T_{2j}^{n+1} = \\
& = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{N_\varphi-1,j}^n - \frac{2\kappa}{h} \cdot T^0 - q_{N_\varphi-1,j}^n.
\end{aligned}$$

- $i = N_\varphi - 1, j = N_z - 1$

С учетом (5) и (8),

$$\begin{aligned}
& \frac{1}{r^2} \cdot \frac{\lambda^n_{(N_\varphi-1)-\frac{1}{2}, N_z-1}}{h_\varphi^2} \cdot T_{N_\varphi-2, N_z-1}^{n+1} + \frac{\lambda^n_{N_\varphi-1, (N_z-1)-\frac{1}{2}}}{h_z^2} \cdot T_{N_\varphi-1, N_z-2}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \right. \\
& + \frac{\lambda^n_{(N_\varphi-1)-\frac{1}{2}, N_z-1} + \lambda^n_{(N_\varphi-1)+\frac{1}{2}, N_z-1}}{r^2 \cdot h_\varphi^2} + \frac{\lambda^n_{N_\varphi-1, (N_z-1)-\frac{1}{2}} + \lambda^n_{N_\varphi-1, (N_z-1)+\frac{1}{2}}}{h_z^2} - \\
& \left. + \frac{\left(\lambda^n_{N_\varphi-1, (N_z-1)+\frac{1}{2}} \right)^2}{h_z^2 \cdot \left(\lambda^n_{N_\varphi-1, (N_z-1)+\frac{1}{2}} + \kappa \cdot h_z \right)} \right) \cdot T_{N_\varphi-1, N_z-1}^{n+1} + \frac{\lambda^n_{(N_\varphi-1)+\frac{1}{2}, N_z-1}}{r^2 \cdot h_\varphi^2} \cdot T_{2, N_z-1}^{n+1} = \\
& = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{N_\varphi-1, N_z-1}^n - \left(\frac{2\kappa}{h} + \frac{\lambda^n_{N_\varphi-1, (N_z-1)+\frac{1}{2}} \cdot \kappa}{h_z \cdot \left(\lambda^n_{N_\varphi-1, (N_z-1)+\frac{1}{2}} + \kappa \cdot h_z \right)} \right) \cdot T^0 - q_{N_\varphi-1, N_z-1}^n.
\end{aligned}$$

- $i = 3, 4, \dots, N_\varphi - 2, j = 2$

С учетом (8),

$$\begin{aligned}
& \frac{1}{r^2} \cdot \frac{\lambda^n_{i-\frac{1}{2}, 2}}{h_\varphi^2} \cdot T_{i-1, 2}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda^n_{i-\frac{1}{2}, 2} + \lambda^n_{i+\frac{1}{2}, 2}}{r^2 \cdot h_\varphi^2} + \frac{\lambda^n_{i, 2-\frac{1}{2}} + \lambda^n_{i, 2+\frac{1}{2}}}{h_z^2} - \right. \\
& \left. - \frac{\left(\lambda^n_{i, 1+\frac{1}{2}} \right)^2}{h_z^2 \cdot \left(\lambda^n_{i, 1+\frac{1}{2}} + \kappa \cdot h_z \right)} \right) \cdot T_{i2}^{n+1} + \frac{\lambda^n_{i, 2+\frac{1}{2}}}{h_z^2} \cdot T_{i3}^{n+1} + \frac{\lambda^n_{i+\frac{1}{2}, 2}}{r^2 \cdot h_\varphi^2} \cdot T_{i+1, 2}^{n+1} = \\
& = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{i2}^n - \left(\frac{2\kappa}{h} + \frac{\lambda^n_{i, 1+\frac{1}{2}} \cdot \kappa}{h_z \cdot \left(\lambda^n_{i, 1+\frac{1}{2}} + \kappa \cdot h_z \right)} \right) \cdot T^0 - q_{i2}^n.
\end{aligned}$$

- $i = 3, 4, \dots, N_\varphi - 2, j = N_z - 1$

С учетом (8),

$$\begin{aligned}
& \frac{1}{r^2} \cdot \frac{\lambda^n_{i-\frac{1}{2}, N_z-1}}{h_\varphi^2} \cdot T_{i-1, N_z-1}^{n+1} + \frac{\lambda^n_{i, (N_z-1)-\frac{1}{2}}}{h_z^2} \cdot T_{i, N_z-2}^{n+1} - \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda^n_{i-\frac{1}{2}, N_z-1} + \lambda^n_{i+\frac{1}{2}, N_z-1}}{r^2 \cdot h_\varphi^2} + \right. \\
& \left. + \frac{\lambda^n_{i, (N_z-1)-\frac{1}{2}} + \lambda^n_{i, (N_z-1)+\frac{1}{2}}}{h_z^2} - \frac{\left(\lambda^n_{i, (N_z-1)+\frac{1}{2}} \right)^2}{h_z^2 \cdot \left(\lambda^n_{i, (N_z-1)+\frac{1}{2}} + \kappa \cdot h_z \right)} \right) \cdot T_{i, N_z-1}^{n+1} + \frac{\lambda^n_{i+\frac{1}{2}, N_z-1}}{r^2 \cdot h_\varphi^2} \cdot T_{i+1, N_z-1}^{n+1} = \\
& = \left(\frac{2\kappa}{h} - \frac{\rho \cdot c}{\tau} \right) \cdot T_{i, N_z-1}^n - \left(\frac{2\kappa}{h} + \frac{\lambda^n_{i, (N_z-1)+\frac{1}{2}} \cdot \kappa}{h_z \cdot \left(\lambda^n_{i, (N_z-1)+\frac{1}{2}} + \kappa \cdot h_z \right)} \right) \cdot T^0 - q_{i, N_z-1}^n.
\end{aligned}$$

2.3. Выбор инструментария для разработки программного обеспечения

Основными требованиями к разрабатываемому программному обеспечению являются:

- скорость работы;
- визуализация данных, отображающих структуру стали в какой-то промежуток времени;
- наличие необходимого функционала для проведения компьютерных экспериментов.

На первом этапе разработки программного обеспечения, было необходимо выбрать язык программирования, на котором будет реализовано приложение. После чего будет решаться вопрос о выборе среды разработки и готовых, вспомогательных инструментов, предназначенных для разработки программного обеспечения.

Перечень языков программирования, из которого необходимо сделать выбор, включает:

- Python;
- C++;
- FORTRAN.

2.3.1. Анализ существующих программных средств для реализации программного обеспечения

Перед описанием всех альтернативных вариантов языков программирования, необходимо составить перечень требований, которые будут определять подходит ли данный вариант для разработки программного обеспечения, или нет.

С практической точки зрения, все варианты могут подойти, так как все они являются высокоуровневыми языками программирования общего назначения, но некоторые из них могут иметь уже готовые реализации необходимых для разработки методов. Поэтому перечень требований будет включать в себя как возможности язык программирования, так и готовые реализации необходимых методов на данном языке программирования.

Требования к языкам программирования:

- возможность решения систем линейных уравнений;
- методы для оптимизации негладких функций;
- методы для визуализации данных;
- возможность работы с различными законами распределения.

2.3.2. PYTHON

Python - высокоуровневый язык программирования общего назначения[9]. Python используется во многих проектах как основной язык программирования или для создания расширений и интеграции приложений.

В стандартной библиотеке включен большой объем полезных функций, а сам Python поддерживает несколько парадигм программирования, в нашем случае важна поддержка объектно–ориентированной. Python имеет поддержку многопоточных вычислений (Модуль Threading) и удобные высокоуровневые структуры данных. Язык имеет полную динамическую типизацию, и обладает автоматическим управлением памятью. Если на первый взгляд это может казаться преимуществом, то при разработке программ с повышенным требованием к эффективности, Python может значительно проигрывать по

скорости своим статическим братьям (C++, FORTRAN), что является основным минусом данного языка программирования [18].

Python располагает большой базой библиотек для линейной алгебры, оптимизации, интеграции и статистики (Numpy, Scipy, Scipy.optimize) [19].

Numpy:

NumPy – пакет для научных вычислений, который содержит:

- мощный N-мерный объект массива
- сложные (широковещательные) функции
- инструменты для интеграции кода C / C ++ и Fortran
- полезная линейная алгебра, преобразование Фурье и возможности случайных чисел

Помимо очевидных научных применений, NumPy можно также использовать в качестве эффективного многомерного контейнера общих данных. Произвольные типы данных могут быть определены. Это позволяет NumPy легко и быстро интегрироваться с широким спектром баз данных.

Scipy:

Еще одним из основных пакетов для разработки является библиотека SciPy, которая обеспечивает множество эффективных числовых процедур, таких как подпрограммы для численной интеграции и оптимизации.

Возможности:

- поиск минимумов и максимумов функций;
- вычисление интегралов функций;
- поддержка специальных функций;
- обработка сигналов;
- обработка изображений;
- работа с генетическими алгоритмами;
- решение обыкновенных дифференциальных уравнений;
- и др.

Scipy.optimize:

Пакет предоставляет несколько алгоритмов, обычно используемых оптимизаций.

Модуль содержит:

- неограниченная и ограниченная минимизация многомерных скалярных функций (`minimize`) с использованием множества алгоритмов (например, BFGS, симплекс Nelder-Mead, градиент Newton Conjugate, COBYLA или SLSQP)
- глобальные (грубые силы) процедуры оптимизации (например `basinhopping`, `differential_evolution`)
- алгоритмы минимизации наименьших квадратов (`least_squares`) и кривой (`curve_fit`)
- скалярные одномерные функции `minimizers` (`minimize_scalar`) и корневые искатели (`newton`)
- многовариантные решатели системы уравнений (`root`) с использованием множества алгоритмов (например, гибридный Пауэлл, Левенберг-Марквардт или крупномасштабные методы, такие как Ньютон-Крылов).

2.3.3. C++

C++ – компилируемый, статически типизированный язык программирования общего назначения [10].

C++ имеет богатую стандартную библиотеку и также как и Python поддерживает несколько парадигм программирования. Имеется поддержка многопоточности. Изучение подключаемых пакетов к стандартной библиотеки C++ показало, что необходимые методы и процедуры не реализованы. Но так как в данной работе основной упор ставится на использование готовых методов для реализации сложных вычислений, то данный факт является большим минусом в сторону данного языка программирования.

Основными минусами данного языка программирования являются:

- долгая компиляция;

- большой объем сгенерированного машинного кода;
- C++ является сложным для изучения;
- отсутствие системы сборки мусора;
- плохая поддержка модульности.

2.3.4. FORTRAN

FORTRAN также как и C++, Python является языком программирования высокого уровня. Современный FORTRAN (Fortran 95 и Fortran 2003) приобрёл черты, необходимые для эффективного программирования, для новых вычислительных архитектур; позволяет применять современные технологии программирования, в частности, обобщённое и модульное программирование, ООП, сохраняя при этом преемственность с более ранними версиями[11]. Широкое применения и использование данного языка приходится в первую очередь на научные и инженерные вычисления. Так же к безусловным плюсам данного языка программирования относится, скорость работы, которая превосходит даже C++.

Одно из преимуществ современного FORTRAN - большое количество написанных на нём программ и библиотек подпрограмм. Имеет поддержку многопоточных вычислений. Язык предоставляет готовые методы для оптимизации негладких функций. FORTRAN располагает методами для работы с различными функциями распределения.

С технической точки зрения FORTRAN является отличным инструментом для разработки научно-технических прикладных программ, но его практически не используют, так как считается, что на данный момент он не актуален в отличии от своих конкурентов – Python, C++.

2.3.5. Вывод по выбору инструментария для разработки программного обеспечения

После изучения выше описанных инструментариев для разработки программного обеспечения. Описав основные преимущества и недостатки данных инструментариев, можно сделать вывод, что, Python и FORTRAN

больше подходят для разработки будущего программного обеспечения, но стоит учитывать тот факт, что разработчики на данный момент склоняются к выбору в пользу Python-а, который не уступает в количестве готовых методов, реализованных на данных языках. По итогу, для разработки программного обеспечения, будет использоваться Python.

2.4. Реализация моделирования тепловых полей.

Для проведения компьютерных экспериментов, была разработана программа для моделирования тепловых полей на равномерной сетке. Программа позволяет построить равномерную сетку, и провести расчеты температурных полей по времени. Как описано выше, необходимо построить систему линейных уравнений и решить ее. Все данные по температурам, записываются в текстовый файл для дальнейшего использования полученных данных. Алгоритм расчета температурных полей представлен на рисунке 2.

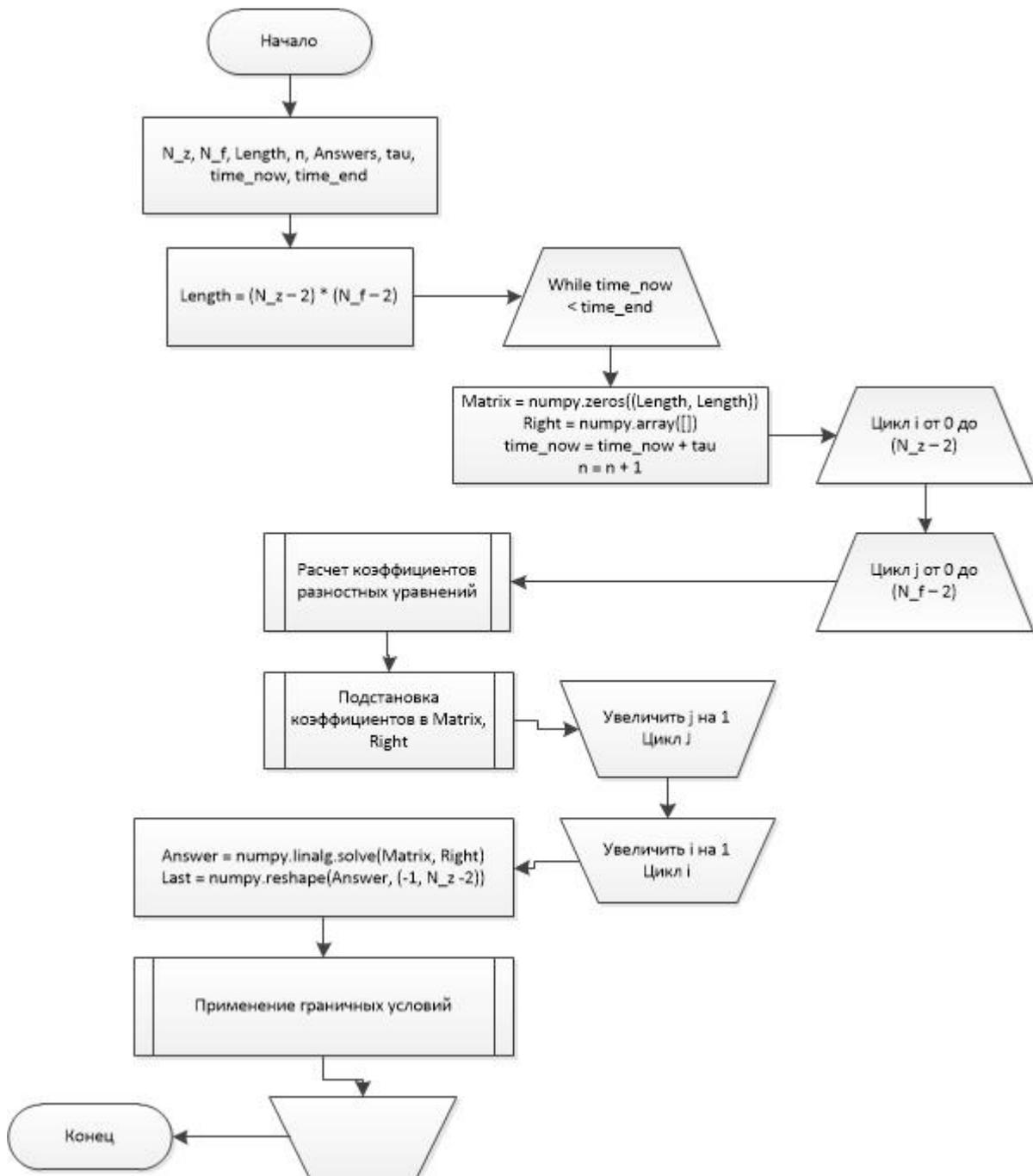


Рисунок 2. Алгоритм расчета температурных полей.

Алгоритм программы:

- Расчет внутренних параметров программы.
- Определение переменных внутри программы:
 - Шаг по времени (tau) : шаг по длине трубы/ скорость лазера;
 - Количество шагов по длине трубы (N_z) : длина трубы / шаг по длине трубы;

- Шаг по сечению трубы (h_f) : $\text{ПИ} \cdot \text{диаметр трубы} / \text{количество шагов по сечению трубы}$;
 - Плотность потока (q) : $\text{мощность лазера} / \text{шаг по сечению трубы} / \text{шаг по длине трубы} / \text{толщина трубы}$;
 - Время нагревания (timeWarm) : $\text{количество шагов по длине трубы} \cdot \text{шаг по времени}$;
 - Массив ответов (answers);
- Формирование начальной матрицы.
Создание матрицы $N_f \times N_z$, где значение каждой ячейке равно T_0 .
Добавление матрицы к массиву ответов.
 - Расчет температуры на первом шаге.
Создание матрицы $(N_z-2 \times N_f-2) \times (N_z-2 \times N_f-2)$, значения в которой будут являться коэффициентами линейных уравнений.
Создание пустого массива, значения которого будут представлять собой значение правой части линейных уравнений.
Прогонка цикла, где происходит заполнение ячеек матрица при помощи метода, возвращающего коэффициенты линейного уравнения и номера колонок, куда необходимо вставить коэффициенты в матрице.
Решение системы уравнений при помощи метода `numpy.linalg.solve()`.
Преобразование одномерного массива ответов, полученного после работы метода `numpy.linalg.solve()`, в двумерный массив.
Создание массива $N_f \times N_z$, для записи конечных результатов, температуры в разных узлах сетки. В центр конечной матрицы вставляем результаты преобразованный в двумерный массив, после чего заполняем границы матрицы.
Полученную конечную матрицу добавляем к массиву ответов. Запускаем цикл с условием, где условием будет являться не достигло ли текущее время времени нагрева.
В цикле проводим те же операции что и в пункте 4, изменяя при этом текущее время. Таким образом, программа дойдет до момента, когда пройдет время

нагревания, и будет заполнен массив ответов, где каждый элемент массива будет представлять собой матрицу, в ячейках которой будут значения температуры.

Запускаем цикл с условием, где условием будет являться достижение минимальной температуры.

В цикле проводим те же операции что и в пункте 4, изменяя текущее время и высчитывая минимальную температуру последней полученной матрицы с температурами. Таким образом, программа дойдет до момента, когда минимальное значение температуры в последней полученной матрице, будет меньше или равно установленной минимальной температуре.

- Далее происходит запись в первый файл, куда попадают все параметры, при которых работала программа, и сохранение этого файла.
- Формирование файла с ответами, полученными в ходе работы программы.

Структура программы представлена на диаграмме классов (рисунок 3):

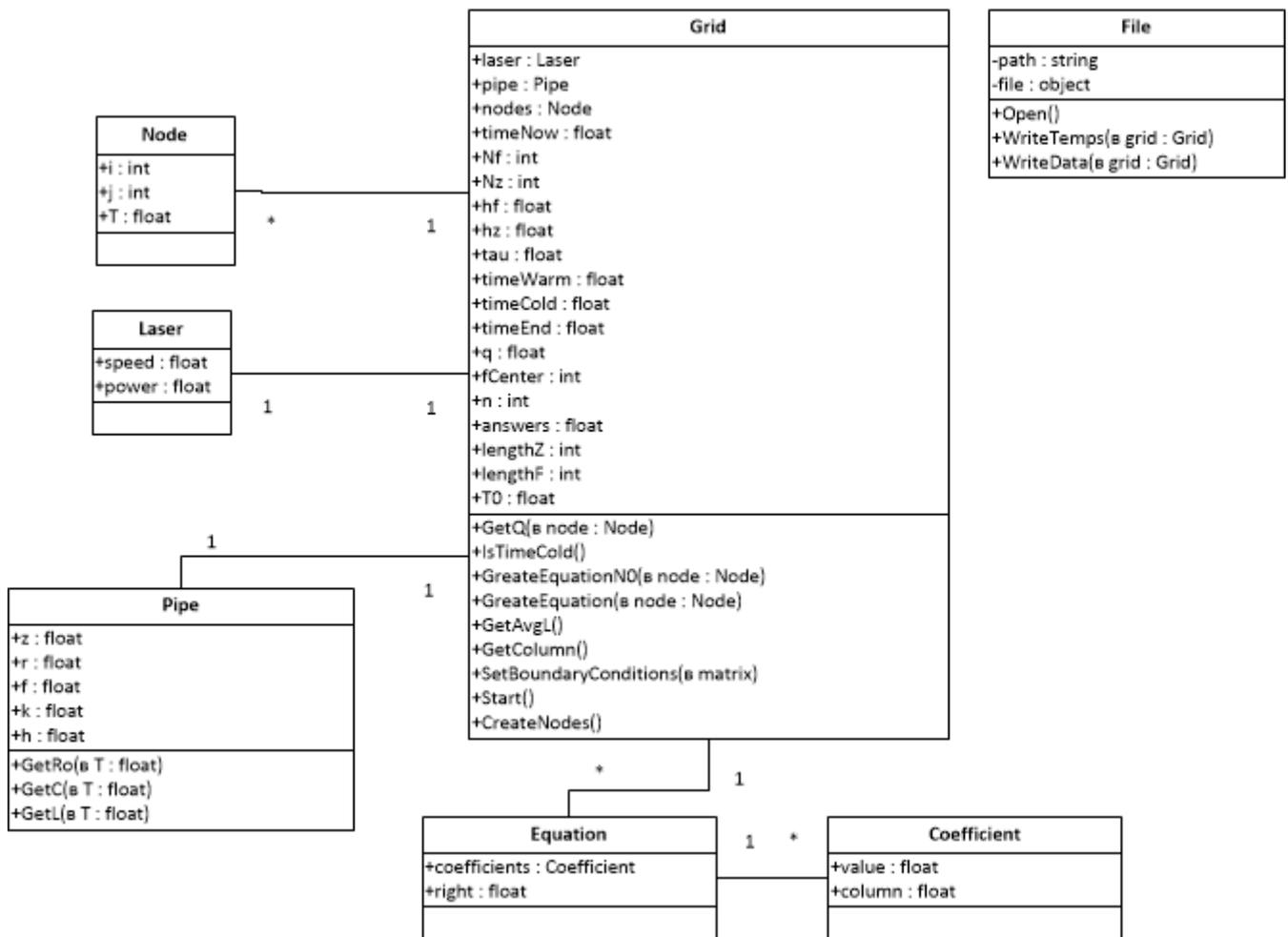


Рисунок 3. Диаграмма классов программы для расчета тепловых полей.

Таблица 1. Класс «Node»

Node	
Атрибуты	
i	Номер узла по оси z
j	Номер узла по оси φ
T	Температура в данном узле
Методы	
Название	Описание

Таблица 2. Класс «Laser»

Laser	
Атрибуты	
speed	Скорость лазера
power	Мощность лазера
Методы	
Название	Описание

Таблица 3. Класс «File»

File	
Атрибуты	
path	Путь к файлу
file	Объект для работы с файлом
Методы	
Название	Описание
Open	Метод для открытия файла на редактирование
WriteTemps	Метод для записи значений температурных полей в файл
WriteData	Метод для записи входных данных эксперимента в файл

Таблица 4. Класс «Pipe»

Pipe	
Атрибуты	
z	Длина трубы
r	Радиус трубы
f	Длина окружности трубы
k	Коэффициент теплоотдачи
h	Толщина трубы
Методы	
Название	Описание
GetRo	Возвращает значение плотности
GetC	Возвращает значение теплоемкости
GetL	Возвращает значение теплопроводности

Таблица 5. Класс «Equation»

Equation	
Атрибуты	
coefficients	Массив с объектами класса Coefficient (таблица 6)
right	Значение в правой части уравнения
Методы	
Название	Описание

Таблица 6. Класс «Coefficient»

Coefficient	
Атрибуты	
value	Значение коэффициента
column	Номер колонки соответствующий данному коэффициенту
Методы	
Название	Описание

Таблица 7. Класс «Grid»

Grid	
Атрибуты	
laser	Массив с объектами класса Laser (таблица 2)
pipe	Массив с объектами класса Pipe (таблица 4)
nodes	Массив с объектами класса Node (таблица 1)
timeNow	Текущее время
timeWarm	Время продолжительности нагревания
timeCold	Время продолжительности охлаждения
timeEnd	Время завершения моделирования
Nf	Количество шагов по φ
Nz	Количество шагов по z
hf	Шаг по φ
hz	Шаг по z
tau	Шаг по времени
q	Начальная плотность потока
fCenter	Индекс центрального узла по φ
n	Индекс по времени
answers	Массив с температурными полями во времени
lengthZ	Размерность матрицы по z
lengthF	Размерность матрицы по φ
T0	Начальная температура

Методы	
Название	Описание
GetQ	Возвращает значение плотности потока лазера
IsTimeCold	Возвращает Истину если наступил момент времени охлаждения
CreateEquationN0	Создание уравнений для нулевого шага по времени
CreateEquation	Создание уравнений
GetAvgL	Возвращает среднее значение теплопроводности
GetColumn	Возвращает номер колонки для узла
SetBoundaryConditions	Вычисление температур для узлов на границах
Start	Метод для запуска моделирования
CreateNodes	Метод для создания всех узлов равномерной сетки

2.5. Результаты моделирования тепловых полей.

Моделирование процесса нагревания зоны сварного шва сфокусированным лазерным лучом, а также ее последующего охлаждения выполнялось для различных значений мощности излучения в пределах от 1 до 6 кВт и значений скорости движения лазерного луча от 0,12 м/мин до 3,0 м/мин.

Расчеты выполнялись для трубы диаметром 10 см толщиной от 2 до 10 мм. Предполагалось, что при указанных значениях толщины стенки трубы процесс распределения тепла будет адекватно описываться двумерной моделью, описанной выше, соотношениями (5), (7), (8). Значение плотности мощности излучения q в точке сварки определялось с учетом расстояния между узлами пространственной сетки как

$$q = \frac{Q}{h \cdot (h_\varphi \cdot r) \cdot h_z},$$

где Q – мощность лазерного источника.

Для трубы рассматриваемого диаметра базовое значение шага по φ , используемое при решении уравнений (7), составило $h_\varphi = 0,017$ рад. (1°), что соответствует линейному расстоянию по окружности трубы 0,87 мм; значение h_z шага по z варьировалось в зависимости от скорости движения лазерного луча.

Значения плотности материала, удельной теплоемкости и теплопроводности задавались, исходя из физических характеристик стали 12Х18Н10Т в соответствии с таблицей 8.

Таблица 8. Физические характеристики стали 12Х18Н10Т.

Физические свойства стали 12Х18Н10Т			
T (°C)	λ (Вт/(м·°C))	ρ (кг/м ³)	c (Дж/(кг·°C))
20	15	7920	
100	16		462
200	18		496
300	19		517
400	21		538
500	23		550
600	25		563
700	27		575
800	26		596

Для каждого из моделируемых режимов сварки рассчитывались температурные поля (значения температуры для всех узлов пространственной сетки) в зависимости от времени. Анализировались максимальные значения температуры, достигаемые в зоне сварного шва и околошовной зоне, а также скорости охлаждения на различных расстояниях от линии движения лазера.

Далее будет рассмотрен пример полученных результатов при толщине стержня 2мм.

Для трубы толщиной 2 мм расчеты выполнялись при следующих значениях мощности лазера и скорости движения лазерного луча:

- $Q = 1$ кВт; $v = 0,5$ м/мин;
- $Q = 2$ кВт; $v = 1$ м/мин;
- $Q = 3$ кВт; $v = 2$ м/мин;

- $Q = 5$ кВт; $v = 3$ м/мин.

На рисунках 4 – 6 показаны результаты моделирования температурных полей при $Q = 1$ кВт и $v = 0,5$ м/мин.

Рисунок 4 показывает зависимость температуры от времени в точке, лежащей на линии сварного шва.

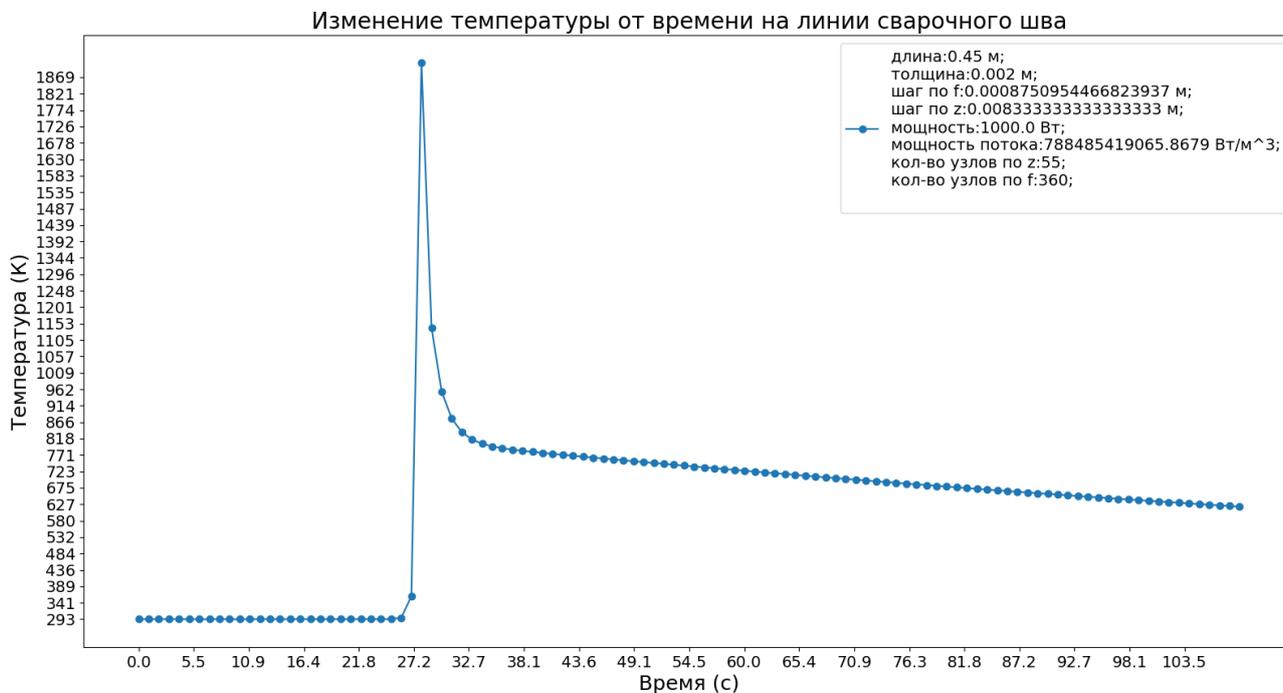


Рисунок 4. Зависимость температуры от времени на линии сварного шва. Толщина трубы 2 мм, мощность лазера 1 кВт, скорость движения 0,5 м/мин.

На рисунке 5 представлена аналогичная зависимость для точки, отстоящей от линии сварки.

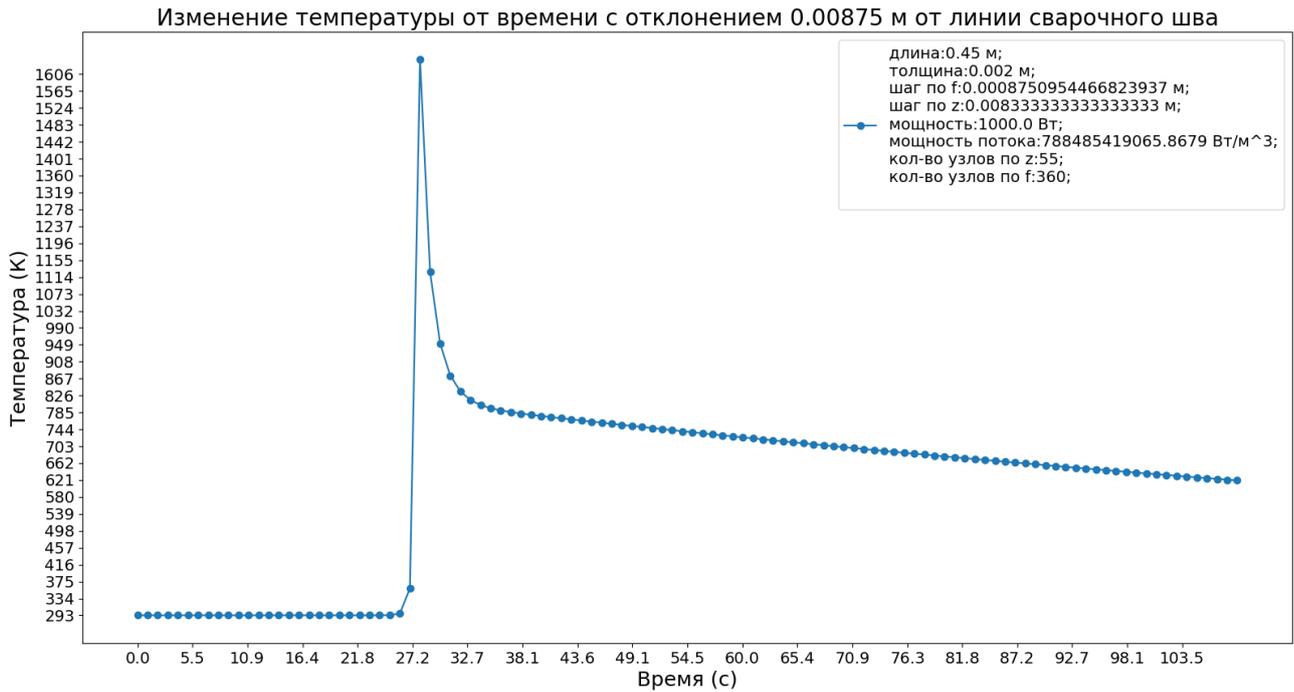


Рисунок 5. Зависимость температуры от времени на расстоянии от линии сварного шва. Толщина трубы 2 мм, мощность лазера 1 кВт, скорость движения 0,5 м/мин.

На рисунке 6 представлен график изменения температуры вдоль поперечной линии, проходящей через точку максимального нагрева.

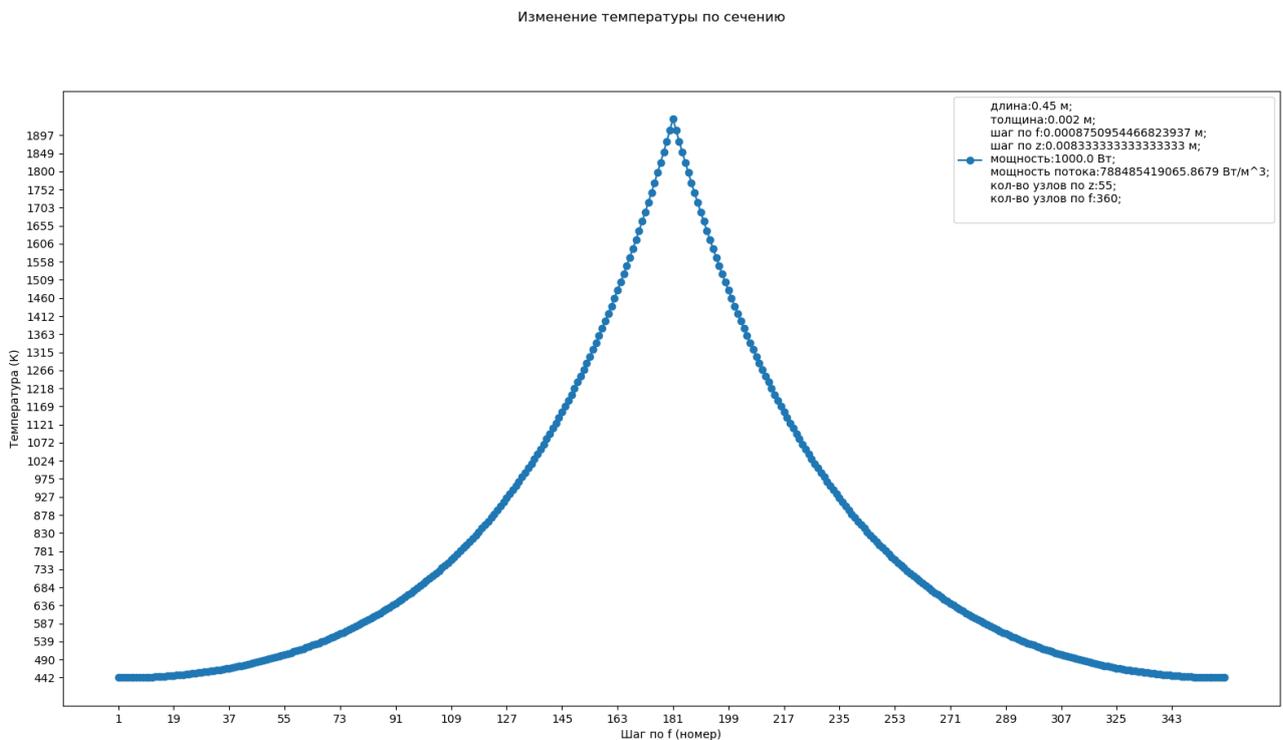


Рисунок 6. Изменение температуры по сечению трубы. Толщина трубы 2 мм, мощность лазера 1 кВт, скорость движения 0,5 м/мин.

На рисунках 7 – 9 показаны результаты моделирования температурных полей при $Q = 2$ кВт и $v = 1$ м/мин.

Рисунок 7 показывает зависимость температуры от времени в точке, лежащей на линии сварного шва.

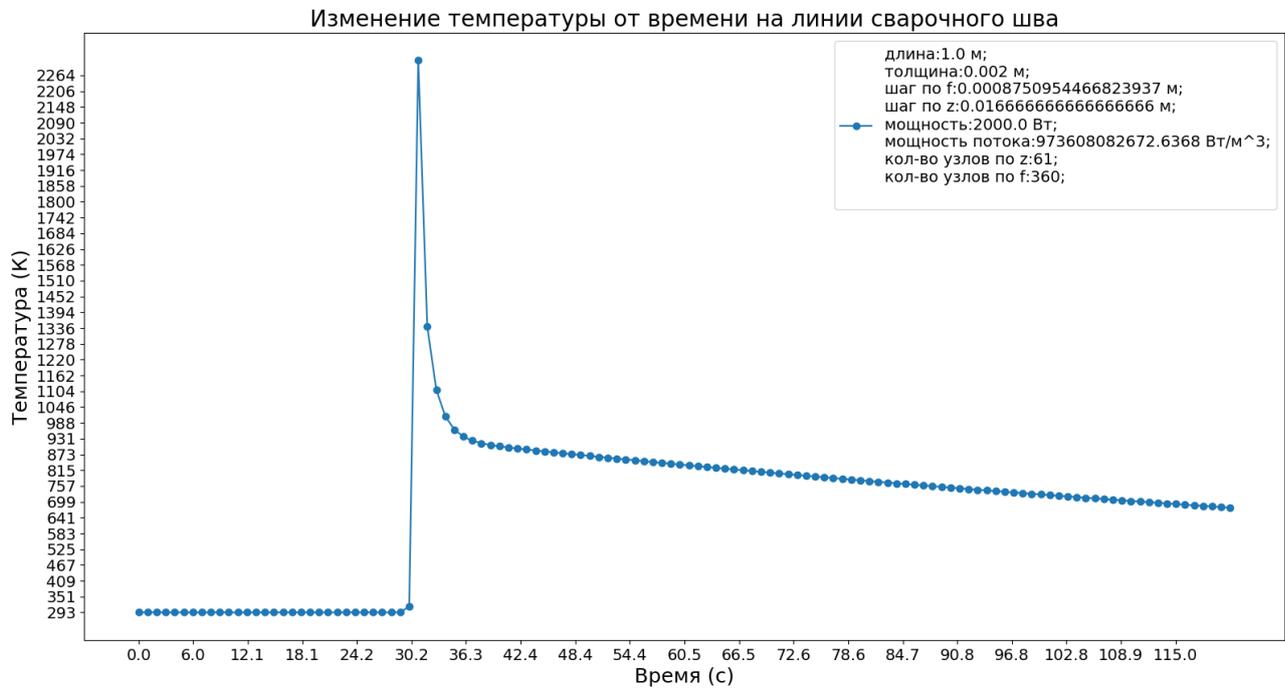


Рисунок 7. Зависимость температуры от времени на линии сварного шва. Толщина трубы 2 мм, мощность лазера 2 кВт, скорость движения 1 м/мин.

На рисунке 8 представлена аналогичная зависимость для точки, отстоящей от линии сварки.

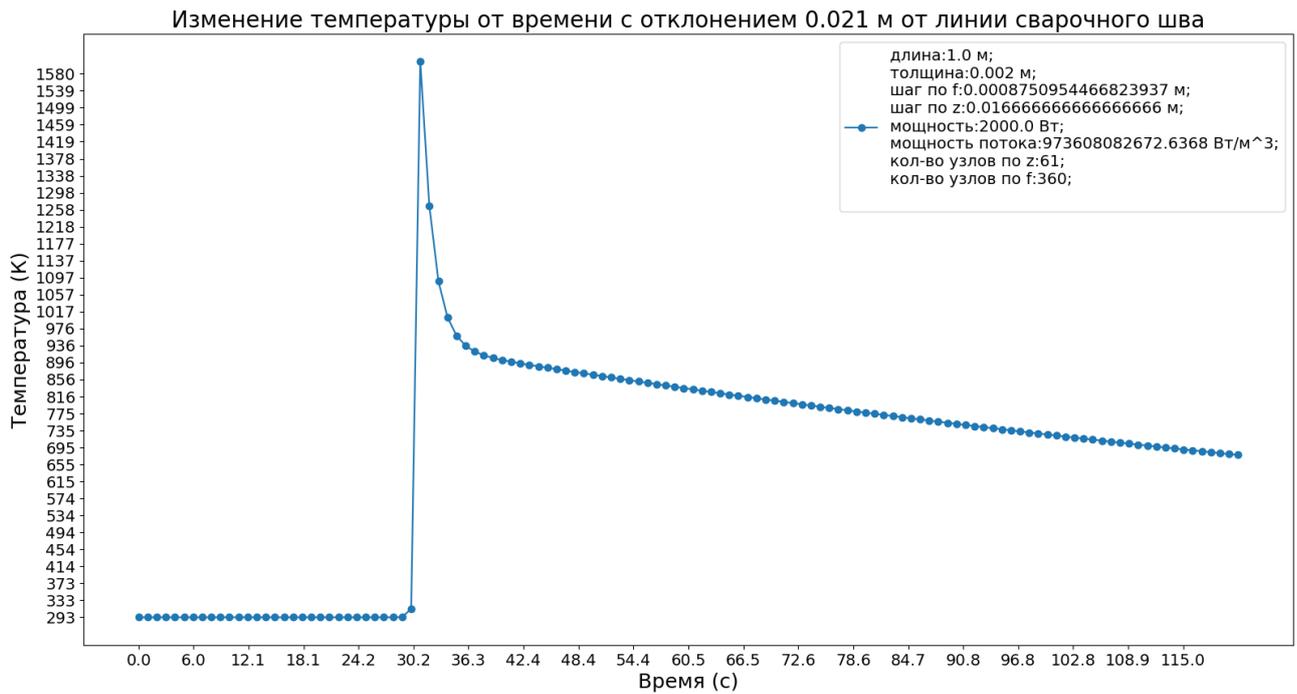


Рисунок 8. Зависимость температуры от времени на расстоянии от линии сварного шва. Толщина трубы 2 мм, мощность лазера 2 кВт, скорость движения 1 м/мин.

На рисунке 9 представлен график изменения температуры вдоль поперечной линии, проходящей через точку максимального нагрева.

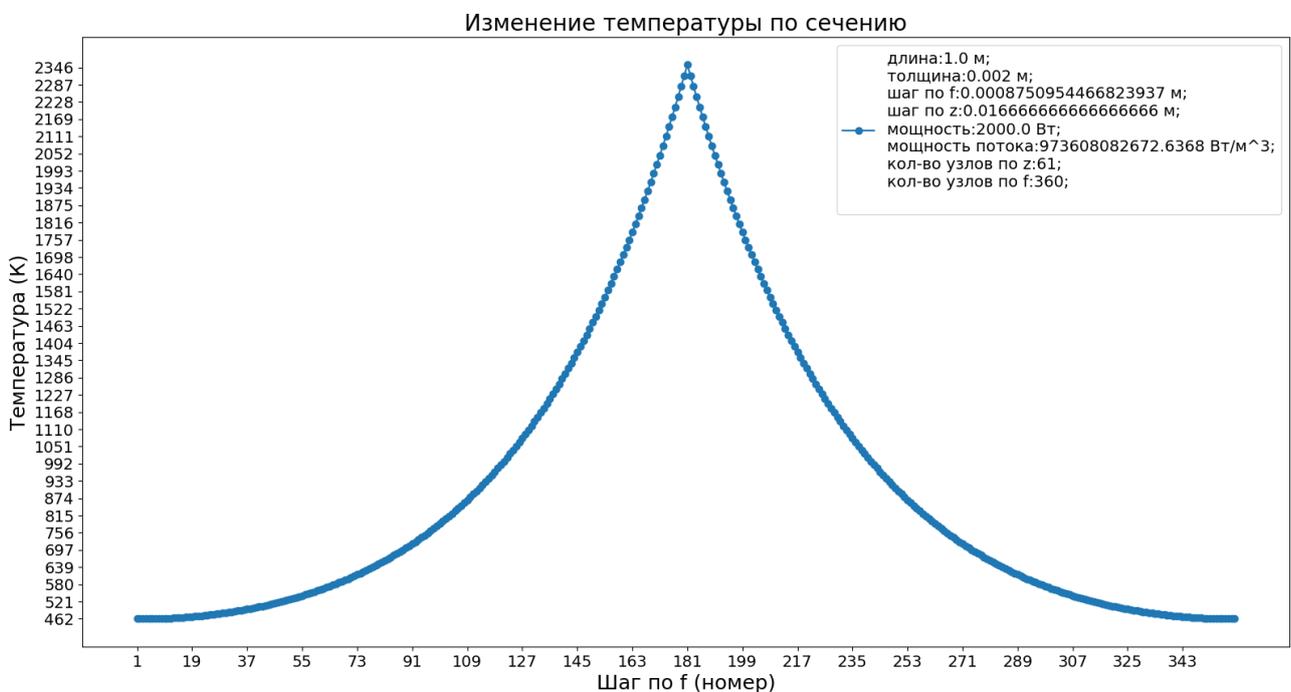


Рисунок 9. Изменение температуры по сечению трубы. Толщина трубы 2 мм, мощность лазера 2 кВт, скорость движения 1 м/мин.

На рисунках 10 – 12 показаны результаты моделирования температурных полей при $Q = 3$ кВт и $v = 2$ м/мин.

Рисунок 10 показывает зависимость температуры от времени в точке, лежащей на линии сварного шва.

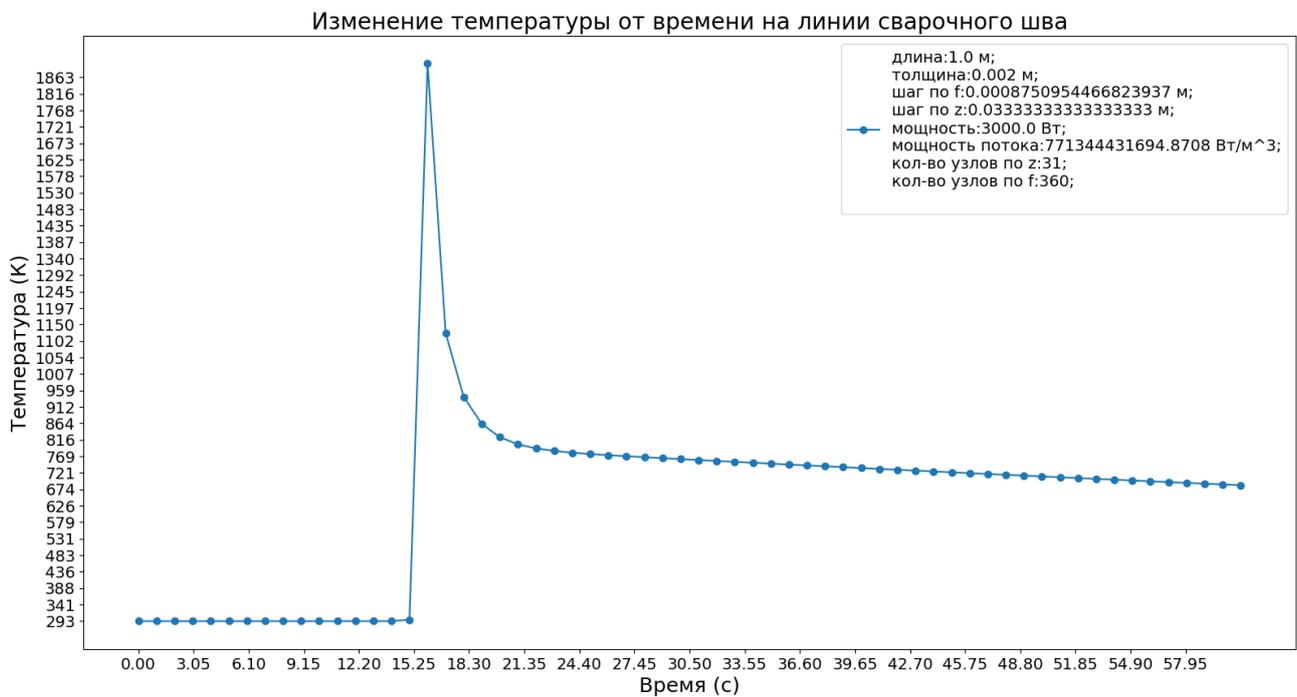


Рисунок 10. Зависимость температуры от времени на линии сварного шва. Толщина трубы 2 мм, мощность лазера 3 кВт, скорость движения 2 м/мин.

На рисунке 11 представлена аналогичная зависимость для точки, отстоящей от линии сварки.

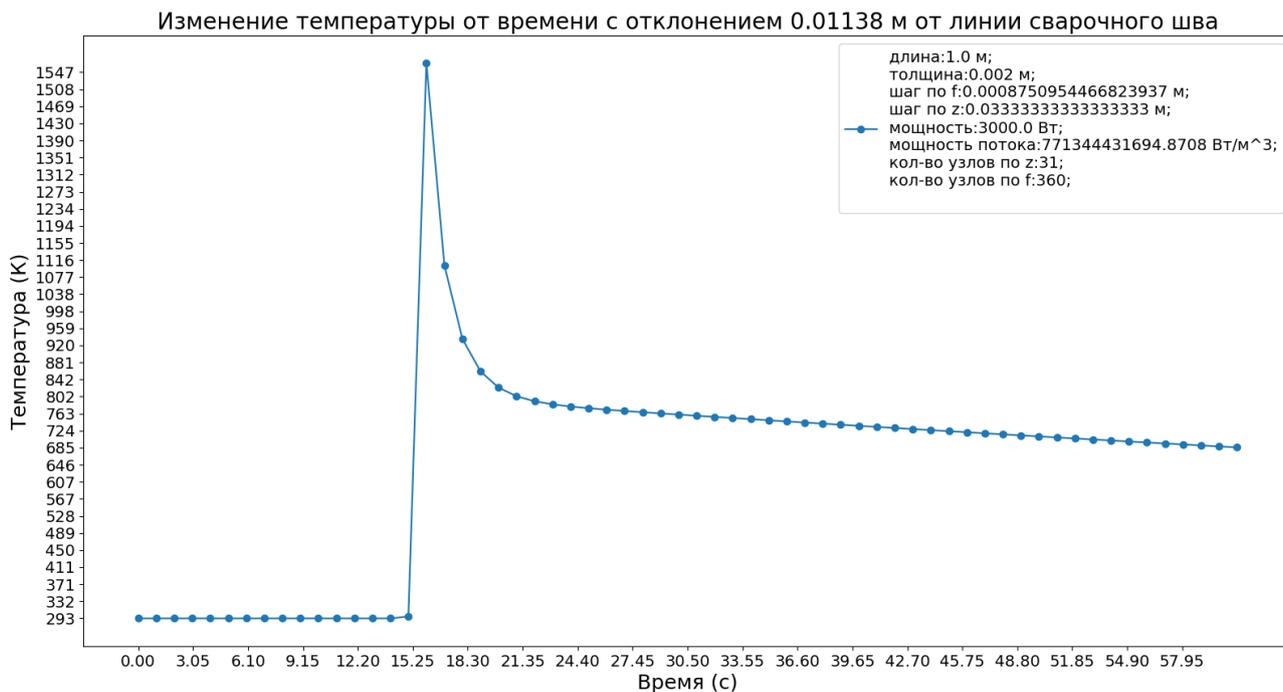


Рисунок 11. Зависимость температуры от времени на расстоянии от линии сварного шва. Толщина трубы 2 мм, мощность лазера 3 кВт, скорость движения 2 м/мин.

На рисунке 12 представлен график изменения температуры вдоль поперечной линии, проходящей через точку максимального нагрева.

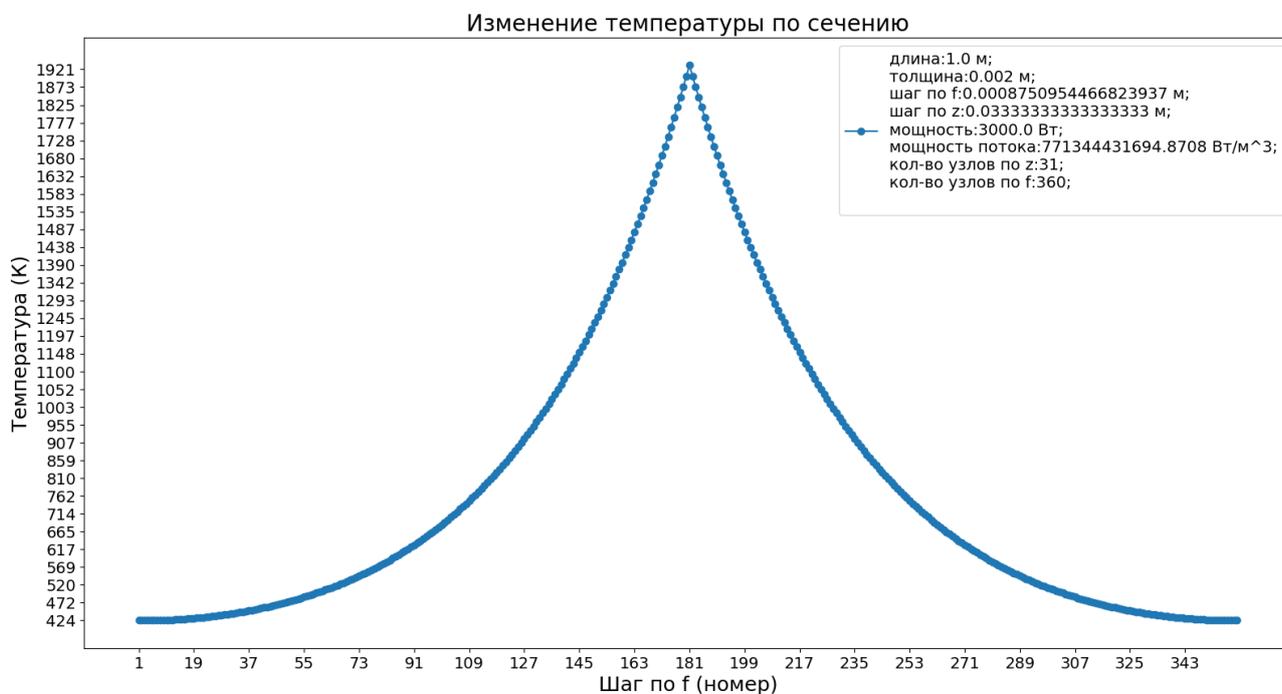


Рисунок 12. Изменение температуры по сечению трубы. Толщина трубы 2 мм, мощность лазера 3 кВт, скорость движения 2 м/мин.

На рисунках 13 – 15 показаны результаты моделирования температурных полей при $Q = 5$ кВт и $v = 3$ м/мин.

Рисунок 13 показывает зависимость температуры от времени в точке, лежащей на линии сварного шва.

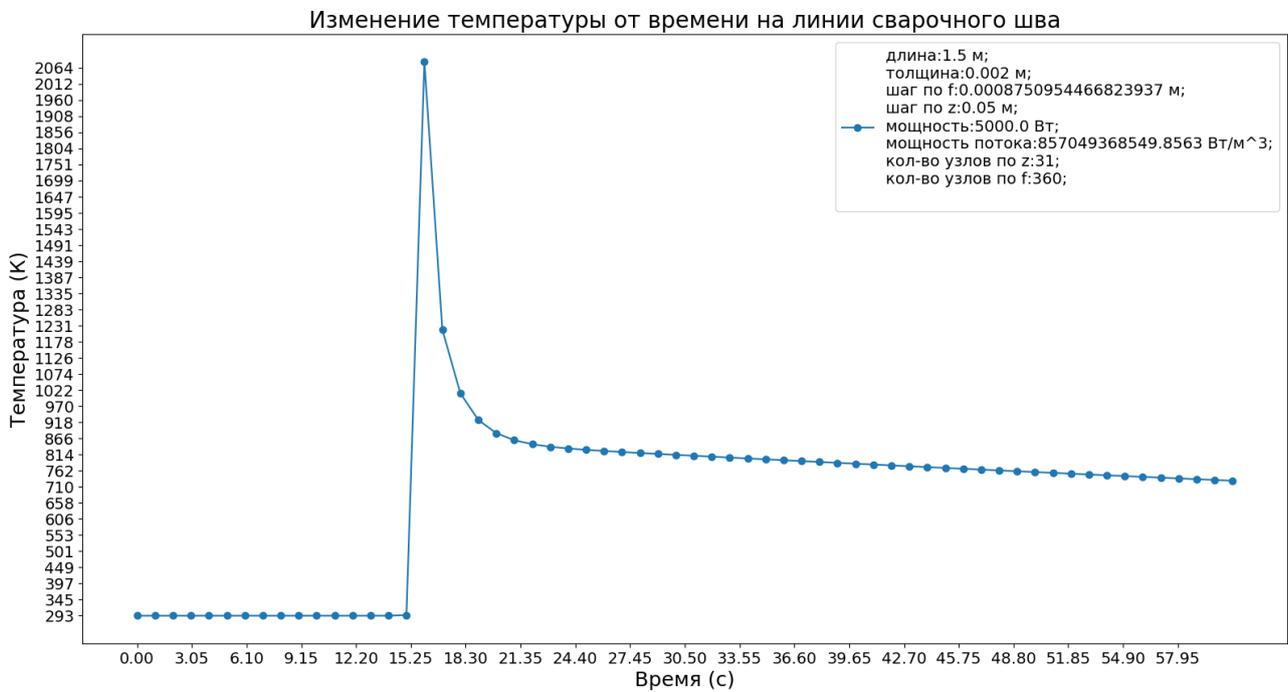


Рисунок 13. Зависимость температуры от времени на линии сварного шва. Толщина трубы 2 мм, мощность лазера 5 кВт, скорость движения 3 м/мин.

На рисунке 14 представлена аналогичная зависимость для точки, отстоящей на 16 мм от линии сварки.

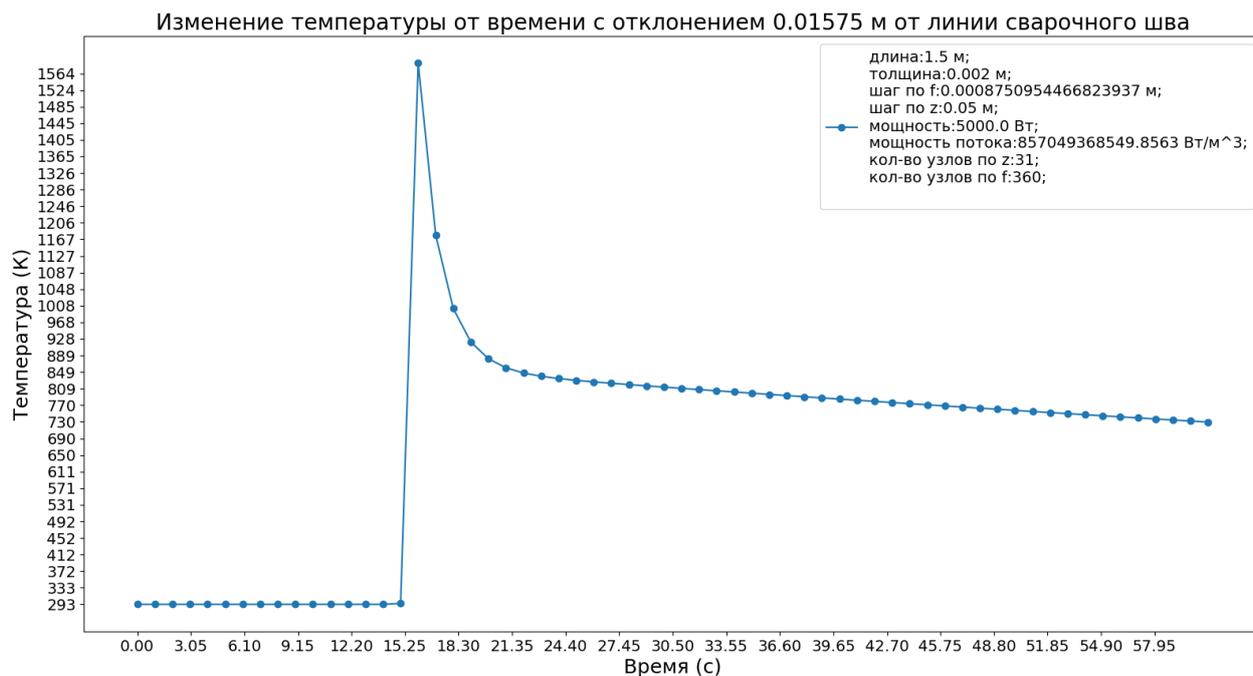


Рисунок 14. Зависимость температуры от времени на расстоянии от линии сварного шва. Толщина трубы 2 мм, мощность лазера 5 кВт, скорость движения 3 м/мин.

На рисунке 15 представлен график изменения температуры вдоль поперечной линии, проходящей через точку максимального нагрева.

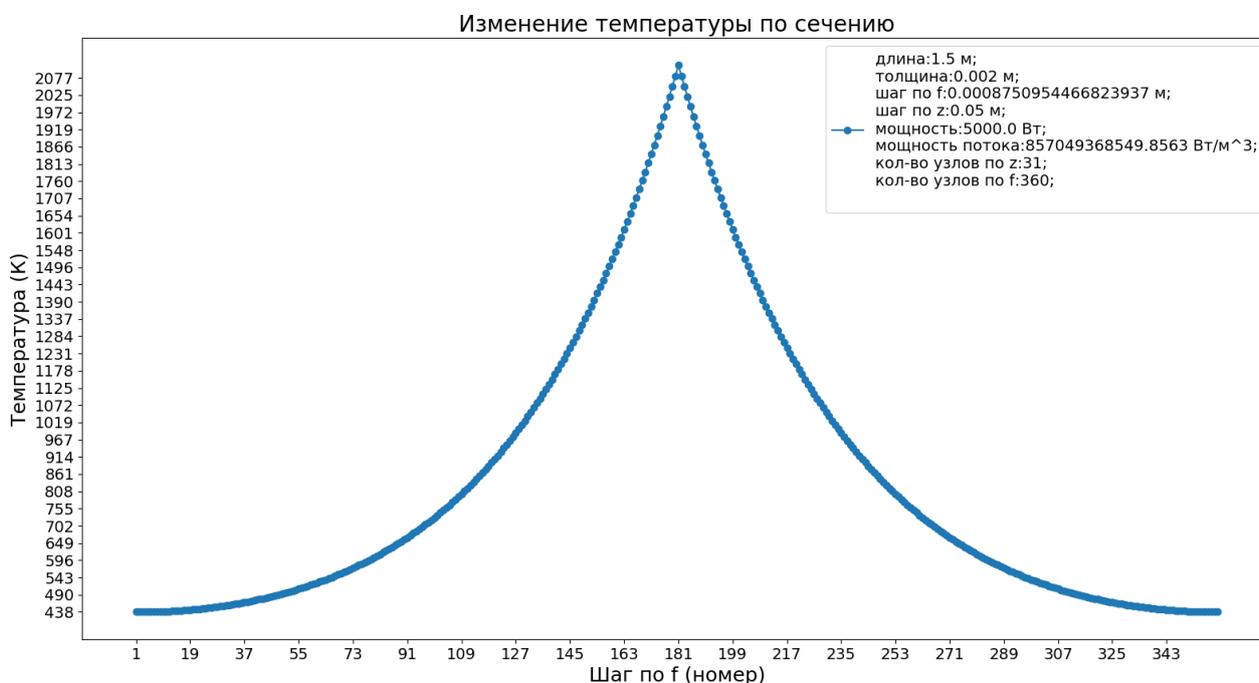


Рисунок 15. Изменение температуры по сечению трубы. Толщина трубы 2 мм, мощность лазера 5 кВт, скорость движения 3 м/мин.

2.6. Выводы по результатам, полученным на равномерной сетке

На рис. 4, 7, 10 и 13 на линии сварного шва можно заметить резкий скачок температуры в районе ее максимального значения. Это связано с тем, что в объеме трубы построена равномерная сетка, причем величина шага по времени ограничена снизу расстоянием между узлами сетки по оси z . При моделировании фазовых превращений в зоне сварного шва наблюдаемые скачки температуры могут привести к большой погрешности прогнозирования фазового состава. В рамках модели тепловых полей на равномерной сетке указанную проблему можно было бы решить путем уменьшения расстояния между узлами сетки. Однако это автоматически приведет к увеличению размерности задачи и, следовательно, к резкому увеличению вычислительных ресурсов, необходимых для решения этой задачи. С учетом ограниченности ресурсов такое решение является неприемлемым.

Альтернативным способом решения данной проблемы является применение адаптивной сетки для построения системы разностных уравнений теплопроводности. Идея такого подхода описана в работе [27], где представлена схема построения адаптивной сетки и соответствующих уравнений для двумерного случая, а также примеры применения данного подхода.

Еще одно наблюдение, которое можно сделать на основании полученных результатов, состоит в следующем. Предположение о допустимости применения двумерной модели (т. е., фактически, о постоянстве температуры по толщине трубы) можно считать приемлемым только для труб небольшой толщины (порядка 2-4 мм). Для более толстой трубы следует использовать трехмерную модель.

В связи с этим, было принято решение использовать идею построения модели на адаптивной сетке, описанную в [27], модифицировав сетку и уравнения для трехмерного случая.

3. МОДЕЛЬ ТЕПЛОВЫХ ПОЛЕЙ НА АДАПТИВНОЙ СЕТКЕ

3.1. Схема построения адаптивной сетки

Решение, предложенное в [27], применяется для замкнутой области в декартовых или цилиндрических координатах:

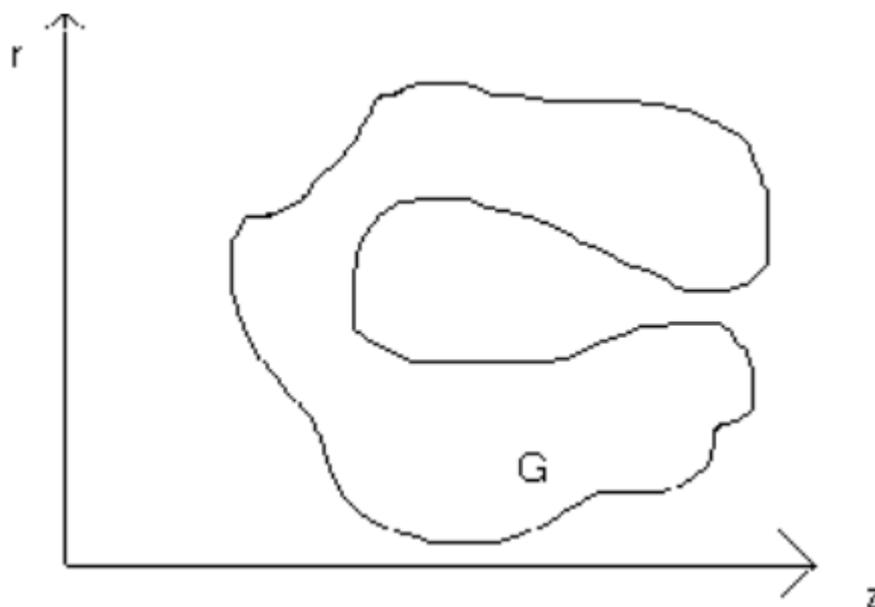


Рисунок 16. Пример области для решения уравнения теплопроводности [27].

В работе [27] были введены следующие обозначения:

T – температура;

$\varepsilon(T)$ – удельная внутренняя энергия;

$\chi(T)$ – коэффициент теплопроводности;

r, z – координаты;

n – единичный вектор нормали к поверхности Γ ;

$v = 0$ – обозначает декартовы координаты;

$v = 1$ – цилиндрические координаты;

Γ – граница двумерной замкнутой области G ;

$A(T_\Gamma), B(T_\Gamma)$ – произвольные функции на границе области.

Разностными уравнениями аппроксимируется исходная система уравнений.

Построение адаптивной сетки происходит в следующие три этапа:

1. Предварительное разбиение, где изначально задается область и количество разбиений.

2. Если в области G задана подобласть, то все ячейки, попавшие в эту подобласть делятся на определенное количество частей, до того момента, пока не будет достигнута заданная минимальная размерность ячейки.

3. На данном шаге происходит проверка условия на количество соседей с каждой стороны новой ячейки (не более двух).

В результате постоянного разбиения ячеек возникает проблема с их нумерацией, вследствие чего приходится постоянно перенумеровывать полученные ячейки. Помимо нумерации, необходимо производить переопределение соседей с каждой стороны. Поэтому при построении системы разностных линейных уравнений получается разреженная матрица без определённой структуры. На рисунке 17 показан пример дробления сетки в круге радиуса $R = 3$ см.

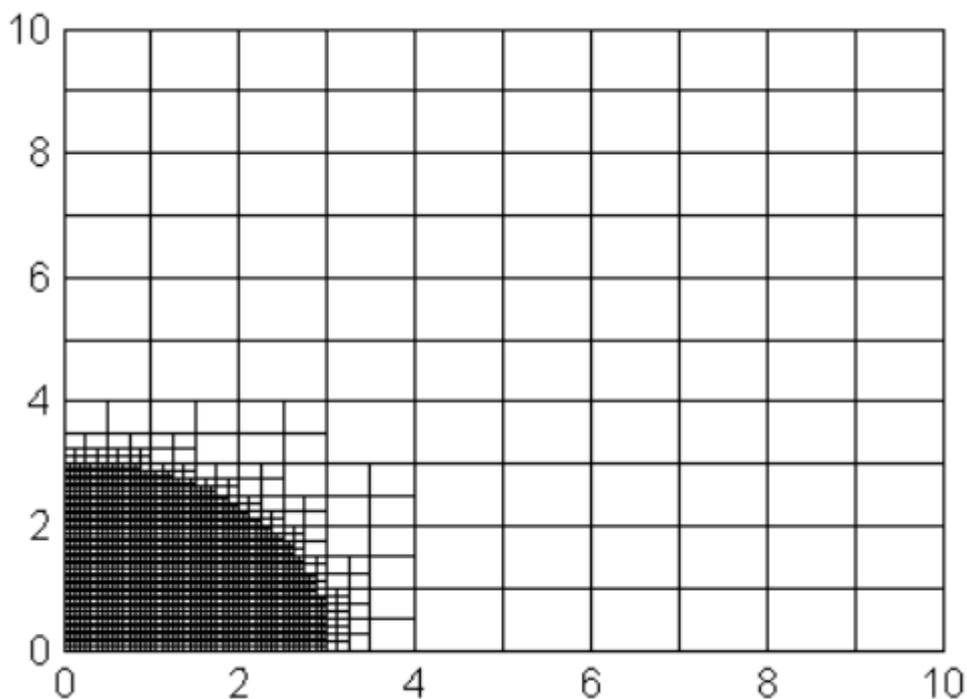


Рисунок 17. Пример построения сетки

Разностная схема для оператора дивергенции от градиента температуры в каждой ячейке имеет различный вид в зависимости от состояния сетки. В работе [27]

рассматривается 10 различных комбинаций. На рисунках 17 – 20 представлены комбинации, необходимые для моделирования интересующего нас случая, когда измельчение ячеек сетки происходит вдоль линии сварного шва. В написании формул используются следующие обозначения h – размер ячейки, $\tilde{h}_i = 0.5 (r_i^2 - r_{i-1}^2)$ – в цилиндрических координатах, $\tilde{h}_i = h$ – в декартовых координатах, где r_i, r_{i-1} – верхняя и нижняя граница i -ой ячейки.

Комбинация №1:

$$\frac{\varepsilon(T^{n+1}) - \varepsilon(T^n)}{\tau} - \left(\frac{\chi(T_4^{n+1}) + \chi(T_1^{n+1})}{2} \frac{T_4^{n+1} - T_1^{n+1}}{h^2} - \frac{\chi(T_3^{n+1}) + \chi(T_1^{n+1})}{2} \frac{T_1^{n+1} - T_3^{n+1}}{h^2} \right) -$$

$$- (r_1^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_2^{n+1})}{2} \frac{T_2^{n+1} - T_1^{n+1}}{0.5 \cdot \tilde{h}_1 \cdot (\tilde{h}_2 + \tilde{h}_1)} - r_5^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_5^{n+1})}{2} \frac{T_1^{n+1} - T_5^{n+1}}{0.5 \cdot \tilde{h}_1 \cdot (\tilde{h}_5 + \tilde{h}_1)}) = 0 \quad (10)$$

Где: T_k – значение температуры в k -ой ячейке в соответствии с рисунком 18.

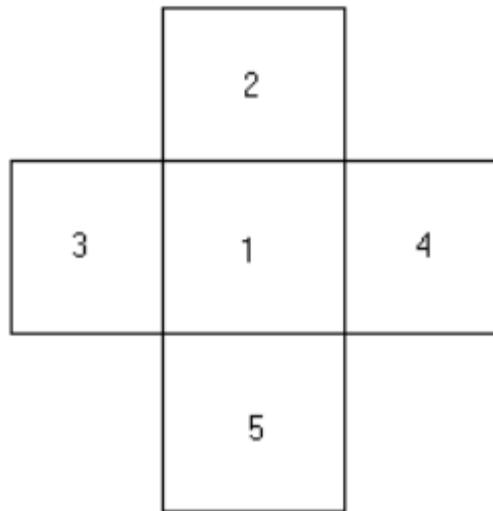


Рисунок 18. Расположение внутренних ячеек с равномерной сеткой.

Комбинация №2:

$$\begin{aligned}
& \frac{\varepsilon(T^{n+1}) - \varepsilon(T^n)}{\tau} - \left(\frac{\chi(T_5^{n+1}) + \chi(T_1^{n+1})}{2} \cdot \frac{T_5^{n+1} - T_1^{n+1}}{h^2} - \right. \\
& \left. - \frac{\chi(T_3^{n+1}) + \chi(T_1^{n+1}) + \chi(T_4^{n+1})}{3} \cdot \frac{T_1^{n+1} - 0,5(T_3^{n+1} + T_4^{n+1})}{h^2} \right) - \\
& - (r_1^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_2^{n+1})}{2} \frac{T_2^{n+1} - T_1^{n+1}}{0,5 \cdot \tilde{h}_1 \cdot (\tilde{h}_2 + \tilde{h}_1)} - \\
& - r_6^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_6^{n+1})}{2} \frac{T_1^{n+1} - T_6^{n+1}}{0,5 \cdot \tilde{h}_1 \cdot (\tilde{h}_5 + \tilde{h}_1)}) = 0.
\end{aligned} \tag{11}$$

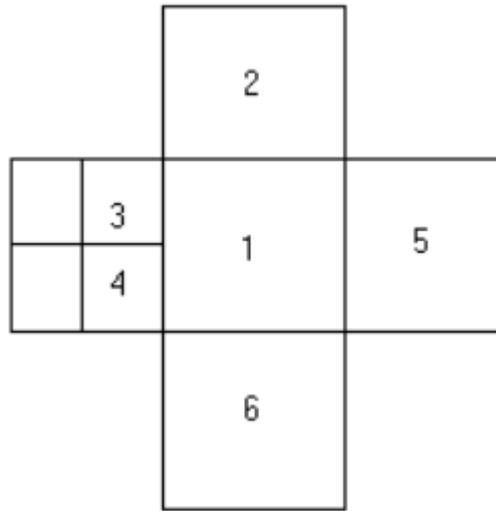


Рисунок 19. Расположение внутренних ячеек, сопрягающихся с разбиваемыми ячейками (измельчение слева – по оси φ).

Комбинация №3:

$$\begin{aligned}
& \frac{\varepsilon(T^{n+1}) - \varepsilon(T^n)}{\tau} - \left(\frac{\chi(T_5^{n+1}) + \chi(T_1^{n+1})}{2} \cdot \frac{T_5^{n+1} - T_1^{n+1}}{h^2} - \right. \\
& \left. - \frac{\chi(T_3^{n+1}) + \chi(T_1^{n+1}) + \chi(T_4^{n+1})}{3} \cdot \frac{T_1^{n+1} - 0,5(T_3^{n+1} + T_4^{n+1})}{h^2} \right) - \\
& - (r_1^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_2^{n+1})}{2} \frac{T_2^{n+1} - T_1^{n+1}}{0,5 \cdot \tilde{h}_1 \cdot (\tilde{h}_2 + \tilde{h}_1)} - \\
& - r_6^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_6^{n+1})}{2} \frac{T_1^{n+1} - T_6^{n+1}}{0,5 \cdot \tilde{h}_1 \cdot (\tilde{h}_5 + \tilde{h}_1)}) = 0.
\end{aligned} \tag{12}$$

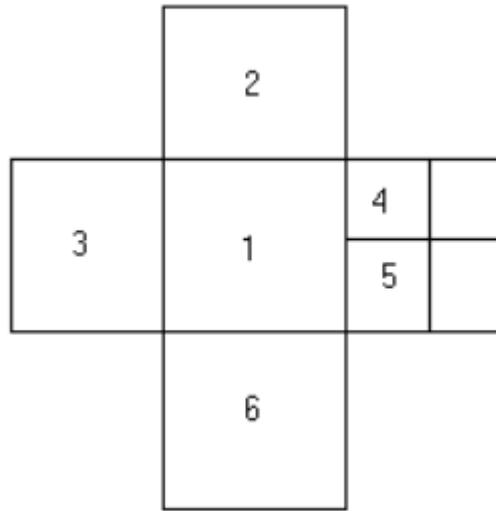


Рисунок 20. Расположение внутренних ячеек, сопрягающихся с разбиваемыми ячейками (измельчение справа – по оси φ).

Комбинация №4:

$$\begin{aligned}
 & \frac{\varepsilon(T^{n+1}) - \varepsilon(T^n)}{\tau} - \left(\frac{\chi(T_5^{n+1}) + \chi(T_1^{n+1}) + \chi(T_4^{n+1})}{3} \cdot \frac{0,5(T_5^{n+1} + T_4^{n+1}) - T_1^{n+1}}{h^2} - \right. \\
 & \quad \left. - \frac{\chi(T_3^{n+1}) + \chi(T_1^{n+1})}{2} \cdot \frac{T_1^{n+1} - T_3^{n+1}}{h^2} \right) - \\
 & \quad - (r_1^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_2^{n+1})}{2} \cdot \frac{T_2^{n+1} - T_1^{n+1}}{0,5 \cdot \tilde{h}_1 \cdot (\tilde{h}_2 + \tilde{h}_1)} - \\
 & \quad - r_6^{2\nu} \frac{\chi(T_1^{n+1}) + \chi(T_6^{n+1})}{2} \cdot \frac{T_1^{n+1} - T_6^{n+1}}{0,5 \cdot \tilde{h}_1 \cdot (\tilde{h}_5 + \tilde{h}_1)}) = 0.
 \end{aligned}$$

(13)

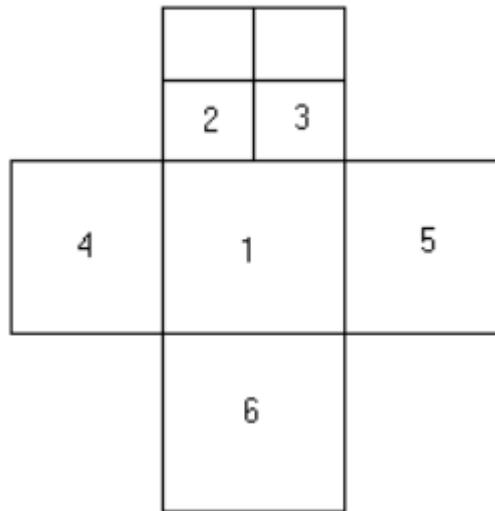


Рисунок 21. Расположение внутренних ячеек, сопрягающихся с разбиваемыми ячейками (измельчение сверху – по оси r).

3.2. Математическая модель распространения тепла на адаптивной сетке.

Важным этапом расчета температурных полей при помощи адаптивной сетки, является построение самой сетки.

3.2.1. Описание обозначений, условий и доработок схемы построения адаптивной сетки.

Перед описанием алгоритма построения адаптивной сетки, стоит обратить внимание на доработки предложенной схемы решения, а так же основные условия, которые в ней предложены:

- Важным отличием и дополнением к предложенной схеме решения, является то, что в ней рассматривалась двумерная модель. В данной же работе рассматривается трехмерная модель, которая описывает распределение тепла, в том числе и по толщине трубы, поэтому для описания изменения температуры необходимо использовать три координаты – (r, ϕ, z) .
- Условие деления узлов на более мелкие остается неизменяемым, у каждого узла не должно быть более двух соседей с одной стороны, так как это является базовым условием построения линейных уравнений. Вследствие этого, при

построении адаптивной сетки для трехмерной модели, были проведены некоторые изменения в алгоритме переопределения соседей узлов.

- Построение адаптивной сетки происходит лишь один раз и по всему стержню.
- По длине стержня (z) не задается ограничение разбиения узлов, поэтому адаптивная сетка строится по всей длине.
- По толщине стержня (r) не задается ограничение разбиения узлов, поэтому разбиение происходит только у верхнего слоя стержня, где проходит лазер.
- Задается количество узлов по углу отклонения (φ), для определения зоны, где будет происходить дробление узлов.

Разметка ячеек исследуемой области строится по принципу, показанному на рисунке 22 (на рисунке показаны декартовы координаты, а не цилиндрические, но принцип нумерации тот же, за исключением сопряжения граничных ячеек по φ , но это будет отражено в уравнениях).

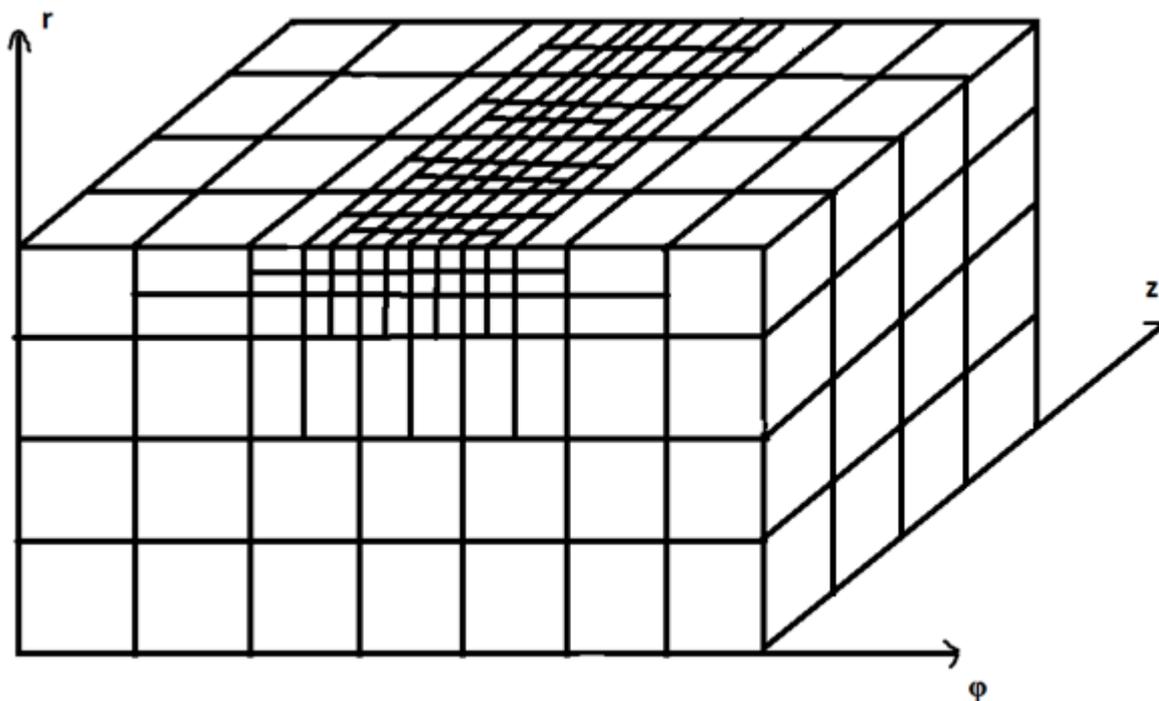


Рисунок 22. Разметка ячеек исследуемой области.

Введем следующие обозначения:

k – номер текущей ячейки;

k_t – номер ячейки, которая является для ячейки k «соседом сверху»;

k_d – номер ячейки, которая является для ячейки k «соседом снизу»;
 k_l – номер ячейки, которая является для ячейки k «соседом слева»;
 k_r – номер ячейки, которая является для ячейки k «соседом справа»;
 k_f – номер ячейки, которая является для ячейки k «соседом спереди»;
 k_b – номер ячейки, которая является для ячейки k «соседом сзади»;
 n – номер шага по времени;

Из всех возможных комбинаций построения линейных уравнений, были взяты лишь 4 случая, которые нас интересуют.

3.2.2. Описание алгоритма построения адаптивной сетки

Исходя из выше описанных условий и доработок, необходимо продумать алгоритм построения адаптивной сетки и алгоритм, по которому будет решаться система линейных уравнений с точки зрения реализации программного обеспечения.

Введем следующие обозначения:

z – длина стержня;

r – радиус стержня;

f – угол сечения стержня;

N_z – количество шагов по длине стержня;

N_r – количество шагов по радиусу стержня;

N_f – количество шагов по углу сечения стержня;

h_z – длина шага по z ;

h_r – длина шага по r ;

h_f – длина шага по f ;

T_0 – начальная температура

В первую очередь, необходимо построить обычную равномерную сетку, с заданными шагами (h_z , h_r , h_f) по всем трем измерениям. Изначально мы задаем тот шаг, который нас будет устраивать с точки зрения времени вычисления и технических ограничений (оперативная память). Построение узлов равномерной сетки происходит обычным перебором количества шагов (N_z , N_r , N_f) по каждому измерению и созданием узлов. Количество узлов сетки будет определяться как произведение количества шагов по трем осям ($N_f * N_r * N_z$).

Один узел, это отдельный элемент, который хранит в себе следующую информацию:

z_low – нижняя граница узла по оси z;

z_top – верхняя граница узла по оси z;

r_low – нижняя граница узла по оси r;

r_top – верхняя граница узла по оси r;

f_low – нижняя граница узла по оси f;

f_top – верхняя граница узла по оси f;

id – уникальный идентификатор (UUID);

T – текущая температура узла;

children – массив узлов после того, как узел был раздроблен;

left – соседи слева;

right – соседи справа;

top – соседи сверху;

back – соседи сзади;

front – соседи спереди;

bottom – соседи снизу;

При создании узлов равномерной сетки, узлам передаются данные нижних и верхних границах узла, уникальный идентификатор, начальная температура (T_0).

После создания равномерной сетки, необходимо определить у каждого узла всех соседей. Это делается с помощью верхних и нижних границ узла, по которым можно легко определить всех соседей.

Таким образом, была построена трехмерная равномерная сетка, где каждый узел знает информацию о каждом рядом стоящем узле.

Итак, можно приступать к первому разбиению узлов, в интересующей нас области. Исходя из выше описанных условий и ограничений, мы должны определить зону по оси φ . Зона будет ограничена количеством узлов (`quantity_node_f`) слева и справа от линии сварки стали, количество делаем заведомо нечетным, чтобы с каждой стороны было одинаковое количество узлов. На рисунке 23 показана блок схема, где

будет показано, как определяются нижние границы по оси φ , по которым будут взяты нужные узлы.

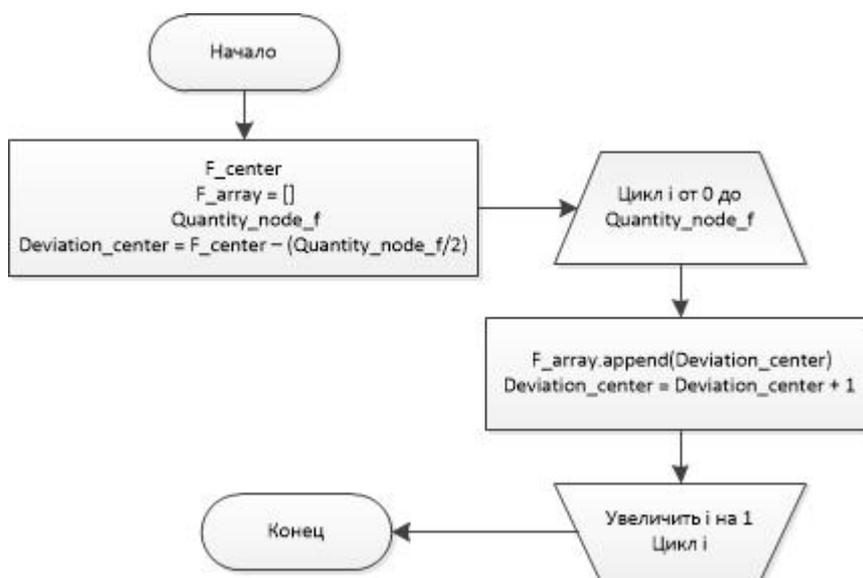


Рисунок 23. Алгоритм определения нижних границ ячеек сетки.

После того, как мы определили зону разбиения по оси φ , можно определить все узлы по трем осям, которые в дальнейшем будут раздроблены на более мелкие узлы. Как было описано выше, сетка измельчается по всей длине на верхнем слое трубы. Далее описан алгоритм, описывающий сбор узлов с необходимой зоны:

•

Итак, мы подошли к тому моменту, когда все узлы, которые будут подвержены измельчению, собраны и можно переходить к дроблению.

На рисунке 16 можно заметить, что дробление происходит по-разному. В текущей задаче измельчение происходит четырьмя разными путями:

- Измельчение узла справа на 2 узла;
- Измельчение узла слева на 2 узла;
- Измельчение узла снизу на 2 узла;
- Измельчение текущего узла на 8 узлов.

Рассмотрим все примеры того, как дробятся узлы:

- 1) Измельчение узла слева на 2 узла.

После определения зоны по оси φ , у нас есть два крайних узла, слева и справа, первое деление происходит от крайнего левого узла. Крайний левый узел будет измельчен на 8 новых узлов, и после переопределения соседей, у левого соседа, будет 4 новых соседа, что по условиям построения линейных уравнений нам не подходит. Поэтому было принято решение разделить левого соседа на 2 новых узла по оси r , где у каждого нового узла будет по 2 соседа справа.

2) Измельчение узла справа на 2 узла.

Для крайнего правого узла проделываем такую же процедуру, что и для крайнего левого. После чего у 2 новых узлов, будет 2 соседа слева.

3) Измельчение узла снизу на 2 узла.

В определенной зоне по оси φ , помимо крайних узлов, есть еще узлы, в зависимости от того, какое количество узлов было выбрано при определении зоны. Все эти узлы, так же как и левый и правый крайние будут разделены на 8 новых узлов. По условиям, мы дробим только верхний слой, поэтому соседа сверху у них нет. Но есть сосед снизу, у которого после переопределения соседей, будет 4 новых соседа, что так же не подходит под выше поставленное условие. Поэтому всех соседей снизу необходимо так же поделить на 2 новых узла по оси φ . После чего у каждого из них будет по 2 соседа сверху.

4) Измельчение текущего узла на 8 узлов.

В определенной зоне по оси φ , начиная от крайнего левого до крайнего правого, происходит дробление узлов на 8 новых узлов. Которые становятся новыми соседями для тех узлов, которые находятся рядом с родительским.

Стоит заметить, что дробление происходит не простой заменой старого узла новыми. Все новые узлы сохраняются в поле children раздробленного узла. Тем самым можно просмотреть всю иерархию измельченных узлов. Достоинство такого подхода в том, что переопределение соседей у новых узлов проходит по довольно простому алгоритму, где мы определяем соседа (с любой стороны) раздробленного

узла, выясним, был ли он раздроблен, или остался неизменным, а после определяем новых соседей. Метод определения новых соседей работает после каждого дробления. Алгоритм определения новых соседей подразумевает постоянный переход по иерархии вверх, что позволяет определить были ли раздроблены соседи с каждой стороны, определить тип дробления и далее определить новых соседей для созданных узлов.

Таким образом, мы получаем адаптивную сетку, измельчая определенную зону до необходимого нам размера шага по осям, в нашем случае нас больше всего интересует шаг по длине трубы (h_z).

Далее на основе полученной адаптивной сетки, можно построить систему разностных уравнений, описывающую модель распространения тепла, и решить ее. Решением системы уравнений будут являться новые значения температуры в узлах сетки, которые становятся входными данными для расчета температурных полей, на следующем шаге по времени.

Важным в построении матрицы для решения системы линейных уравнений, является сопоставление каждому узлу номера столбца, чтобы система выдавала правильный результат. Как уже было описано выше, каждый узел имеет уникальный идентификатор. Каждому идентификатору сопоставляется номер столбца в матрице. И при расчете коэффициентов уравнения, полученным значениям сопоставляется номер столбца матрицы, куда и будет записано значение коэффициента.

3.2.3. Получение разностных уравнений

Как было ранее отмечено в разделе 3.1, в работе [27] описана схема построения двумерной модели тепловых полей на адаптивной сетке, представлены комбинации расположения узлов и соответствующие им разностные уравнения. Основываясь на них, мы можем получить разностные уравнения для нашей доработанной модели.

Уравнение №1:

Уравнение строится для комбинации, где узел расположен внутри сетки с одним соседом с каждой стороны.

После изменения уравнения (10) и перегруппировки слагаемых получим:

$$\begin{aligned}
& \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T_{kr}^n) + \lambda(T_k^n)}{2h_\varphi^2} + \frac{\lambda(T_{kl}^n) + \lambda(T_k^n)}{2h_\varphi^2} + \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} + \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} + \right. \\
& \left. + r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kt} + \tilde{h}_k)} + r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \right) \cdot T_k^{n+1} - \frac{\lambda(T_{kr}^n) + \lambda(T_k^n)}{2h_\varphi^2} \cdot T_{kr}^{n+1} - \\
& - \frac{\lambda(T_{kl}^n) + \lambda(T_k^n)}{2h_\varphi^2} \cdot T_{kl}^{n+1} - \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kb}^{n+1} - \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kf}^{n+1} - \\
& - r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kt} + \tilde{h}_k)} \cdot T_{kt}^{n+1} - r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \cdot T_{kd}^{n+1} = \frac{\rho \cdot c}{\tau} \cdot T_k^n.
\end{aligned}$$

Уравнение №2:

Уравнение строится для комбинации, где узел расположен внутри сетки и с левой стороны узел имеет двух соседей. Для изменения уравнения (11) понадобятся дополнительные обозначения:

klt – номер ячейки, которая является для ячейки k «верхним соседом слева»;

kld – номер ячейки, которая является для ячейки k «нижним соседом слева».

После перегруппировки и приведения подобных получим:

$$\begin{aligned}
& \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T_{kr}^n) + \lambda(T_k^n)}{2h_\varphi^2} + \frac{\lambda(T_{klt}^n) + \lambda(T_k^n) + \lambda(T_{kld}^n)}{3h_\varphi^2} + \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} + \right. \\
& \left. + \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} + r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kt} + \tilde{h}_k)} + r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \right) \cdot T_k^{n+1} - \\
& - \frac{\lambda(T_{kr}^n) + \lambda(T_k^n)}{2h_\varphi^2} \cdot T_{kr}^{n+1} - \frac{\lambda(T_{klt}^n) + \lambda(T_k^n) + \lambda(T_{kld}^n)}{6h_\varphi^2} \cdot T_{klt}^{n+1} - \\
& - \frac{\lambda(T_{klt}^n) + \lambda(T_k^n) + \lambda(T_{kld}^n)}{6h_\varphi^2} \cdot T_{kld}^{n+1} - \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kb}^{n+1} - \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kf}^{n+1} - \\
& - r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kt} + \tilde{h}_k)} \cdot T_{kt}^{n+1} - r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \cdot T_{kd}^{n+1} = \frac{\rho \cdot c}{\tau} \cdot T_k^n.
\end{aligned}$$

Уравнение №3:

Уравнение строится для комбинации, где узел расположен внутри сетки и с правой стороны узел имеет двух соседей. Для изменения уравнения (12) понадобятся дополнительные обозначения:

Понадобятся дополнительные обозначения:

krt – номер ячейки, которая является для ячейки k «верхним соседом справа»;
 krd – номер ячейки, которая является для ячейки k «нижним соседом справа».

После перегруппировки и приведения подобных получим:

$$\begin{aligned} & \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T_{kl}^n) + \lambda(T_k^n)}{2h_\varphi^2} + \frac{\lambda(T_{krt}^n) + \lambda(T_k^n) + \lambda(T_{krd}^n)}{3h_\varphi^2} + \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} + \right. \\ & \left. + \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} + r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kt} + \tilde{h}_k)} + r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \right) \cdot T_k^{n+1} - \\ & - \frac{\lambda(T_{kl}^n) + \lambda(T_k^n)}{2h_\varphi^2} \cdot T_{kl}^{n+1} - \frac{\lambda(T_{krt}^n) + \lambda(T_k^n) + \lambda(T_{krd}^n)}{6h_\varphi^2} \cdot T_{krd}^{n+1} - \\ & - \frac{\lambda(T_{krt}^n) + \lambda(T_k^n) + \lambda(T_{krd}^n)}{6h_\varphi^2} \cdot T_{krt}^{n+1} - \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kb}^{n+1} - \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kf}^{n+1} - \\ & - r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kt} + \tilde{h}_k)} \cdot T_{kt}^{n+1} - r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \cdot T_{kd}^{n+1} = \frac{\rho \cdot c}{\tau} \cdot T_k^n. \end{aligned}$$

Уравнение №4: Уравнение строится для комбинации, где узел расположен внутри сетки и сверху узел имеет двух соседей. Для изменения уравнения (12) понадобятся дополнительные обозначения:

Понадобятся дополнительные обозначения:

ktl – номер ячейки, которая является для ячейки k «левым соседом сверху»;

ktr – номер ячейки, которая является для ячейки k «правым соседом сверху».

После перегруппировки и приведения подобных:

$$\begin{aligned} & \left(\frac{\rho \cdot c}{\tau} + \frac{\lambda(T_{kr}^n) + \lambda(T_k^n)}{2h_\varphi^2} + \frac{\lambda(T_{kl}^n) + \lambda(T_k^n)}{2h_\varphi^2} + \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} + \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} + \right. \\ & \left. + r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{ktl}^n) + \lambda(T_{ktr}^n)}{1,5 \cdot \tilde{h}_k \cdot (\tilde{h}_{ktl} + \tilde{h}_k)} + r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \right) \cdot T_k^{n+1} - \frac{\lambda(T_{kr}^n) + \lambda(T_k^n)}{2h_\varphi^2} \cdot T_{kr}^{n+1} - \\ & - \frac{\lambda(T_{kl}^n) + \lambda(T_k^n)}{2h_\varphi^2} \cdot T_{kl}^{n+1} - \frac{\lambda(T_{kb}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kb}^{n+1} - \frac{\lambda(T_{kf}^n) + \lambda(T_k^n)}{2h_z^2} \cdot T_{kf}^{n+1} - \\ & - r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{ktl}^n) + \lambda(T_{ktr}^n)}{3\tilde{h}_k \cdot (\tilde{h}_{ktl} + \tilde{h}_k)} \cdot T_{ktl}^{n+1} - r_k^2 \frac{\lambda(T_k^n) + \lambda(T_{ktl}^n) + \lambda(T_{ktr}^n)}{3\tilde{h}_k \cdot (\tilde{h}_{ktl} + \tilde{h}_k)} \cdot T_{ktr}^{n+1} - \\ & - r_{kd}^2 \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{\tilde{h}_k \cdot (\tilde{h}_{kd} + \tilde{h}_k)} \cdot T_{kd}^{n+1} = \frac{\rho \cdot c}{\tau} \cdot T_k^n. \end{aligned}$$

3.2.4. Учет граничных условий.

Нижний слой (по r).

k – номер ячейки нижнего слоя.

После перегруппировки:

$$\left(\frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{2h_r} + \kappa \right) \cdot T_k^{n+1} - \frac{\lambda(T_k^n) + \lambda(T_{kt}^n)}{2h_r} \cdot T_{kt}^{n+1} = \kappa \cdot T^0.$$

Верхний слой (по r).

k – номер ячейки верхнего слоя.

После перегруппировки:

$$\left(\frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{2h_r} + \kappa \right) \cdot T_k^{n+1} - \frac{\lambda(T_k^n) + \lambda(T_{kd}^n)}{2h_r} \cdot T_{kd}^{n+1} = \kappa \cdot T^0.$$

«Передний» слой (по z).

k – номер ячейки «переднего» слоя.

После перегруппировки:

$$\left(\frac{\lambda(T_k^n) + \lambda(T_{kb}^n)}{2h_z} + \kappa \right) \cdot T_k^{n+1} - \frac{\lambda(T_k^n) + \lambda(T_{kb}^n)}{2h_z} \cdot T_{kb}^{n+1} = \kappa \cdot T^0.$$

«Дальний» слой (по z).

k – номер ячейки «дальнего» слоя.

После перегруппировки:

$$\left(\frac{\lambda(T_k^n) + \lambda(T_{kf}^n)}{2h_z} + \kappa \right) \cdot T_k^{n+1} - \frac{\lambda(T_k^n) + \lambda(T_{kf}^n)}{2h_z} \cdot T_{kf}^{n+1} = \kappa \cdot T^0.$$

Условия сопряжения (по φ).

$k1$ – номер ячейки первого слоя по φ ;

$k2$ – номер смежной с ней ячейки последнего слоя по φ .

С учетом порядка нумерации на самом деле $k2 = kl$ (совпадает с номером соседа слева).

Поэтому

$$T_{k1}^{n+1} = T_{kl}^{n+1}, \quad T_{k1r}^{n+1} = T_{k2}^{n+1} = T_{kl}^{n+1},$$

где kll – номер соседа слева ячейки kl ; $k1r$ – номер соседа справа ячейки с номером $k1$.

3.2.5. Реализация моделирования тепловых полей.

Для проведения компьютерных экспериментов, была разработана программа для моделирования тепловых полей на адаптивной сетке. Программа позволяет построить равномерную сетку, и провести расчеты температурных полей по времени. Все данные по температурам, записываются в текстовый файл для дальнейшего использования полученных данных. Алгоритм расчета температурных полей представлен на рисунке 24.

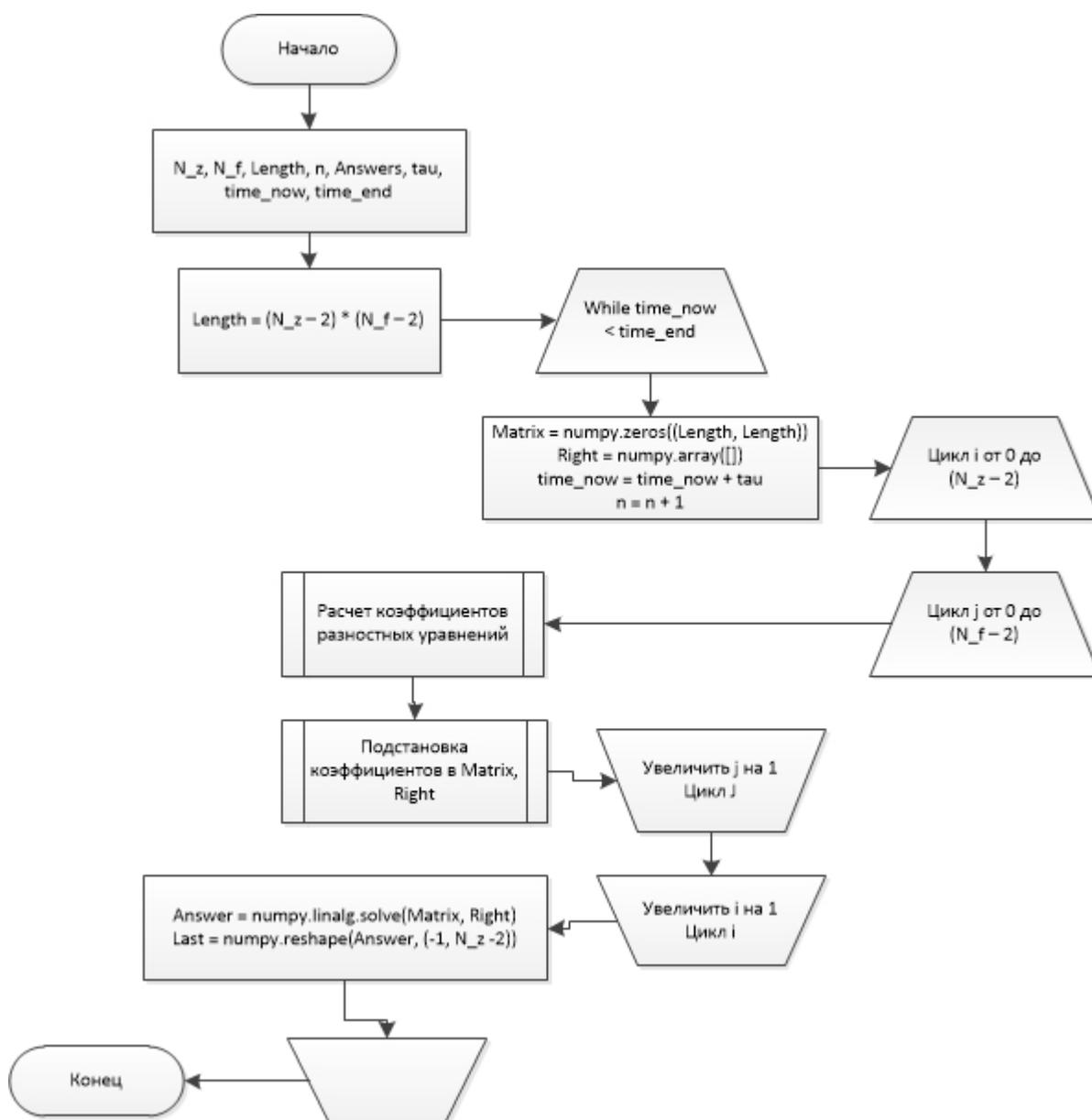


Рисунок 24. Алгоритм расчета температурных полей.

Алгоритм программы:

- Расчет внутренних параметров программы.
- Определение переменных внутри программы.
- Создание всех узлов.
- Определение зоны измельчения узлов.
- Измельчение узлов в определенной зоне.
- Переопределение соседей у узлов, находящихся в определенной зоне, а так же переопределение соседей у рядом стоящих к зоне узлов.
- Завершение дробления узлов.
- Сопоставление полученных узлов с колонками матрицы для решения системы разностных уравнений.
- Запуск цикла моделирования температурных полей
- Создание нулевой матрицы, размер которой равен количеству полученных узлов.
- Расчет коэффициентов уравнения для каждого узла и подстановка полученных коэффициентов в матрицу.
- Решение системы разностных уравнений.
- Расчет температур для начальных узлов.
- Далее происходит запись в первый файл, куда попадают все параметры, при которых работала программа, и сохранение этого файла.
- Формирование файла с ответами, полученными в ходе работы программы.

Структура программы представлена на диаграмме классов (рисунок 25):

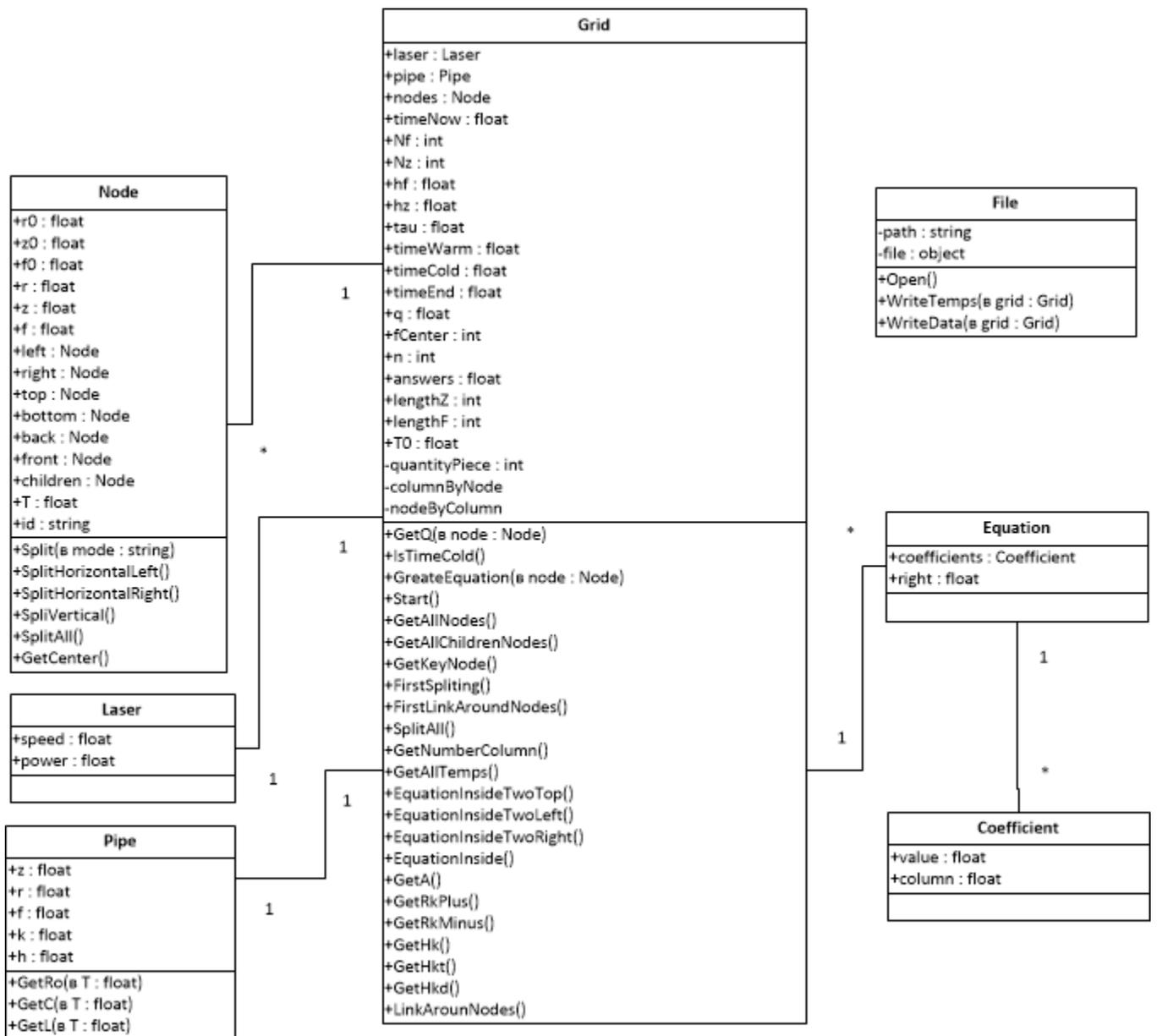


Рисунок 25. Диаграмма классов программы для расчета тепловых полей.

Таблица 9. Класс «Node»

Node	
Атрибуты	
z0	Номер координаты по оси z (начало)
r0	Номер координаты по оси r (начало)
f0	Номер координаты по оси φ (начало)
z	Номер координаты по оси z (конец)
r	Номер координаты по оси r (конец)
f	Номер координаты по оси φ (конец)
left	Объект класса Node, либо массив объектов Node определяющих соседей слева
right	Объект класса Node, либо массив объектов Node определяющих соседей справа
top	Объект класса Node, либо массив объектов Node определяющих соседей сверху
bottom	Объект класса Node, либо массив объектов Node определяющих соседей снизу
back	Объект класса Node, либо массив объектов Node определяющих соседей сзади
front	Объект класса Node, либо массив объектов Node определяющих соседей спереди
children	Объекты класса Node, определяющие новые ячейки после дробления родительской ячейки
T	Текущая температура в узле
id	Уникальный идентификатор
Методы	
Название	Описание
Split	Метод перенаправляет на другой метод в зависимости от переданного параметра
SplitHorizontalLeft	Метод для дробления ячейки по горизонтали и переопределение узлов у левого соседа
SplitHorizontalRight	Метод для дробления ячейки по горизонтали и переопределение узлов у правого соседа
SplitVertical	Метод для дробления ячейки по вертикали и переопределение узлов у нижнего соседа
SplitAll	Метод для дробления ячейки на 8 частей и определение соседей для новых узлов
GetCenter	Возвращает список координат центра ячейки

Таблица 10. Класс «Laser»

Laser	
Атрибуты	
speed	Скорость лазера
power	Мощность лазера
Методы	
Название	Описание

Таблица 11. Класс «File»

File	
Атрибуты	
path	Путь к файлу
file	Объект для работы с файлом
Методы	
Название	Описание
Open	Метод для открытия файла на редактирование
WriteTemps	Метод для записи значений температурных полей в файл
WriteData	Метод для записи входных данных эксперимента в файл

Таблица 12. Класс «Pipe»

Pipe	
Атрибуты	
z	Длина трубы
r	Радиус трубы
f	Длина окружности трубы
k	Коэффициент теплоотдачи
h	Толщина трубы
Методы	
Название	Описание
GetRo	Возвращает значение плотности
GetC	Возвращает значение теплоемкости
GetL	Возвращает значение теплопроводности

Таблица 13. Класс «Equation»

Equation	
Атрибуты	
coefficients	Массив с объектами класса Coefficient (таблица 6)
right	Значение в правой части уравнения
Методы	
Название	Описание

Таблица 14. Класс «Coefficient»

Coefficient	
Атрибуты	
value	Значение коэффициента
column	Номер колонки соответствующий данному коэффициенту
Методы	
Название	Описание

Таблица 15. Класс «Grid»

Grid	
Атрибуты	
laser	Массив с объектами класса Laser (таблица 2)
pipe	Массив с объектами класса Pipe (таблица 4)
nodes	Массив с объектами класса Node (таблица 1)
timeNow	Текущее время
timeWarm	Время продолжительности нагревания
timeCold	Время продолжительности охлаждения
timeEnd	Время завершения моделирования
Nf	Количество шагов по φ
Nz	Количество шагов по z
hf	Шаг по φ
hz	Шаг по z
tau	Шаг по времени
q	Начальная плотность потока
fCenter	Индекс центрального узла по φ
n	Индекс по времени
answers	Массив с температурными полями во времени
lengthZ	Размерность матрицы по z
lengthF	Размерность матрицы по φ
T0	Начальная температура

quantityPiece	Количество ячеек по оси φ для последующего дробления
columnsByNode	Список для сопоставления номера колонки к идентификатору ячейки
nodeByColumn	Список для сопоставления идентификатора ячейки к номеру колонки
Методы	
Название	Описание
GetQ	Возвращает значение плотности потока лазера
IsTimeCold	Возвращает Истину если наступил момент времени охлаждения
CreateEquation	Создание уравнений
Start	Метод для запуска моделирования
GetAllNodes	Метод для создания всех узлов равномерной сетки
GetAllChildrenNodes	Возвращает все полученные ячейки
GetKeyNode	Возвращает уникальный идентификатор для начальных ячеек
FirstSplitting	Измельчение начальных ячеек
FirstLinkAroundNodes	Определение соседей для начальных ячеек
GetNumberColumn	Возвращает номер колонки
GetAllTemps	Возвращает массивы тепловых полей
EquationInsideTwoTop	Создание уравнения для внутренних ячеек
EquationInsideTwoLeft	Создание уравнения для внутренних ячеек
EquationInsideTwoRight	Создание уравнения для внутренних ячеек
EquationInside	Создание уравнения для внутренних ячеек
LinkAroundNodes	Переопределение соседей

4. МОДЕЛИРОВАНИЕ ФАЗОВЫХ ПРЕВРАЩЕНИЙ

4.1. Обзор методов исследования кинетики фазовых превращений

Исследование структурных превращений в стали при термической обработке необходимо для прогнозирования свойств и структуры стали. Исследование структурных превращений экспериментальным путем приводит к построению изотермических и термокинетических диаграмм, и изучению прокаливаемости стали. Но данный подход имеет ограниченное применение описанное в работах [8, 9]. Таким образом, изотермические диаграммы используются только для качественной оценки влияния химического состава на процесс распада аустенита, термокинетические диаграммы используются только для количественной оценки устойчивости аустенита при непрерывном охлаждении. Прогнозирование структуры стали на основе ее химического состава, после термической обработки в результате изучения прокаливаемости стали невозможно. Ограничения экспериментального изучения можно уменьшить при помощи построения математических моделей, параметры которых будут определяться экспериментальным путем [3].

В работах М.Ж. Аврами и А.Н. Колмогорова [3] описаны фундаментальные (базовые) исследования в области моделирования кинетики фазовых превращений, где строятся уравнения, описывающие зависимость доли структурных составляющих в изучаемом объеме от вероятности зарождения центров фаз, скорости роста зародышей и времени превращения. Однако использование этих уравнений для моделирования превращения при непрерывном охлаждении позволяет получить только качественное соответствие данным эксперимента, что было подмечено в ряде других работ. В частности, это связано с тем, что в уравнениях Колмогорова и Аврами не учитывается инкубационный период. Инкубационный период представляется собой момент времени, который возникает при переходе системы из стабильного состояния в метастабильное. В данный момент вероятность образования зародышей новой фазы определяется не только температурой. Система постоянно находится в инкубационном периоде, когда речь заходит о непрерывном охлаждении.

Теоретический анализ кинетики фазовых превращений в твердых металлах и сплавах проводится в работе Б.Я. Любова,[1]. В работе Дж. Кристиана содержится подробное описание основных видов фазовых превращений выделения фазы из пересыщенного твердого раствора, эвтектоидного распада, полиморфных превращений и др. Также приводится классификация всех фазовых превращений в твердом состоянии и рассматривается теория процессов зарождения и роста, термодинамика, кинетика, атомный механизм и кристаллография этих превращений [2]. Описанные в данных работах аналитические модели не учитывают стохастическую природу процесса зарождения новых фаз, а также не предусматривают возможность рассмотрения зарождения, роста и разрушения отдельных структурных составляющих различных фаз, вследствие чего модель не позволяет провести анализ стереометрических параметров в полной мере.

Помимо аналитического подхода к изучению процесса фазовых превращений существует направление имитационного моделирования, основателем которого является И.Л. Миркин.

В работах [3, 4] была построена модель, где основой является теории И.Л. Миркина. Данная модель позволяет учитывать стохастическую природу процесса зарождения и инкубационный период процесса, а также позволяет получить размер зерна. Минусами данной модели являются: высокая нагрузка на компьютер, трудности реализации процесса в трехмерном пространстве, это связано с тем, что зона расплава и кристаллов представляются в виде набора элементарных фрагментов.

В работе [6] была предложена и реализована имитационная модель распада переохлажденного аустенита. Данная имитационная модель учитывает особенности структуры порошковых сталей и позволяет определить значения кинетических параметров превращения на основе экспериментально полученных изотермических диаграмм. Диаграммы строятся для некоторых значений пористости. Далее, модель использует найденные значения кинетических параметров, для этих же значений пористости, а также изотермические и термокинетические диаграммы данной стали с другими значениями пористости. Основной проблемой данной работы состоит в том,

что во время имитации не учитываются физические процессы, которые протекают во время переходов из одной фазы в другую.

4.2. Моделирование фазовых превращений при охлаждении трубы с использованием методологии теории массового обслуживания

Существует множество литературных источников, где авторы описывают кинетику фазовых превращений математически. Большинство данных моделей, которые построено на основе уравнения кристаллизации А.Н. Колмогорова или уравнения Аврами, описывают превращения в изотермических условиях. Но выше уже было сказано о том, что использование данных уравнений приведет только к получению качественного соответствия данным эксперимента. Также не стоит забывать об инкубационном периоде, во время которого вероятность образования зародышей новой фазы определяется не только температурой. При непрерывном охлаждении система постоянно находится в нестационарном периоде, причем это обстоятельство не учитывается в уравнениях Колмогорова или Аврами.

В данной работе использован другой подход к описанию кинетики фазовых превращений, который основан на имитационном моделировании процесса с использованием методологии теории массового обслуживания.

4.3. Общая схема построения имитационной модели

Модели, описанные в работах [3, 4], как и модель Колмогорова, основаны на представлении о зарождении центров новой фазы во времени, где перед зарождением новой фазы, изучаемый объем занят старой фазой. Вследствие чего, было принято решение использования методологии теории массового обслуживания. При использовании теории массового обслуживания система принимает представление, где центры возможного зарождения рассматривались, как каналы обслуживания, куда поступает поток заявок (моментов времени зарождения очередного зерна новой фазы), при образовании нового зерна, соответствующий канал становился занятым. Обслуживание заявки подразумевает рост зерна новой фазы. При этом учитывалось, что зародыш новой фазы, не достигший критического размера, может распасться с определенной вероятностью, что соответствует освобождению канала, связанного с данным зародышем.

Имитационная модель прогнозирования кинетики распада аустенита в процессе термической обработки порошковых конструкционных сталей была построена и реализована на основе применения описанного подхода. С одной стороны, успешное применение методов имитационного моделирования [3, 4] дает основание сформулировать гипотезу о применимости данной методологии к описанию кинетики фазовых превращений для материалов другого химического состава. С другой стороны, анализ процессов, происходящих при охлаждении материала интересующего нас химического состава от температуры плавления до комнатной температуры, показывает несколько иную картину превращения.

По данным [5, 6] при изучении превращения δ -феррита в материалах аналогичного химического состава следует выделять несколько температурных диапазонов. В частности, в первом (верхнем) температурном диапазоне δ -феррит, образовавшийся при кристаллизации, взаимодействует с жидким металлом, в результате чего по перитектической реакции образуется аустенит. При снижении температуры ниже температуры солидуса происходит растворение δ -феррита в аустените (превращение в твердой фазе). Авторы также отмечают, что при дальнейшем понижении температуры в результате эвтектоидной реакции может начаться выделение феррита из аустенита, однако развитие этого процесса кинетически заторможено, и потому может не приниматься во внимание.

Таким образом, имея данные об объемной доле δ -феррита в сплаве в момент окончания процесса кристаллизации, можно сконцентрироваться только на моделировании растворения феррита в аустените в твердом состоянии. При этом, в случае использования имитационного моделирования, следует внести существенные изменения в имитирующие алгоритмы, описанные в [3, 4]. Основное изменение касается уже отмеченного выше момента: вместо имитации процесса зарождения и роста зерен новой фазы в объеме, изначально занятом старой фазой, необходимо реализовать уменьшение (вплоть до исчезновения) зерен одной из двух фаз, изначально существующих в имеющемся объеме сплава.

Далее будет описана схема, по которой построена и реализована имитационная модель.

- Задается область (моделируемый объём), которая в начальный момент времени занята зернами двух фаз (δ -феррита и аустенита).
- Моделируемый объем разбивается на фрагменты (зерна). Границы зерен задаются уравнениями плоскостей, каждому из которых сопоставляется номер.
- Некоторое количество фрагментов (в соответствии с начальной долей δ -феррита) помечается как зерна δ -феррита; остальные фрагменты – как зерна аустенита.
- В процессе охлаждения моделируемого объема происходит смещение границ зерен δ -феррита в сторону уменьшения объема этих зерен, вплоть до полного их исчезновения.
- Скорость смещения границ (т. е. скорость растворения ферритных зерен) уменьшается с понижением температуры. Кроме того, предполагалось, что при сохранении постоянной температуры скорость движения границ замедляется со временем вследствие уменьшения разности концентраций элементов.

Для построения имитирующего алгоритма использовался принцип «от события к событию». Очередным событием являлось начало процесса растворения некоторого зерна δ -феррита. При этом каждое событие характеризуется моментом наступления τ_j и номером зерна δ -феррита, включенного в процесс фазового превращения. При построении модели предполагалось, что поток τ_j моментов наступления событий и поток соответствующих им номеров зерен могут быть описаны отдельно как независимые случайные объекты соответствующими законами распределения. В терминологии теории массового обслуживания зерна δ -феррита являются каналами обслуживания, на которые приходит поток заявок – моменты «включения» того или иного зерна в процесс растворения.

Последовательность моментов наступления событий рассматривалась как однородный поток. Исходя из физических закономерностей процесса фазового превращения, было принято, что этот поток обладает свойствами *стационарности* (вероятность наступления события в течение заданного промежутка времени определяется только длиной этого промежутка и не зависит от расположения данного промежутка на временном отрезке процесса превращения), *отсутствия последствия* (вероятность наступления события в течение заданного промежутка времени не зависит от того, сколько событий наступило на любом другом

промежутке, не пересекающемся с данным промежутком) и *ординарности* (вероятность наступления двух и более событий в течение малого промежутка времени пренебрежимо мала по сравнению с вероятностью наступления одного события). Тогда рассматриваемый поток является простейшим (стационарным пуассоновским) потоком, и, в соответствии с положениями теории вероятностей, время между наступлениями двух «соседних» событий имеет показательное распределение с параметром λ , где λ – интенсивность потока. Если интенсивность потока задана, то можно получить последовательность $t_1, t_2, \dots, t_k, \dots$, где t_1 – время наступления первого события, t_k , $k = 2, 3, \dots$ – время между наступлениями $(k-1)$ -го и k -го события, как последовательность псевдослучайных чисел, имеющих показательное распределение с параметром λ . Тогда моменты $\tau_1, \tau_2, \dots, \tau_k, \dots$ появления центров новой фазы могут быть выражены соотношениями

$$\tau_1 = t_1, \quad \tau_2 = t_1 + t_2, \quad \dots, \quad \tau_k = t_1 + t_2 + \dots + t_k, \quad \dots$$

Номер зерна δ -феррита, для которого инициировался процесс фазового превращения, определялся путем генерации равномерно распределенного псевдослучайного числа.

Зависимость скорости движения границ от температуры описывалась выражением $e^{-\frac{l_1}{T}}$, где l_1 – параметр, подлежащий определению. Кроме того, учитывался фактор замедления скорости со временем, прошедшим от начала превращения $e^{-l_2 \cdot t}$, и, таким образом, скорость движения границ описывалась выражением $l = e^{-\frac{l_1}{T}} \cdot e^{-l_2 \cdot t}$.

В ходе работы имитирующего алгоритма в каждый момент наступления очередного события рассчитывается объем области, занятой новой фазой. Для определения этого объема использовался метод Монте-Карло.

Кинетика процесса образования новой фазы в данной модели определяется следующими параметрами:

v_0 – объемная доля δ -феррита в начальный момент времени,

λ – интенсивность потока событий (моментов начала распада зерен δ -феррита),
 l_1 и l_2 – параметры, определяющие линейную скорость движения границ зёрен δ -феррита.

Параметр ν_0 зависит от режимов нагрева и последующего охлаждения материала, обусловленных скоростью движения лазера; параметры λ , l_1 и l_2 зависят от температуры, при которой происходит растворение δ -феррита.

Было разработано программное обеспечение, позволяющее произвести компьютерный эксперимент, в ходе которого, согласно описанной выше схеме, для данного набора значений кинетических параметров, соответствующего температуре T , моделировался реальный процесс фазового превращения, и рассчитывалась доля нерастворенного δ -феррита $V_T(\lambda, l_1, l_2, t)$ в зависимости от времени t . Процесс моделирования продолжался до полного растворения δ -феррита либо до момента времени, соответствующего окончанию реального эксперимента. Процедура определения оптимальных значений параметров модели отрабатывалась на имеющихся экспериментальных данных о кинетике растворения δ -феррита в стали 12X18H10T [6]. Для получения оптимальных значений кинетических параметров минимизировалась сумма квадратов отклонений функции $V_T(\lambda, l_1, l_2, t)$ от экспериментальных значений. В качестве метода оптимизации использовался метод Нелдера-Мида [7]. С целью уменьшения влияния случайных факторов (распределения зерен δ -феррита в исходном объеме и моментов начала растворения отдельных зерен) на результат имитационного эксперимента для каждого набора значений параметров выполнялось 10 прогонов модели, по результатам которых строилась усредненная кривая фазового превращения.

Для выполнения моделирования фазового превращения при непрерывном охлаждении необходимо описать зависимость температуры от времени, а также зависимость кинетических параметров превращения от температуры (для конкретного режима охлаждения – от времени), и при проведении имитации учитывать эти зависимости. Зависимость температуры от времени моделировалась на основе уравнения теплопроводности, как описано выше. Зависимость кинетических параметров превращения от температуры описывалась путем линейной

интерполяции значений параметров модели, полученных в результате оптимизации на этапе моделирования превращения в изотермических условиях.

Моделирование фазового превращения при непрерывном охлаждении, как и в случае изотермического превращения, выполнялось по принципу «от события к событию». На основе текущих значений кинетических параметров определялся момент времени, соответствующий наступлению следующего события. На этот момент времени определялось значение температуры, и рассчитывались значения параметров модели, соответствующих этой температуре; определялся объем нерастворенного δ -феррита. Новые значения параметров становились текущими.

Процесс моделирования продолжался до достижения комнатной температуры.

4.4. Алгоритм, имитирующий растворение δ -феррита в изотермических условиях

Уравнения плоскостей, определяющие границы раздела фаз (аустенита и δ -феррита), в декартовой системе координат имеют вид $A_i \cdot x + B_i \cdot y + C_i \cdot z + D_i = 0$, $i = 1, 2, \dots, K$, где K – общее количество таких плоскостей. Проверка выполнения неравенства

$$A_i \cdot x_j + B_i \cdot y_j + C_i \cdot z_j + D_i > 0 \quad (A_i \cdot x_j + B_i \cdot y_j + C_i \cdot z_j + D_i < 0) \quad (23)$$

позволяет определить положение точки с координатами (x_j, y_j, z_j) относительно i -й границы. Таким образом, принадлежность точки с заданными координатами к одному из зерен аустенита или δ -феррита математически описывается системой линейных неравенств вида (23).

В ходе первоначальной разметки зерен, определяющей принадлежность каждого зерна к одной из двух фаз, определялась система неравенств, описывающая каждое зерно δ -феррита (коэффициенты уравнения каждой из границ зерна и соответствующие им знаки неравенств).

Процесс растворения ферритных зерен соответствует перемещению границ раздела фаз в направлении области, занятой δ -ферритом. С математической точки зрения это означает изменение коэффициентов уравнения, определяющего каждую границу. Перемещение плоскости $A_i \cdot x + B_i \cdot y + C_i \cdot z + D_i = 0$ на единицу длины означает изменение коэффициентов ее уравнения в соответствии с правилами

$$A'_i = A_i \pm \frac{A_i}{|\vec{n}_i|}, \quad B'_i = B_i \pm \frac{B_i}{|\vec{n}_i|}, \quad C'_i = C_i \pm \frac{C_i}{|\vec{n}_i|}, \quad D'_i = D_i \pm \frac{D_i}{|\vec{n}_i|}, \quad (24)$$

где $|\vec{n}_i| = \sqrt{A_i^2 + B_i^2 + C_i^2}$, а выбор знака определяется расположением фаз (аустенита и δ -феррита) относительно данной границы.

С учетом того, что скорость движения границ при температуре T определяется, как было указано выше, соотношением $l = e^{-\frac{l_1}{T}} \cdot e^{-l_2 \cdot t}$, за промежуток времени Δt каждая плоскость, ограничивающая растворяющееся зерно, должна переместиться на

$$\Delta l = \int_t^{t+\Delta t} e^{-\frac{l_1}{T}} \cdot e^{-l_2 \cdot t} dt = \frac{e^{-\frac{l_1}{T}}}{l_2} \cdot (e^{-l_2 t} - e^{-l_2(t+\Delta t)}). \quad (25)$$

Объединяя (24) и (25), получим правила для изменения коэффициентов уравнений границ растворяющегося зерна за промежуток времени Δt :

$$A'_i = A_i \pm \frac{A_i}{|\vec{n}_i|} \cdot \Delta l, \quad B'_i = B_i \pm \frac{B_i}{|\vec{n}_i|} \cdot \Delta l, \quad C'_i = C_i \pm \frac{C_i}{|\vec{n}_i|} \cdot \Delta l, \quad D'_i = D_i \pm \frac{D_i}{|\vec{n}_i|} \cdot \Delta l, \quad (26)$$

В процессе работы алгоритма в каждый момент наступления события в системе рассчитывается доля объема, занятого δ -ферритом. Сначала переопределяются коэффициенты в уравнениях границ всех зерен, уже включенных в процесс фазового превращения, в соответствии с (26). Для расчета доли объема δ -феррита используется метод Монте-Карло. В соответствии с этим методом в область исследуемого пространства выбрасываются случайные точки с координатами (x, y, z) , где случайные величины X, Y и Z имеют равномерное распределение на интервале, определяемом размерами области. Для каждой такой точки выполняется проверка на принадлежность одному из зерен δ -феррита (последовательная проверка выполнения систем линейных неравенств, определяющих каждое зерно с текущими значениями коэффициентов). Если обозначить через N общее число выброшенных точек, а через N_p – число точек, попавших в какое-либо зерно δ -феррита, то доля области, занятой новой фазой, составит

$$V_f = \frac{N_p}{N}. \quad (27)$$

4.5. Алгоритм определения значений кинетических параметров превращения в изотермических условиях

Кинетические параметры процесса (параметры скорости растворения l_1 и l_2 и интенсивность включения зерен δ -феррита в процесс растворения λ) подбирались так, чтобы кривые распада, полученные методом компьютерного моделирования, наилучшим образом описывали данные экспериментов, представленные в литературных источниках [5, 6]. С этой целью для каждого значения температуры T изотермической выдержки, для которого имелись данные о кинетике превращения, минимизировалась функция

$$Vm_T(\lambda, l_1, l_2) = \sum_{i=1}^{n_T} (y(\lambda, l_1, l_2, t_i) - y_i)^2, \quad (28)$$

где

(t_i, y_i) – точки кинетической кривой по данным [6],

$y(\lambda, l_1, l_2, t_i)$ – доля δ -феррита, оставшаяся нерастворенной к моменту времени t_i , по данным компьютерного моделирования,

n_T – число экспериментальных точек для температуры T .

Расчеты выполнялись следующим образом. Для текущих значений λ , и l_1 и l_2 производилось компьютерное моделирование, в ходе которого определялись значения функции $y(\lambda, l_1, l_2, t_i)$ для заданных моментов времени t_i . Для определения оптимальных значений кинетических параметров в соответствии с методом наименьших квадратов ставилась задача

$$Vm_T(\lambda, l_1, l_2) \rightarrow \min \quad (29)$$

Задача (29) имеет следующие особенности. Во-первых, минимизируемая функция не задана аналитически. Поэтому классические методы оптимизации, использующие градиент функции и производные второго порядка (например, метод Ньютона и его модификации), неприменимы к данной задаче. Поэтому для решения задачи (29) следовало выбрать один из методов прямого поиска, используемых для оптимизации негладких функций [7].

Во-вторых, учитывая стохастическую природу рассматриваемого процесса, следует иметь в виду, что значение функции $Vm_T(\lambda, l_1, l_2)$ при заданных λ, l_1, l_2 и T представляет собой случайную величину (ее значение зависит как от начального распределения зерен двух фаз в исследуемом объеме, так и от случайных факторов, определяющих динамику изменения объема фаз: случайности моментов времени, «запускающих» процесс для отдельных зерен, случайного выбора зерна, для которого «запускается» процесс растворения). Это обстоятельство затрудняет применение методов оптимизации (включая и методы прямого поиска), так как изменение значения функции, которое ожидается в связи с переопределением одного из параметров на очередном шаге процесса оптимизации, может не превосходить случайного разброса значений оптимизируемой функции.

Для корректного применения оптимизационных алгоритмов в случае статистического моделирования необходимо при проверке выполнения неравенств вида $Vm_T(\lambda^{(1)}, l_1^{(1)}, l_2^{(1)}) > Vm_T(\lambda^{(2)}, l_1^{(2)}, l_2^{(2)})$ применять статистические критерии. В случае задачи (29) предлагается использовать следующий подход.

Для каждого значения температуры изотермической выдержки T при одних и тех же значениях параметров λ , и l_1 и l_2 , соответствующих данной температуре, производилась серия компьютерных экспериментов, моделирующих растворение δ -феррита (не менее 50), в результате которых определялась серия значений $Vm_T^{(i)}(\lambda, l_1, l_2)$, $i = 1, 2, \dots, m$, где m – число экспериментов в серии. После этого по данным полученной выборки оценивалось среднее значение $\overline{Vm_T}(\lambda, l_1, l_2)$, и среднее квадратическое отклонение значений серий, и проверялась гипотеза о нормальном распределении случайной величины $Vm_T(\lambda, l_1, l_2)$. Проверка гипотезы выполнялась с помощью критерия Пирсона [8, 9] при уровне значимости $\alpha = 0,05$. После проведения необходимых расчетов для большинства выборок следовало сделать вывод о том, что гипотеза о нормальном распределении не противоречит опытным данным.

По данным различных источников [8, 9] известно, что t -критерий (критерий Стьюдента), используемый при сравнении средних, относительно нечувствителен к отклонениям исследуемой случайной величины от нормального закона

распределения, особенно в случае больших выборок. Поэтому было принято решение использовать этот критерий в процессе минимизации функции $Vm_T(\lambda, l_1, l_2)$.

Использование t -критерия в процессе поиска оптимальных значений кинетических параметров заключалось в следующем. Если при очередном изменении значений параметров оказывалось, что $\overline{Vm_T}(\lambda^{(1)}, l_1^{(1)}, l_2^{(1)}) > \overline{Vm_T}(\lambda^{(2)}, l_1^{(2)}, l_2^{(2)})$, т. е. новые значения параметров приводят к уменьшению среднего значения минимизируемой величины, то с помощью t -критерия проверялась статистическая гипотеза

$$H_0: M(Vm_T(\lambda^{(1)}, l_1^{(1)}, l_2^{(1)})) = M(Vm_T(\lambda^{(2)}, l_1^{(2)}, l_2^{(2)}))$$

при альтернативной гипотезе

$$H_1: M(Vm_T(\lambda^{(1)}, l_1^{(1)}, l_2^{(1)})) > M(Vm_T(\lambda^{(2)}, l_1^{(2)}, l_2^{(2)})).$$

В случае принятия альтернативной гипотезы значения кинетических параметров становились равными $(\lambda^{(2)}, l_1^{(2)}, l_2^{(2)})$.

В качестве метода оптимизации использовался метод Нелдера-Мида [7]. Суть этого метода заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума функции. Под симплексом в данном случае понимается выпуклый многогранник с числом вершин, равным $(N+1)$, где N – количество параметров, используемых в процессе оптимизации.

Далее представлен оптимизационный алгоритм, построенный на основе метода Нелдера-Мида, и дополненный проверкой статистических гипотез.

1. Выбор начальной точки $x^0 = (x_1^0, x_2^0, x_3^0) = (\lambda, l_1, l_2)$.

Изменение каждой из координат точки x^0 на значение шага h и определение точек $x^i = (x_1^i, x_2^i, x_3^i)$, $i = 1, 2, 3$, образующих симплекс.

Вычисление значений минимизируемой функции f в точках x^i .

2. Выбор из вершин симплекса трех точек: x^{max} с наибольшим значением оптимизируемой функции, x^{min} – с наименьшим значением функции и x^{mid} – с промежуточным значением функции. При сравнении значений функции и вытекающем из него выборе точек x^{max} , x^{min} и x^{mid} применялся t -критерий: для средних значений функции f , полученных в результате серии прогонов

имитирующего алгоритма для сравниваемых точек x^1 и x^2 , вычислялась статистика

$$t = \frac{\bar{f}(x^1) - \bar{f}(x^2)}{\sqrt{(n-1) \cdot (s_{x^1}^2 + s_{x^2}^2)}} \cdot \sqrt{n \cdot (n-1)},$$

где s – исправленное среднее квадратическое отклонение, а n – число прогонов; после этого значение t сравнивалось с критическим значением $t_{кр}$, определяемым как критическая точка одностороннего распределения Стьюдента с числом степеней свободы, равным $2n - 2$ и заданным уровнем значимости. В случае принятия альтернативной гипотезы принималось соответствующее решение относительно выбора точек x^{max} , x^{min} и x^{mid} .

Определение значений функции $f^{max} = (x^{max})$, $f^{min} = (x^{min})$ и $f^{mid} = (x^{mid})$. Дальнейшие шаги направлены на уменьшение значения f^{max} .

3. Определение центра тяжести всех точек, кроме x^{max} : $x^c = \frac{1}{3} \cdot \sum_i x^i$.
4. Отражение точки x^{max} относительно x^c с коэффициентом $\alpha > 0$ (при $\alpha = 1$ применяется центральная симметрия), в результате чего получается точка $x^r = (1 + \alpha) \cdot x^c - \alpha \cdot x^{max}$ и вычисляется значение функции $f^r = (x^r)$.
5. Сравнение значения f^r со значениями f^{max} , f^{min} и f^{mid} :
 - 5.1. В случае $f^r < f^{min}$ производится растяжение с коэффициентом $\gamma > 0$: определяется новая точка $x^b = (1 + \gamma) \cdot x^c + \gamma \cdot x^r$ и значение функции $f^b = (x^b)$.
Если $f^b < f^{min}$, то $x^{max} = x^b$; переход на шаг 9.
 - 5.2. Если $f^{min} < f^r < f^{mid}$, $x^{max} = x^r$; переход на шаг 9.
 - 5.3. Если $f^{min} > f^r > f^{mid}$, то точки x^{max} и x^r меняются местами (как и соответствующие им значения функции); переход на шаг 6.
 - 5.4. Если $f^r > f^{max}$, то переход на шаг 6.
6. Построение точки $x^s = \beta \cdot x^{max} + (1 - \beta) \cdot x^c$ ($\beta > 0$ – коэффициент сжатия) и определение значения $f^s = (x^s)$.
7. Если $f^s < f^{max}$, то $x^{max} = x^s$; переход на шаг 9.
8. Если $f^s > f^{max}$, то сжатие симплекса – гомотетия к точке x^0 :

$$x^i \rightarrow x^0 + \frac{1}{2}(x^i - x^0) \text{ для всех точек } x^i.$$

9. Проверка сходимости: проверка взаимной близости полученных вершин симплекса, следствием чего предполагается их близость к искомому минимуму. Если требуемая точность не достигнута, то переход на шаг 2.

4.6. Алгоритм, имитирующий растворение δ -феррита при непрерывном охлаждении

Значения кинетических параметров фазового превращения, найденные в результате решения задачи (29), использовались для моделирования растворения δ -феррита при непрерывном охлаждении (в процессе остывания трубы после завершения сварки).

Режимы нагрева и охлаждения тех или иных фрагментов определялись в соответствии с моделями распределения тепла, рассмотренными в разделе 1. Для каждой интересующей точки (по заданным координатам) строилась зависимость температуры от времени (с заданным в модели шагом), которая и определяла моделируемый режим охлаждения.

Моделирование фазового превращения производилось от момента достижения температуры солидуса. Предполагалось, что в течение промежутка времени, равного шагу по времени, использованному при моделировании распространения тепла, температура рассматриваемой точки сохраняет постоянное значение (определенное в ходе моделирования температурных полей). В течение этого промежутка времени имитировался процесс растворения δ -феррита так, как это делалось для случая изотермической выдержки. При этом использовались значения кинетических параметров, найденные в ходе решения задачи (29). Для значений температуры, по которым отсутствовали литературные данные, выполнялась линейная интерполяция значений параметров, полученных по известным температурам.

Описанный процесс продолжался до достижения рассматриваемым фрагментом комнатной температуры.

4.7. Реализация моделирования фазовых превращений.

Для проведения компьютерных экспериментов, была разработана программа для моделирования растворения δ -феррита. Данная программа изначально применяется для определения кинетических параметров с помощью метода оптимизации Нелдера-Мида. После расчета кинетических параметров, они подставляются в программу и программа производит моделирование распада δ -феррита и рассчитывает его остаток.

Функционал для определения кинетических параметров растворения феррита:

Происходит вызов метода `Scipy.optimize.minimize()`, куда передаются: метод, реализующий имитацию кристаллизации, начальные параметры I_1 , I_2 , λ и алгоритм минимизации.

Далее будет описан алгоритм функции, результат который необходимо оптимизировать.

- Ввод параметров с клавиатуры.
- Создание внутренних параметров программы.
- Создание зерен.

Изначально имеется некоторая область (моделируемый объем).

Моделируемый объем разбивается на фрагменты, программа формирует $N_x * N_y * N_z$ число зерен. С помощью неравенств задаем границы каждого зерна, и определяем номер каждого зерна.

- Определение типа зерна в зависимости от начального процента содержания феррита.

Запускаем цикл и случайным образом выбираем полученные зерна, которым присваиваем тип ферритное зерно, и подсчитываем общий объем зерен с таким типом. Цикл прекращает работу в момент, когда общий объем ферритных зерен будет больше или равен изначально заданному объему феррита.

- Имитация растворения ферритных зерен.

Запускается цикл с условием, где условием является достижение текущего времени с временем растворения феррита. Момент времени определяются случайным способом, в зависимости от параметра интенсивности потока событий. В текущий момент времени, случайным образом выбирается зерно, которое с данного момента времени начнет растворяться. Выбранное зерно перемещается в массив ферритных зерен, у которых начался сдвиг границ. Далее у всех зерен находящиеся в этом массиве начинается движение границ, причем движение границ происходит в случае, если соседнее зерно имеет другой тип. Скорость движения границ определяется с помощью параметров I_1 , I_2 , текущего времени и времени, с которого зерно началось движение границ.

- Удаление зерен, которые были полностью растворены.
- Расчет общего объема феррита.
- Расчет целевой функции.

Целевой функцией метода является, сумма квадрата разностей между полученными долями феррита в определенные моменты времени и долями феррита, полученными из других работ.

Описание алгоритма прогнозирования количества феррита в сварочном шве:

- Создание внутренних параметров программы.
- Загрузка температурных полей с файла.
- Формирование одномерного массива изменения температуры в одной точке при непрерывном охлаждении (при помощи входных параметров z и f).
- Запуск цикла по всех температурам и запуск имитации растворения феррита. Для текущей температуры находятся параметры I_1 , I_2 , λ с помощью интерполяции по известным параметрам для других температур.

Запуск метода имитации с вычисленными параметрами.

- Подсчет общей доли феррита на текущее время.
- Запись результатов в файл.

4.8. Полученные результаты моделирования фазовых превращений.

После определения скоростей охлаждения зоны сварочного шва с помощью имитирующего алгоритма рассчитывалось количество остаточного δ -феррита. Результаты, полученные для разных режимов сварки, представлены в таблице 16.

Таблица 16. Количество остаточного δ -феррита в зоне сварочного шва при различных режимах сварки.

Режим сварки (мощность, скорость движения лазера)	Доля остаточного δ - феррита, %
$Q = 1$ кВт; $v = 0,5$ м/мин	5
$Q = 2$ кВт; $v = 1$ м/мин	1,8
$Q = 3$ кВт; $v = 2$ м/мин	1,4
$Q = 5$ кВт; $v = 3$ м/мин	0,55

ЗАКЛЮЧЕНИЕ

Для достижения поставленных целей и задач была разработана интегрированная модель фазовых превращений, объединяющая модель изменения температурных полей заготовки в процессе сварки и последующего охлаждения, а также имитационную модель фазовых превращений, происходящих при охлаждении стали. Но стоит сказать о тех проблемах и решениях, с которыми пришлось столкнуться в процессе разработки моделей и проведении компьютерных экспериментов.

При разработке модели тепловых полей на начальном этапе было принято решение построить равномерную сетку на всем стержне, а так же пренебречь распространением тепла внутри стержня. После чего, по полученным результатам моделирования, были построены графики:

- Зависимость температуры от времени на линии сварного шва;
- Зависимость температуры от времени на расстоянии от линии сварного шва;
- Изменение температуры по сечению трубы.

При изучении построенных графиков можно отследить, что до момента наступления температуры солидуса и после, температура изменяется очень резко. Это было связано с тем, что на всем стержне построена равномерная сетка, и мы не можем делать шаг между узлами очень маленький так, как сталкиваемся с техническими ограничениями, и долгими расчетами тепловых полей в каждом узле сетки. Поэтому и встал вопрос о погрешности данных вычислений. Тем более, результаты по температурным полям являются входными данными для моделирования фазовых превращений, что и является основной подцелью данной работы, после чего погрешность в моделировании тепловых полей приведет к еще большей погрешности в моделировании фазовых превращений. Стоит так же учесть, что данную проблему невозможно решить в рамках модели тепловых полей на равномерной сетке.

После изучения существующих решений данной проблемы, была найдена схема решения в статье [27], где была описана схема решения уравнения

теплопроводности на адаптивной сетке. В данной статье был описан подход, для решения двумерной задачи, который был взят за основу, для создания новой модели расчета температурных полей. В новой, созданной модели расчета температурных полей, был включен ряд изменений и допущений, которые делают модель не универсальной, и используемой только в данной работе.

Было разработано программное обеспечение для моделирования тепловых полей на адаптивной сетке. В данный момент программа находится на этапе отладки, поэтому невозможно представить результаты расчетов температурных полей на адаптивной сетке.

СПИСОК ЛИТЕРАТУРЫ

1. Любов Б.Я. Кинетическая теория фазовых превращений. - М.: Металлургия, 1969.
2. Кристиан Дж. Теория превращений в металлах и сплавах. Часть 1. -М.: Мир, 1978.
3. Карякин, Иван Юрьевич. Математическое моделирование и прогнозирование структуры стали после термической обработки: дис. ; науч. рук. А.Г. Ивашко.; Тюм. гос. ун-т. - Тюмень, 2011.
4. Миркин И.Л. Фазовые превращения. // Структура и свойства сталей. - М, Оборонгиз, 1941.
5. Колмогоров А.Н. К статистической теории кристаллизации металлов // Известия АН СССР. Серия математическая, 1937.
6. Ивашко А.Г., Цыганова М.С. Набатов Р.И. Исследование кинетики распада переохлажденного аустенита порошковых сталей различной пористости методом имитационного моделирования, Журнал «Известия ВУЗов. Черная металлургия», 2016, С. 191-196
7. Фазовые превращения в стали, URL: http://metallicheckiy-portal.ru/articles/chermet/fazovie_sostoyania
8. Atlas of Time-Temperature Diagrams for Irons and Steels. – ASM International, 1991. – 804 p.
9. Ю.Малинкина Е.И., Ломакин В.Н. Прокаливаемость стали. — М.: Машиностроение, 1969.
- 10.Блантер М.Е. Теория термической обработки. М.: Металлургия, 1984.
- 11.Н.Г. Карлыханов, А.В. Уракова. Неявная схема решения нелинейного уравнения теплопроводности на квадратной адаптивной сетке
- 12.Python, URL: <https://www.python.org/doc/>
- 13.Бьёрн Страуструп. Язык программирования C++. Специальное издание = The C++ programming language. Special edition. — М.: Бином-Пресс, 2007.

14. Бьёрн Страуструп. Программирование: принципы и практика использования C++, исправленное издание = Programming: Principles and Practice Using C++. — М.: Вильямс, 2011.
15. Бьёрн Страуструп. Дизайн и эволюция C++ = The Design and Evolution of C++. — СПб.: Питер, 2007
16. Fortran. Programmer's Reference Manual. The Fortran Automatic Coding System for the IBM 704 EDPM. — IBM Corp., 1956.
17. Роберт У. Себеста. 2.3. Компьютер IBM 704 и язык Фортран // Основные концепции языков программирования = Concepts of Programming Languages / Пер. с англ. — 5-е изд. — М.: Вильямс, 2001.
18. FORTRAN - Энциклопедия языков программирования, URL: <http://progopedia.ru/language/fortran/>
19. Преимущества и недостатки Fortran'a, URL: <http://forum.sources.ru/index.php?showtopic=97419>
20. О бедном Фортране замолвите слово, URL: <https://habrahabr.ru/post/202680/>
21. DISLIN — высокоуровневая библиотека визуализации пользовательских данных, URL: <https://habrahabr.ru/post/125611/>
22. PyOpenGL для начинающих, URL: <https://habrahabr.ru/post/246625/>
23. Краткий курс компьютерной графики, URL: <https://habrahabr.ru/post/248153/>
24. Почему Python? , URL: <https://khashtamov.com/ru/why-python/>
25. 5 причин, почему Python всё еще крут, URL: https://geekbrains.ru/posts/why_love_python
26. OpenGL, URL: <https://www.opengl.org/>
27. Matplotlib mplot3d toolkit, URL: https://matplotlib.org/mpl_toolkits/mplot3d/index.html
28. Н.Г. Карлыханов, А.В. Уракова. Неявная схема решения нелинейного уравнения теплопроводности на квадратной адаптивной сетке. URL: <http://docplayer.ru/60496757-Neyavnaya-shema-resheniya-nelineynogo-uravneniya-teploprovodnosti-na-kvadratnoy-adaptivnoy-setke.html>

29. В. В. Савин. Организация в Калининградской области нового экологически сбалансированного наукоемкого инновационного промышленного производства сварных труб специального назначения из нержавеющей марок сталей, титановых и жаропрочных сплавов с применением высокоскоростной лазерной сварки и последующей высокопроизводительной термической и химико-термической обработки.

URL: http://www.fcpir.ru/participation_in_program/contracts/14.578.21.0252/

30. A G Ivashko, M S Tsyganova, M M Mustafin, V V Savin, L A Savina, V A Chaika, M N Sorokovikov and A M Kalikulov. Simulation of structure formation of austenitic steel weld.