

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа теоретической механики и математической физики

Работа допущена к защите
Директор ВШТМиМФ,
Д.ф.-м.н., чл.-корр. РАН
_____ А.М. Кривцов
« ___ » _____ 2024г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
МАГИСТЕРСКАЯ ДЕССЕРТАЦИЯ
РАЗРАБОТКА ТЕХНОЛОГИИ СОЗДАНИЯ МОДЕЛЕЙ РЕБЕР НА
ОСНОВЕ КОМПЬЮТЕРНОЙ ТОМОГРАФИИ С ПОМОЩЬЮ 3D
ПЕЧАТИ**

По направлению подготовки (специальности)
01.04.03 «Механика и математическое моделирование»
Направленность (профиль)
01.04.03_03 «Механика и цифровое производство»

Выполнил
Студент гр. 5040103/20301
Руководитель
доцент ВШТМиМФ, к.ф.-м.н.

Кульчицкий И. А.
Лобода О. С.

Консультант
По анализу КТ

Суворов В. В.

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**
Физико-механический институт
Высшая школа теоретической механики и математической физики

УТВЕРЖДАЮ

Директор ВШТМиМФ

А. М. Кривцов

«__» _____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Кульчицкому Игорю Андреевичу, гр. 5040103/20301

1. Тема работы: Разработка технологии создания моделей ребер на основе компьютерной томографии с помощью 3D печати
2. Срок сдачи студентом законченной работы: 30.05.2024
3. Исходные данные по работе: снимки компьютерной томографии;
4. Содержание работы (перечень подлежащих разработке вопросов):
 - Обработка КТ снимков;
 - Применение новых методов и алгоритмов для обработки снимков;
 - Преобразование КТ снимков в 3D геометрию;
 - Проведение постобработки 3D моделей;
 - Подбор пористой структуры;
 - Определение материала;
 - Определение типа закрепления.
5. Перечень графического материала (с указанием обязательных чертежей): не предусмотрено
6. Консультанты по работе: Суворов В.В.
7. Дата выдачи задания 26.02.2024

Руководитель ВКР _____ Лобода О.С. – доцент ВШТМиМФ, к.ф.-м.н., доцент

Задание принял к исполнению 26.02.2024

Студент _____ Кульчицкий И.А.

РЕФЕРАТ

На 51 с., 46 рисунков, 5 таблиц

КОМПЬЮТЕРНАЯ ТОМОГРАФИЯ, КТ, МЕДИЦИНСКИЕ ИЗОБРАЖЕНИЯ, МОДЕЛИРОВАНИЕ, ОБРАБОТКА, 3D-ОБЪЕКТ, 3D ПЕЧАТЬ, СЕТКА, FDM SLA, РЕБРА, ИМПЛАНТ

Данная работа посвящена разработке программы для обработки кт снимков, применение этой программы для получения 3д моделей ребер человека, подбор пористой структуры и разработке системы крепления.

В результате была разработана программа выполняющий свой функционал, исследованы пористые структуры и разработана система крепления импланта к ребру.

THE ABSTRACT

On 51 pages, 46 pictures, 5 tables.

COMPUTED TOMOGRAPHY, CT, MEDICAL IMAGING, MODELING, PROCESSING, 3D-OBJECT, 3D PRINTING, MESH, FDM, SLA, RIBS, IMPLANT

This work is devoted to the development of a program for processing CT images, application of this program to obtain 3D models of human ribs, selection of the porous structure and development of the anchoring system.

As a result, a program was developed that performs its functionality, porous structures were investigated and the system of implant attachment to the rib was developed

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ	8
ГЛАВА 2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	9
2.1. Базовая разработки	9
2.1.1. Иконки.....	10
2.1.2. Общая навигация.....	11
2.1.3. Чтение файлов КТ	13
2.1.4. Сохранение и открытие файлов.....	15
2.1.5. Языковые пакеты	16
2.2. Расширенная разработка	17
2.2.1. Конвертация в Mesh.....	17
2.2.2. Декомпозиция.....	18
2.3. Сравнение программ.....	20
2.3.1. Преимущества программы.....	22
ГЛАВА 3. АДДИТИВНЫЕ ТЕХНОЛОГИИ.....	24
3.1. Экструзионная печать (FDM, FFF, FGF)	24
3.2. Стереолитография (SLA, DLP)	25
3.3. Другие технологии печати	26
ГЛАВА 4. ВОЗМОЖНЫЕ РЕАЛИЗАЦИИ В ПЕЧАТИ	27
4.1. Сетки в реализации FDM	27
4.2. Сетки в реализации SLA	29
ГЛАВА 5. ИССЛЕДОВАНИЕ СТРУКТУР	30
5.1. Пропускная способность.....	31
5.1.1. Шаровая ячейка	32
5.1.2. Кубическая ячейка	34
5.2. Нагрузки.....	35
5.2.1. Кубическая ячейка	35

	5
5.2.2. Шаровая ячейка	37
5.3. Получение модели в реальности	39
5.4. Выводы по сеткам	42
ГЛАВА 6. ПОИСК СОЕДИНЕНИЙ.....	43
6.1. Типы соединений	43
6.1.1 Клеммовое соединение.....	44
6.1.2. Защелки.....	44
6.1.3. Клеевое соединение.....	45
6.1.4. Соединение с натягом.....	46
7.2. Выводы по соединениям	46
ГЛАВА 7. РАЗРАБОТКА КРЕПЕЖА.....	47
7.1. Этап 1. Получение меш модели ребра	47
7.2. Этап 2. Модель крепежа.....	49
7.3. Исследование.....	50
7.3.1. Болтовое соединение	50
7.3.2. Крепление с посадкой.....	52
7.4. Сравнение и анализ.....	53
7.5. Анализ результатов.....	55
ЗАКЛЮЧЕНИЕ	56
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	57

ВВЕДЕНИЕ

3D-печать – это широко распространенная технология, которая недавно нашла свое применение в медицинской сфере. Она становится все более популярным инструментом, способствующим быстрому технологическому прогрессу.

Наряду с ростом доступности 3D-печати, этот инструмент начал применяться и в медицинской практике для совершенствования методов хирургического вмешательства при различных патологиях[13][10]. Современные технологии позволяют специалистам получать виртуальные 3D-реконструкции любой части тела для получения точного изображения патологии. На данный момент большинство исследований, посвященных использованию 3D-печати в медицине относится к реконструктивной, онкологической и челюстно-лицевой хирургии [5][6][7][10][12][15].

За последние годы технология 3D-печати значительно усовершенствовалась и стала более доступной. При реконструкции костных структур наиболее важным является акцент на их внешних границах.

Помимо 3D-печати в область медицины постепенно внедряются методы машинного обучения и приобретают все большее значение и актуальность[17]. Программные решения, основанные на алгоритмах машинного обучения, демонстрируют потенциал в оптимизации процессов анализа медицинских изображений, включая компьютерные томографии. Разработка программы, направленной на использование новейших методов машинного обучения для обработки компьютерной томографии (КТ)[2][18][25][26] изображений, оправдывает себя поскольку обеспечивает возможность эффективных диагностических процедур и планирования терапии.

Применение новых технологий начинается всегда с более простых, но важных задач. Так часто возникают ситуации, когда врачам необходимо замещать части ребер[16][19]. Такие решения имеет свои сложности и трудности, не только со стороны реализации, но и со стороны проектирования,

так как необходимо учесть множество факторов, которые могут возникнуть в том числе на практике. Применение твердых материалов в качестве импланта для замещения костной ткани в ребрах несут за собой сложности:

1. Жесткое ребро мешает процессу дыхания;
2. Может травмировать соседние кости из-за существенной разницы упругих свойств материалов.

Чтобы минимизировать негативные последствия замещения ребра, предлагается создать щадящий метод восстановления. На основе снимков КТ получить при помощи новых методов обработки 3D модели с пористой структурой, которые будут обрастать костной тканью, а место крепления переработать, чтобы минимизировать возможные разрушения в месте соединения.

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ

Для решения задачи разработки технологии создания моделей ребер на основе компьютерной томографии необходимо решить много разных задач связанные с разными технологиями. Например, чтобы восстановить ребро, используя твердый материал (титан) для импланта, несет за собой сложность. Такой имплант может привести от трудности с дыханием, до разрешения ребра.

Для решения данной задачи необходимо выполнить следующее:

1. Программный этап:

a. Создание основы графического интерфейса:

- Определение базовых функций;
- Определение дополнительных функций;
- Логика работы приложения.

b. Применение ранее написанного кода для решения первичной задачи:

- Добавления к основному приложению возможностей просмотра и построения первичной оболочки органов;
- Добавление возможностей совмещенного просмотра результата моделирования и начальных данных.

c. Добавление новых функций по кластеризации.

2. Поиск оптимальной сетки:

- a. Определение типа пористой структуры;
- b. Определение пропускной способности пористой структуры;
- c. Поиск реализации в реальности;
- d. Проверка на нагрузки.

3. Поиск и подбор материала;

4. Подбор и разработка нового соединения.

ГЛАВА 2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Основная цель данного этапа является то, что необходимо, с использованием новых методов, обработать снимки КТ. Предполагается, что новые методы и подходы к обработке КТ снимков улучшат результат обработки.

Данная программа должна решать следующие задачи:

1. Просматривать снимки;
2. Проводить легкую обработку снимка как:
 - а. Кропирование;
 - б. Изменение контраста;
 - с. Установка уровней.
3. Разделять на разные области снимок при помощи машинного обучения;
4. Конвертировать КТ снимок в mesh-модель;
5. Работать с Mesh-моделью (проводить постобработку).

Пункт 3 является самым важным, так как именно модели машинного обучения могут помочь врачам в обработке и анализе снимков.

2.1. Базовая разработки

Разработку программы можно разделить на несколько отдельных этапов.

В качестве основного языка программирования был выбран Python. По следующим причинам:

- На данном языке ведется быстро разработка;
- Есть все необходимые базовые библиотеки для работы с КТ снимками;
- Код, который был ранее написан в бакалаврской работе[11], был написан на этом языке программирования.

Для основы графического приложения была использована библиотека PyQt5[22]. Сейчас уже есть 6 версия этой библиотеки. Основные плюсы именно этой библиотеки следующие:

- Библиотека написана на компилируемом языке программирования, что позволяет работать данной библиотеки быстро;
- Поддержка основных платформ, таких как: Windows, macOS, Linux и другие UNIX-подобные ОС;
- Есть классы, позволяющие использовать OpenGL и 3D-графику в приложениях PyQt.

Помимо этой библиотеки были определены сразу следующие:

- NumPy[27] – Отвечает за хранение снимков;
- Pydicom[29] и pylibjpeg – Отвечают за считывание и обработку КТ снимков.

2.1.1. Иконки

Чтобы пользователю было проще взаимодействовать с приложением необходимы графические элементы, такие как иконки разных инструментов. Было решено, что иконки будут именно векторного типа, так как такие иконки позволят легко менять масштаб системы и самого приложения, без особого искажения самих иконок, за исключением растеризации.

Для реализации было выбрано приложение Adobe Illustrator. Данное приложение работает только с векторной графикой, что как раз идеально подходило для решения этой задачи.

Размер иконок был выбран 50x50px. Данного размера вполне достаточно чтобы считать саму иконку человеком и при этом же она не слишком маленькая.

На начальном этапе были выделены иконки для следующих функций:

- Принять;
- Отменить;
- Настройки;
- Помощь;
- Экспортировать;
- Импортировать;

- Открыть;
- Новый;
- Сохранить;
- Выйти;
- Контраст;
- Кропирование;
- Масштаб;
- Выделить;

Данные иконки в том же порядке представлены на рисунке 2.1.



Рис. 2.1 – Иконки

В представлении 50x50px представлен на рисунке 2.2.



Рис. 2.2. – Иконки в представлении 50x50px

Из результатов видно, что иконки различимы и легко читаются.

2.1.2. Общая навигация

Для реализации самого приложения использовалось ООП, то есть объектно-ориентированное программирование. И в данном случае это говорит о том, что почти любая функция, любое действие и само приложение – это объект какого-либо класса, который имеет свои методы реализации, свои функции и пр.

Чтобы легко ориентироваться пользователю в приложении и при этом же четко понимать, что он делает было определено следующее:

- Должна быть область навигации по снимкам;
- Пользователь может работать с несколькими снимками и для легкого понимания что пользователь смотрит, было решено что каждый снимок будет иметь перевью списке снимков;
- Область просмотра, где отображается текущий снимок и его любые изменения;
- Инструменты работы с текущим КТ снимком;
- Инициализатор всего приложения представлен в листинге 2.1.

Листинг 2.1. – Инициализатор приложения

```
def __init__(self, parent=None):
    """Initializer."""
    super().__init__(parent)
    self.setWindowTitle("Heart Viewer; V " + version)
    self.resize(1280, 720)
    self.showMaximized()
    self.setStyleSheet(f"background-color: {colors[T][0]};"
                       f"color: {colors[T][4]}")
    self.timer = QtCore.QTimer()
    self.timer.setSingleShot(True)
    self.timer.setInterval(250)
    self.Name_else = ""
    self.col = 0
    self.check = set()
    self.centralwidget = QtWidgets.QWidget(self)
    self.verticalLayout = QtWidgets.QVBoxLayout(self.centralwidget)
    self.verticalLayout.setObjectName("verticalLayout")
    self.verticalLayout.setContentsMargins(2, 2, 2, 2)
    self.verticalLayout.setSpacing(0)
    self.Mode_Decom = [None, None]
    self.Checks = []
    self._createActions()
    self._createMenuBar()
    self._createToolBars()
    #self._createContextMenu()
    self._createStatusBar()
    self._connectActions()
    #self.contextMenuEvent()
    self._stack_w()
    self._dockWid()
    self.new_forms()
    self.buttons = []
```

В листинге 2.1 как раз видно, что инициализируются все необходимые навигационные области: окно с просмотром, окно со списком снимков, кнопки с

инструментами. А также есть статус бар, контекстное меню (при нажатии на правую кнопку мыши открывается окошко с дополнительными инструментами).

На рисунке 2.3. представлен графический интерфейс программы.

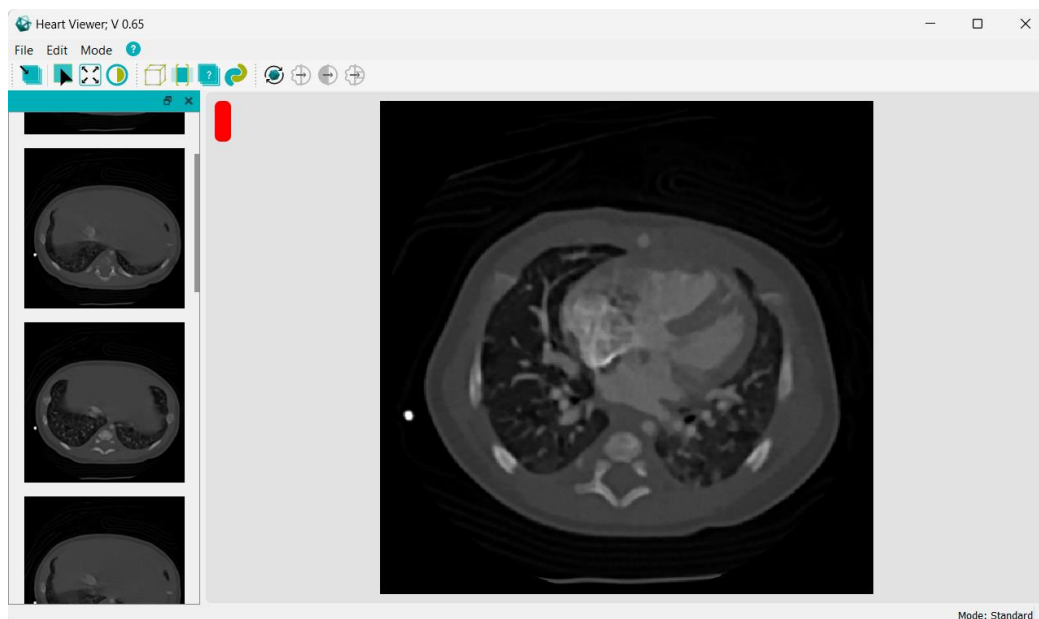


Рис. 2.3. – Графический интерфейс

2.1.3. Чтение файлов КТ

Для считывания файлов необходимо создать метод, который бы это делал. В большинстве случаев форматы хранения КТ снимков делятся на 2 типа:

1. Каждый снимок лежит в своей папке со своими срезами и общей информацией;
2. Все снимки со своим срезами и общей информацией о снимке лежат в общей папке. В этой папке не обязательно лежит файл прочтения этих снимков;
3. Снимки могут быть названы как угодно вне зависимости от их порядка и других снимков.

С этими причинами затруднена задача считывания таких снимков. В листинге 2.2. представлена функция считывания файлов, их сортировка и разделения на отдельные серии снимков.

Листинг 2.2 – Функция считывания файлов

```

def read_files_INTTO(self, id, path):
    mess_of_errors = []
    self.InWorking()
    self.progressBar.setValue(0)
    k, nm = 1, 1
    def readimgs(path):
        global mess_of_errors
        ids = []
        list_img = []
        subdir, files = lists(path)
        if len(subdir):
            ln = len(subdir)
            j = 0
            for folders in subdir:
                j += 1
                vol = 100 * j // ln
                self.progressBar.setValue(vol)
                subsubdir, subfiles = lists(folders)
                if subsubdir != []:
                    mess_of_errors.append(subsubdir)
                    continue
                else:
                    it_can = 0
                    for fls in subfiles:
                        try:
                            img = pd.read_file(fls)
                            it_can = 1
                            break
                        except:
                            pass
                    if it_can == 1:
                        ids.append(0)
                        list_img.append(folders)
        else:
            NumOfSer = []
            help_arr = []
            i = 0
            ln = len(files)
            for imgs in files:
                i += 1
                vol = 50*i*nm//(ln*k) + 50*(nm-1)//k
                self.progressBar.setValue(vol)
                img = pd.read_file(imgs)
                try:
                    cur_Num = img.get(0x00200011).value
                    numOfImg =
str(img.get(0x00200013).value)
                    tru_list = [numOfImg, imgs]
                    if (cur_Num in NumOfSer):
                        index1 = NumOfSer.index(cur_Num)
                        help_arr[index1].append(tru_list)
                    elif (cur_Num not in NumOfSer):
                        NumOfSer.append(cur_Num)
                        help_arr.append([tru_list])
                except:
                    pass
            ln = len(help_arr)
            j = 0
            for IMGs in range(len(help_arr)):
                j += 1
                vol = 50*j*nm//(ln*k) + 50*(nm-1)//k + 50//k
                self.progressBar.setValue(vol)
                indexes = []
                add_list = [""] * len(help_arr[IMGs])
                for paths in help_arr[IMGs]:
                    indexes.append(paths[0])
                for idd in range(len(help_arr[IMGs])):
                    try:
                        index = indexes.index(str(idd + 1))
                        add_list[idd] = help_arr[IMGs][index][1]
                    except:
                        pass
                list_img.append(add_list)
                ids.append(1)
            return ids, list_img
        id_type = []
        Outer_list = []
        Main_List = []
        _obj = 0
        if id == 0: # Стандартный режим работы
            id_type, Main_List = readimgs(path)
        self.win_w.close()
        return id_type, Outer_list, Main_List, _obj

```

Данная функция позволяет считывать все файлы, хранящиеся в папке. Если программа понимает, что там набор папок, то чтобы ускорить процесс считывания, она не проверяет каждый файл, а лишь помнит, что там могут быть снимки и записывает в память. Если все снимки в 1 папке находятся, то приходится проверять каждый файл, и в последствии сортировать снимки, сопоставлять один срез с другим, определять конкретный снимок КТ.

2.1.4. Сохранение и открытие файлов

Для сохранения текущего проекта была использована библиотека `pickle`. Основные преимущества этой библиотеки то, что она позволяет как бы архивировать файлы. То есть собирать множество других файлов и хранить в одном файле в бинарном формате, что позволяет экономить место.

Было выделено 3 основных формата для реализации, но реализован только 1 формат, так как остальные форматы не были столь необходимы.

В файле проекта должна была храниться следующая информация:

- Версия программы;
- Язык приложения;
- Стилль приложения;
- Режим работы приложения;
- 3D объекты;
- Изменение снимки;
- Пути до всех снимков проекта.

Данные параметры позволяли открыть однозначно проект с теми настройками, с которыми пользователь сохранил, и не создавать файл проекта большим по потребляемой памяти, так как в файле хранятся только пути до снимков, а не сами снимки.

В листинге 2.3. представлена функция, отвечающая за сохранение файла проекта.

Листинг 2.3. – Функция сохранения проекта

```
def savefile(path, type):
    global Outer_list, Main_list, _obj, Main_Matrix, id_type_ct, T, IdL, id_M
    ver = ["Version==1.0"]
    try:
        if "(*.cts)" in type:
            tosave = [IdL, id_M, T, id_type_ct, Main_list, Outer_list, Main_Matrix, _obj]
            with open(path, 'wb') as filehandle:
                pickle.dump(ver, filehandle)
                pickle.dump(tosave, filehandle)
    except:
        print("Error to read the file!")
```

В листинге 2.4. представлена функция, отвечающая за открытие файла проекта.

Листинг 2.4. – Функция открытия проекта

```
def openfile(path):
    global Outer_list, Main_list, _obj, Main_Matrix, id_type_ct, T, IdL, id_M
    nn = path.split(".")
    try:
        if nn[1] == "cts":
            with open(path, 'rb') as filehandle:
                ver = pickle.load(filehandle)
                if ver == ["Version==1.0"]:
                    toopen = pickle.load(filehandle)
                    IdL = toopen[0]
                    id_M = toopen[1]
                    T = toopen[2]
                    id_type_ct = toopen[3]
                    Main_list = toopen[4]
                    Outer_list = toopen[5]
                    Main_Matrix = toopen[6]
                    _obj = toopen[7]
                    toopen = []
            else:
                print("Version of the file here than may to read")
    except:
        print("Error to read the file!")
```

2.1.5. Языковые пакеты

Для того чтобы пользователю с иным языковым восприятием было проще ориентироваться в программе, было решено создать языковые пакеты. В каждом отдельном файле должны были быть прописаны названия всех функций, подсказок и каких-либо сообщений на том языке, на котором пользователь будет работать.

Языковые файлы сохраняются через ту же библиотеку, через которую производится сохранение проекта.

В листинге 2.5. представлена функция совместности языковых пакетов в приложении.

Листинг 2.5. – Совместность языковых пакетов

```

def readwriteandshow(list, name):
    with open(name, 'wb') as filehandle:
        # сохраняем данные как двоичный поток
        pickle.dump(list, filehandle)
    with open(name, 'rb') as filehandle:
        # открываем данные как двоичный поток
        placesList = pickle.load(filehandle)
    lang_s = [English, Russian]
def lang(i, j, k):
    try:
        if lang_s[i][j][k] != "":
            return lang_s[i][j][k]
        else:
            return lang_s[0][j][k]
    except:
        return lang_s[0][j][k]

```

Базовый этап разработки на этом заканчивается. Основные сложности, связанные с восприятием итогового результата, были закончены. На этом этапе программа может читать снимки, сохранять и открывать свои проекты. Остальные функции уже дорабатываются и добавляются к уже написанному коду.

2.2. Расширенная разработка

К расширенному этапу разработки относятся даже базовые функции, такие как: контрастность, масштаб, кропирование. Самые интересные и важные функции реализуются на этом же этапе.

Базовые функции: контрастность, масштаб и кропирование, не будут рассматриваться подробно в связи с тем, что данные функции интуитивно понятны.

2.2.1. Конвертация в Mesh

Одна из важных функций программы – это конвертация в сетчатую модель, то есть полигональную модель, которую в последствии можно будет напечатать на 3D принтере.

Данный алгоритм основан на методе Marching Cubes[31], подробно было рассказано еще в бакалаврской работе[11]. Этот алгоритм был реализован и

доработан в ЯП C++ и в последствии встроен, как библиотека в Python к основной программе. Для реализации использовалась библиотека `pybind11`[20]. Обусловлено это тем, что C++ является компилируемым языком программирования, в то время как Python интерпретируемым языком программирования. И как следствие C++ в разы быстрее в математических расчетах.

Чтобы подключить функцию из C++ в Python, необходимо было воспользоваться библиотекой `pybind11`.

В написанной библиотеке была доработана функция слияния вершин. Как итог, выходным файлом был файл формата `.obj`[30].

2.2.2. Декомпозиция

Функция декомпозиции позволяет автоматически разбить области в снимке на основе плотностей материала: по значениям вокселей.

Функция декомпозиции основана на машинном обучении. Главная задача – это кластеризация.

Так как изначальной базы данных, где есть связка снимка с ее моделью, нет. Стоит отметить, что такую базу данных вручную не получить за приемлемые сроки одному человеку, поэтому было решено использовать модель обучения без учителя.

В качестве основного метода был использован `kmeans`[21][3], или же метод `k`-средних. Данный метод является наиболее популярным. Был изобретён в 1950-х годах математиками Гуго Штейнгаузом и Стюартом Ллойдом.

Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров.

Пример его работы приведен на рисунке 2.4.

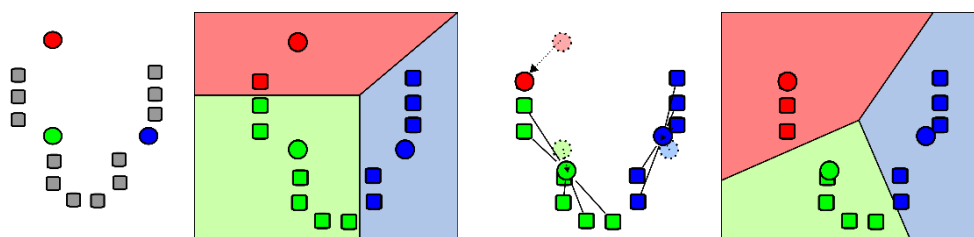


Рис. 2.4. – Пример работы алгоритма

Суть его такова, что в численные данные добавляются случайно той же размерности центры кластеров в количестве выбранным пользователем. К каждому центру кластера, по квадратичным отклонениям, определяются какие именно данные к ним относятся. На основе выделенных кластеров, в первой итерации, определяется их среднее значение и значение центра кластера переопределяется. Так делается несколько итераций, пока отклонение кластера между итерациями будет изменяться в соответствии с требованиями отклонения.

Данный метод сам по себе простой, позволяет получить весьма хорошие результаты за приемлемый срок. Однако есть у данного метода минус: сам по себе метод не определяет количество возможных кластеров. Но зная на основании каких данных будет проводиться кластеризация можно определить количество этих кластеров.

Так как в снимках КТ биологические ткани, то можно определить количество разных материалов и можно исходить из этих соображений.

В листинге 2.6. представлен часть кода где загружается снимок КТ для кластеризации.

Листинг 2.6. – Декомпозиция

```

def DistractEdit(self):
    global id_M, Crops, Limits, Main_Matrix, add_Main_Matrix
    self.col = self.numOfZones.value()
    print(self.col)
    if (self.col == 0 or self.col == 1) and self.Mode_Decom[0] != mtds.auto:
        error = QMessageBox()
        error.setWindowTitle("Error")
        error.setText("Необходимо указать больше чем 1 зону")
        error.setIcon(QMessageBox.Information)
        error.setStandardButtons(QMessageBox.Ok)
        error.exec_()
    else:
        self.Decomposition.close()
        self.DistractAction.setEnabled(True)
        clust = np.empty((Crops[0][1] - Crops[0][0], Crops[1][1] - Crops[1][0], Crops[2][1] - Crops[2][0]), np.int16)
        clust = Main_Matrix[Crops[0][0]:Crops[0][1], Crops[1][0]:Crops[1][1], Crops[2][0]:Crops[2][1]].copy()
        clust = (clust - np.min(clust))
        clus = clust.reshape((clust.shape[0] * clust.shape[1] * clust.shape[2], 1))
        if self.Mode_Decom[0] == mtds.none or self.Mode_Decom[1] == mtds.KMENCE:
            sc = KMeans(init='k-means++', n_clusters=self.col).fit(clus)
            self.Clc = sc.labels_
        elif self.Mode_Decom[1] == mtds.BIRCH:
            sc = Birch(branching_factor=200, threshold = 1).fit(clus)
            self.Clc = sc.predict(clus)
        self.Clc = self.Clc.reshape(Crops[0][1] - Crops[0][0], Crops[1][1] - Crops[1][0], Crops[2][1] - Crops[2][0])
        add_Main_Matrix = self.Clc
        self.viewer.setMatrix(matr = self.Clc, col = self.col, axis=None)

```

2.3. Сравнение программ

Для сравнения использовались разные программы по обработке КТ снимков [1][25][26] и преобразованию в Mesh-объекты. Первой программой для сравнения стала *Mimics Medicsal*.

Остальные программы, которые участвовали в сравнении такие как: *InVesalius*, не имели таких преимуществ как первая программа, в них реализовано только определение границ значений, масок и заполнение дырок в оболочке модели, результат от этого хуже не станет, кроме того, что будет поверхность модели более зашумлена и вес модели будет на много больше, раза в 3-4, около 140 Мб формата .obj.

Первая программа позволяет обрабатывать снимки на уровне вокселей: проводить точечное закрашивание зон, имеет стандартные методы указания интересующих областей (Определение границ значений), а также можно применить различные маски обработки снимков, как для 2d снимков.

На уровне Mesh-объекта данная программа позволяет сгладить объект.

На рисунке 2.5. представлены инструменты, которые применялись для получения результата: первая панель (зеленые инструменты) применялась на уровне вокселей, а вторая панель (синие инструменты) применялась на уровне Mesh-объекта

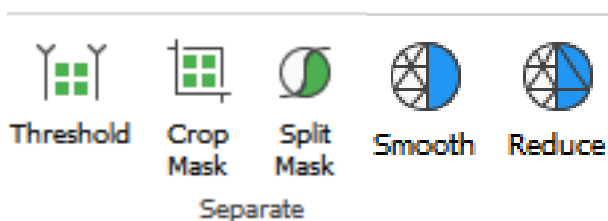


Рис. 2.5. – Инструменты

Однако, стоит отметить, что существует инструмент с автоматическим поиском разных зон, по применению этого инструмента и по описанию это заранее выставленные настройки для определения зон основанные на применении масок для обработки снимков. Данный инструмент работает не так хорошо, как можно было бы. Пример работы такого инструмента представлен на рисунке 2.6.

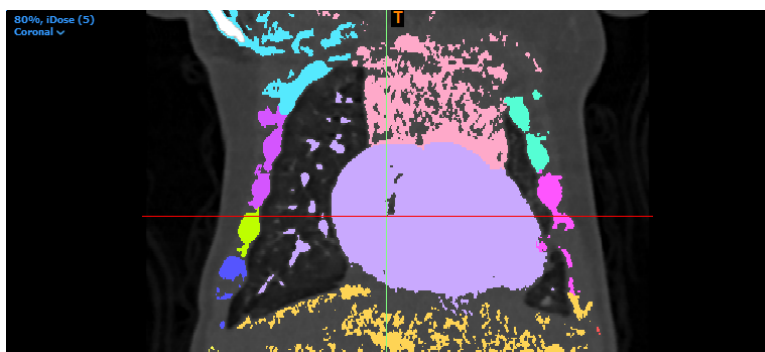


Рис. 2.6. – Автоматическое выделение зон Mimics Medical

На рисунке 2.7. представлен результат 3d модели в сечении полученный лишь только в этой программе, вес данной модели получился 38Мб формата .obj, также стоит отметить, что данную модель необходимо доработать, так как эта модель не пригодна для печати, в ней существуют пустоты в оболочке:

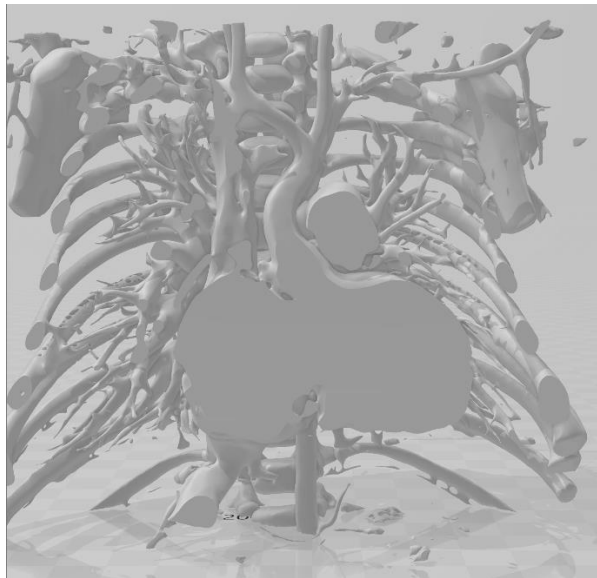


Рис. 2.7. – Результат моделирования Mimics Medical

2.3.1. Преимущества программы

На уровне вокселей существуют стандартные инструменты: определение границ, обрезка. Однако для дальнейших действий с объектами на много проще проводить более точную обработку еще до уровня Mesh. Поэтому на более низком уровне была встроена Декомпозиция.

Данный инструмент, как писалось ранее, позволяет указать сколько отдельных зон существует, а также программа должна будет самостоятельно разделить снимок на отдельные кластеры.

На рисунке 2.8. представлен результат работы инструмента декомпозиции, лучшим вариантов оказалось разделить на 4 зоны, здесь показана 3 зона.



Рис. 2.8. – Результат декомпозиции разрабатываемой программы

Все этапы обработки снимка и получение итоговой 3D модели укладываются в 10 минут рабочего времени.

На рисунке 2.9. представлен результат, полученный с применением этих инструментов. Вес модели получился около 10Мб формата .obj.

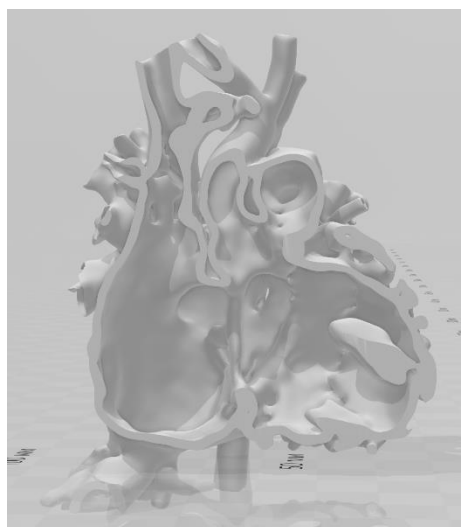


Рис. 2.9. – Результат моделирования разрабатываемой программы

ГЛАВА 3. АДДИТИВНЫЕ ТЕХНОЛОГИИ

Прежде чем подбирать структуру и материал необходимо рассмотреть разные технологии реализации моделей. Соответственно изготовление любого материального предмета, в данном случае это ребра со сложной структурой, их можно сделать всего двумя способами: литье или постепенное наращивание материала. При первом способе изготавливаются металлические модели, что не удовлетворяет исходной задаче. А вот уже вторая технология подходит по основным параметрам.

3.1. Экструзионная печать (FDM, FFF, FGF)

Самая распространенная, популярная, простая, и соответственно недорогая технология это FDM[14][23][24]. Принцип работы проиллюстрирована на рисунке 3.1.

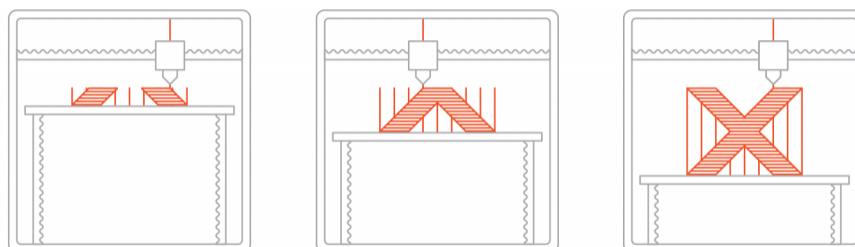


Рис. 3.1 – FDM печать

Через движущееся по заданной траектории сопло выдавливается расплав полимера, который, застывая, послойно наращивает нужную нам деталь.

Основные плюсы данной технологии:

- Лучший способ для быстрого прототипирования;
- Можно печатать разными пластиками;
- Можно создавать функциональные прототипы, свойства которых будут близки к свойствам конечного изделия;

- Себестоимость этих прототипов может быть очень низкой по сравнению с классическими технологиями фрезерования или использования пресс-форм;
- Можно быстро создавать модели сложных геометрических форм, используя растворимые субстанции в моделях принтеров с двумя экструдерами;
- На рынке доступны сотни материалов для 3D-печати;
- Большая накопленная база данных с данной технологией.

Основные минусы данной технологии:

- Низкое качество моделей.
- Печать термопластичными материалами через экструзионные сопла приводит к проблемам с допусками и точностью моделей.

3.2. Стереолитография (SLA, DLP)

Данная технология заключается в послойном отверждении в емкости жидкой фотополимерной смолы[9][3]. Принцип работы проиллюстрирован на рисунке 3.2.

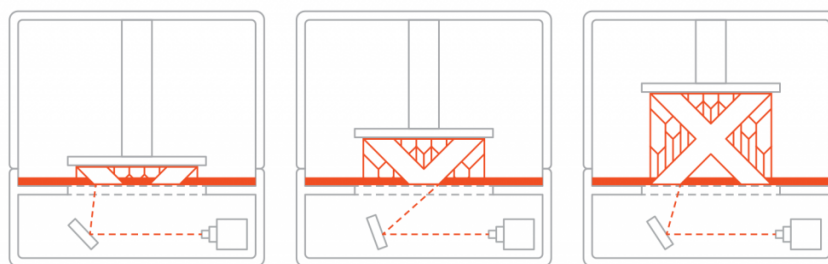


Рис. 3.2. – SLA печать

В качестве “отвердителя” используется ультрафиолетовая засветка лазерным лучом (SLA) или светодиодной матрицей (DLP). В первом случае печать происходит быстрее и точнее, во втором требуется куда более компактное и дешевое оборудование.

Плюсы:

- Высокая точность.

Минусы:

- Небольшая скорость печати;
- Дорогие расходные материалы;
- Трудоемкая подготовка и уборка принтера после печати;
- Небольшой выбор смол;
- Хрупкость готового изделия.

3.3. Другие технологии печати

К другим технологиям печати, которые не были рассмотрены в данной работе это:

- Струйная печать (CJP, PolyJet);
- Струйно-порошковая 3D-печать;
- Селективное лазерное спекание (SLS, SLM, DMLS);
- 3D Наплавление (DED, LENS, DMD, LBAM, WAAM, EBAM и тд);
- Ламинирование (LOM, UAM, CBAM);
- Ротационная сварка трением (FSAM);
- Холодное напыление.

Данные технологии либо недоступны, либо дорогие, либо не удовлетворяют основным требованиям данной задачи, поэтому эти технологии не были рассмотрены вовсе.

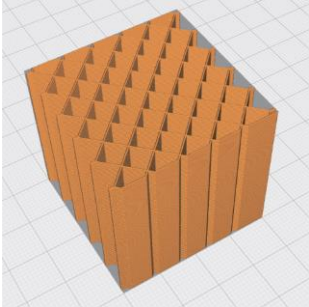
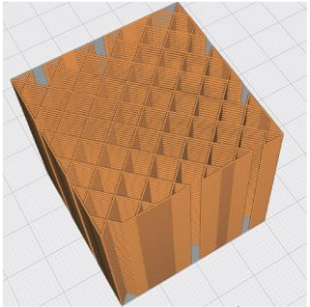
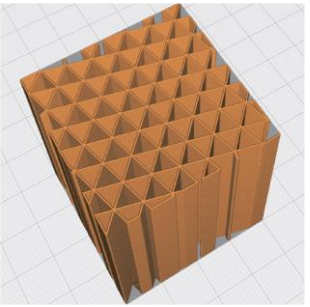
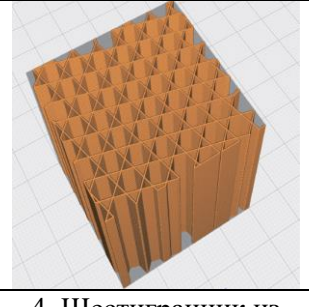
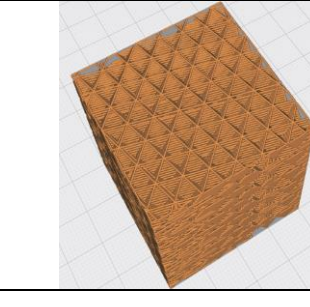
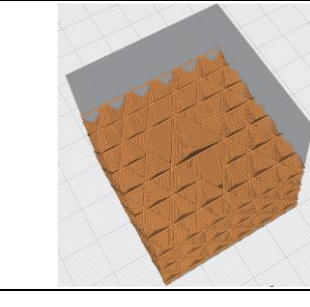
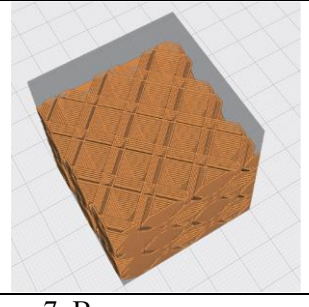
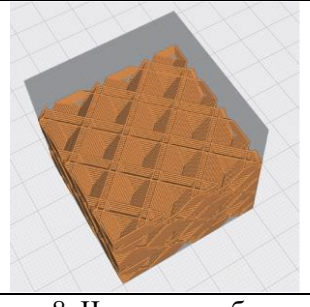
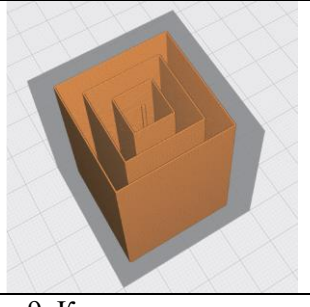
ГЛАВА 4. ВОЗМОЖНЫЕ РЕАЛИЗАЦИИ В ПЕЧАТИ

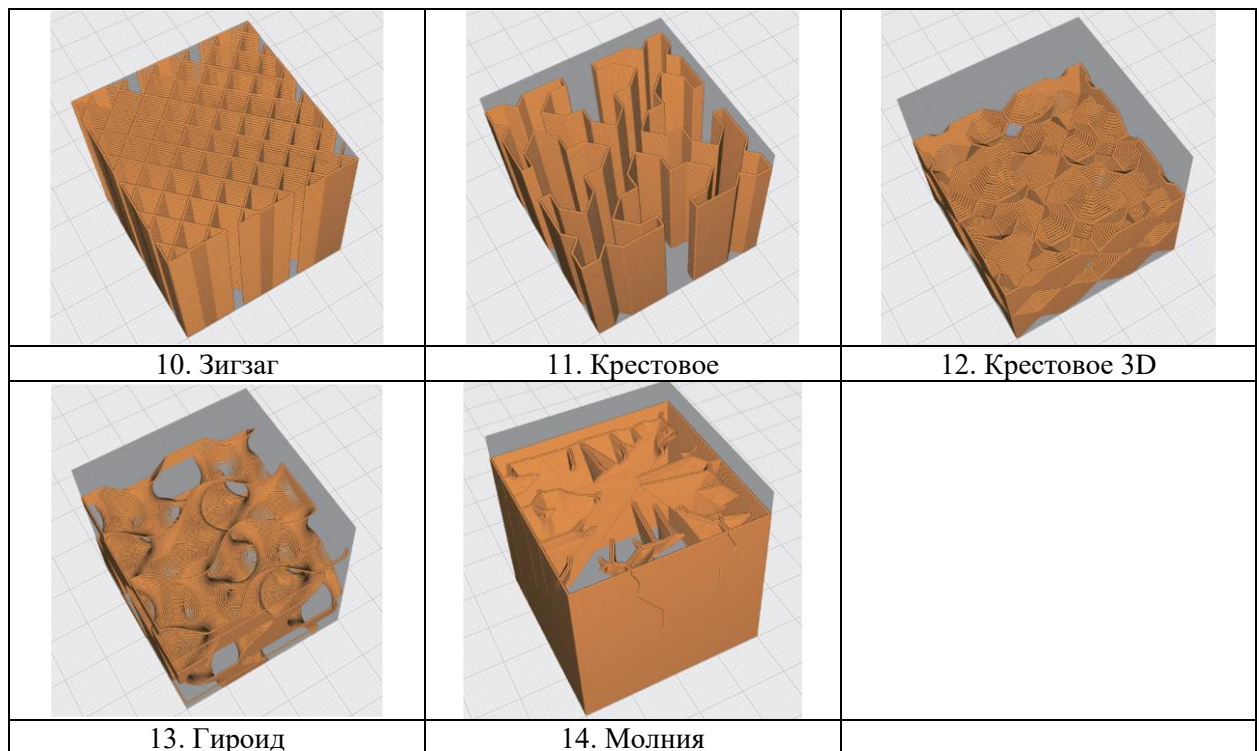
4.1. Сетки в реализации FDM

Основной задачей является определение и исследование структуры заполнения модели при печати.

В стандартных приложениях по печати, в такой как Ultimaker Cura[32] уже есть заготовленные варианты заполнений, которые могут быть интересны и полезны при реализации главной задачи данной работы. Типы сеток представлены в таблице 4.1.

Таблица 4.1 – Типы сеток FDM

		
1. Сетка	2. Линии	3. Треугольник
		
4. Шестигранник из треугольников	5. Куб	6. Динамический куб
		
7. Восьмигранник	8. Четверть куба	9. Концентрическое



Рассмотрим сетки подробнее:

- Заполнения типа 1, 2, 3, 4, 9, 10 и 11 однонаправленно. То есть они не обладают изотропным свойством. И данные типы заполнений не позволят решить основную задачу.
- Заполнение 14 имеет резкое уплотнение в одном из краев
- Заполнения типа 5, 6, 7, 8, 12, 13 изотропны и являются кандидатами для реализации

Из оставленных сеток можно рассмотреть более подробно:

- Заполнение 5: равномерная, изотропная
- Заполнение 6: по середине имеет более разраженную структуру, похожа на заполнение 5
- Заполнение 7: похож на 5
- Заполнение 8: похож на 5
- Заполнение 12: изотропный, кубический
- Заполнение 13: изотропный, кубический, но сглаженный

Из этих сеток только 12 и 13 обладают открытой структурой, остальные являются закрытыми и через себя не могут пропускать что либо, в том числе клетки.

Для исследования оптимальны только эти 2 типа сеток.

4.2. Сетки в реализации SLA

В технологии SLA нет готовых решений заполнения. Поэтому данные типы сетки необходимо делать на уровне Mesh модели.

ГЛАВА 5. ИССЛЕДОВАНИЕ СТРУКТУР

Для реализации импланта ребра необходимо соблюсти не только его форму и физические свойства, но и разработать новый метод восстановления. Сам по себе имплантат может мешать процессу дыхания и в конечном счете разрушать прилегающие ребра.

В качестве решения предлагается временно заменить часть ребра пока имплантат обрастает костной тканью. Чтобы такое было вообще возможно необходимо сам имплантат сделать пористым, с открытой структурой.

Примеры пористых структур представлены на рисунке 5.1.



Рис. 5.1. – Типы структур

Данные структуры являются открытыми, то есть через такие структуры способно проходить вещество: газы или жидкости. Остается вопрос в их пропускной способности.

5.1. Пропускная способность

Прежде чем решать ранее описанную задачу, необходимо определить на сколько подобные структуры способны пропускать через себя вещества. Дело в том, что требуется, чтобы данный имплант мог обрастать костной тканью, а значит должен выполнять свои главные функции.

Рассмотрим структуру по кубической сетке в двух вариантах. Обусловлено это тем, что почти все сетки основаны именно на этих сетках. Шаровая ячейка представлена на рисунке 5.2.

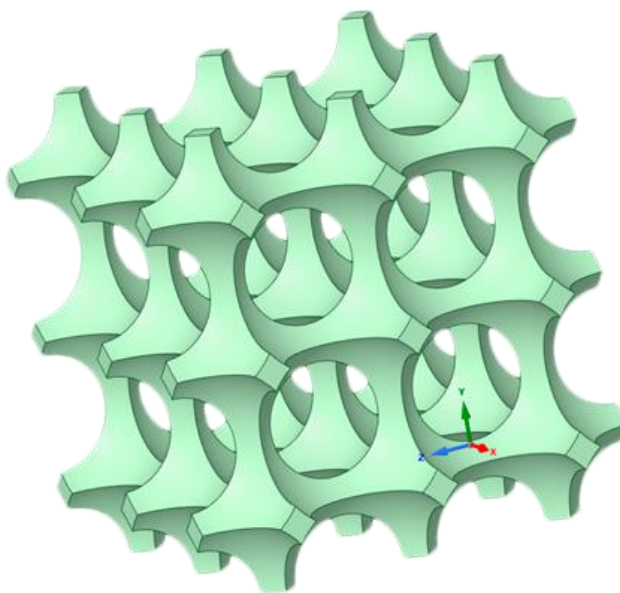


Рис. 5.2. – Шаровая ячейка

На рисунке 5.3. представлена кубическая ячейка.

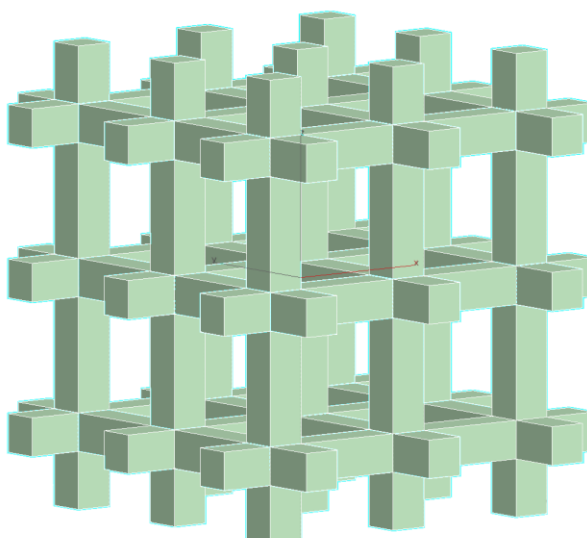


Рис. 5.3. – Кубическая ячейка

Для упрощения расчетов пропускной способности сеток рассмотрим лишь только 1 ячейку размером, соответствующей периодичности данной сетки. В данном случае это 1 мм^3 .

Все расчеты будут произведены в программе ANSYS 2022 R2.

5.1.1. Шаровая ячейка

Первой будет рассмотрена шаровая ячейка на пропускную способность. На рисунке 5.4. представлена расчетная модель

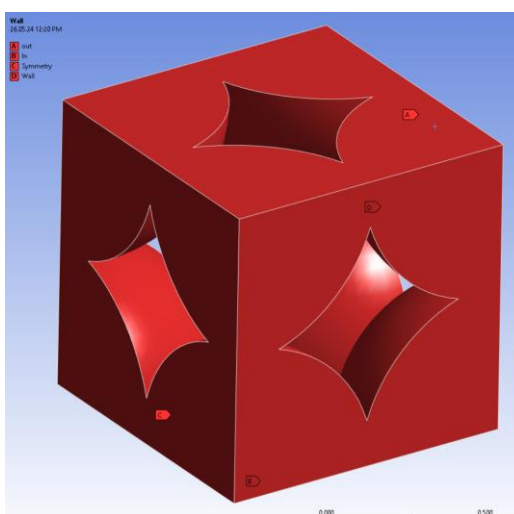


Рис. 5.4. – Расчетная модель шаровой ячейки

Верхняя поверхность – это выход, нижняя вход, боковые поверхности – это поверхности симметрии, а все остальные являются стеной, то есть материалом.

График зависимости перепад давления от скорости представлен на рисунке 5.5. при разных концентрациях.

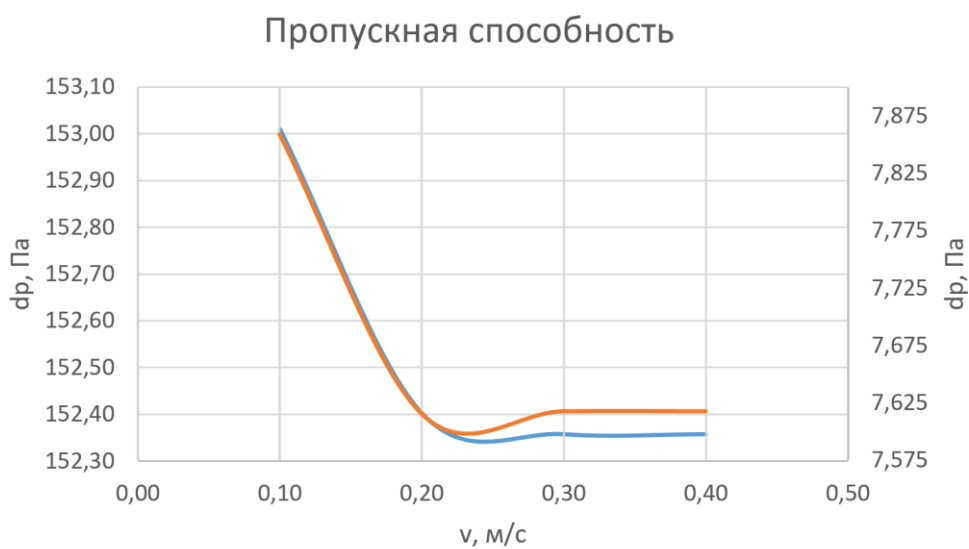


Рис. 5.5. – Пропускная способность шаровой ячейки

Оранжевая линия при концентрации 20%, правая шкала.

Синяя линия при концентрации 40%, левая шкала.

5.1.2. Кубическая ячейка

Граничные условия кубической ячейки аналогичны и представлены на рисунке 5.6.

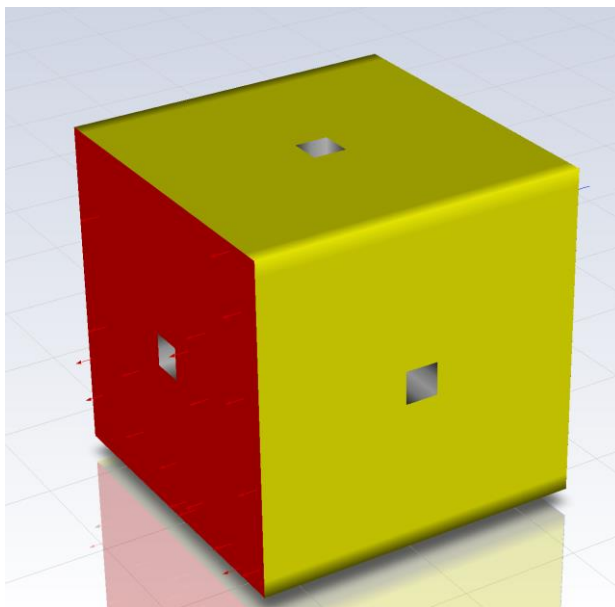


Рис. 5.6. – Расчетная модель кубической ячейки

Результаты кубической ячейки при тех же значениях, что и при шаровой ячейки: 20% и 40% представлены на рисунке 5.7.

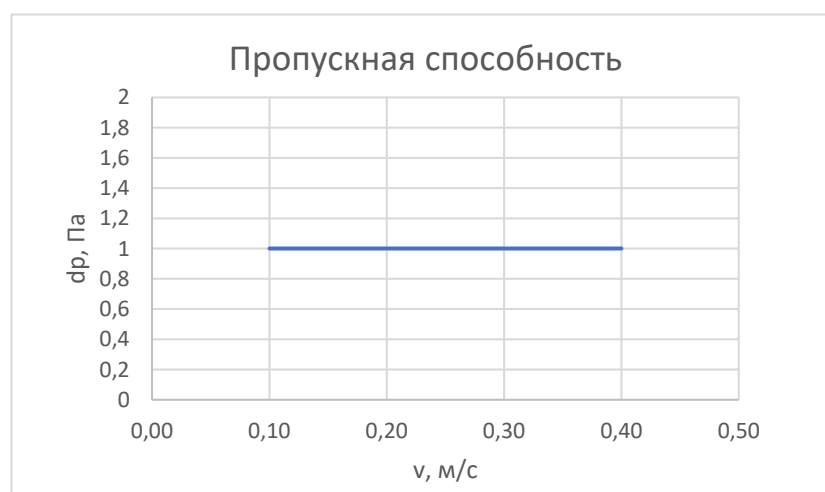


Рис. 5.7. – Пропускная способность кубической ячейки

5.2. Нагрузки

Теперь рассмотрим данные сетки на нагрузки. Способны ли они нести на себе нагрузку или нет. Если сетки могут на себе нести нагрузку, то какие параметры у композитного материала, чтобы можно было для расчетов заменить их на изотропный материал.

5.2.1. Кубическая ячейка

Необходимо рассмотреть эти же сетки на нагрузки. На рисунке 6.8. представлены граничные условия у кубической ячейки. Верхние поверхности подвергаются растяжению на $0,16Н$, нижние поверхности жестко закреплены, остальные поверхности остаются свободными, материал используется ABS. По причине того, что эти сетки будут напечатаны с этим материалом, и эти модели будут использованы для экспериментального исследования.

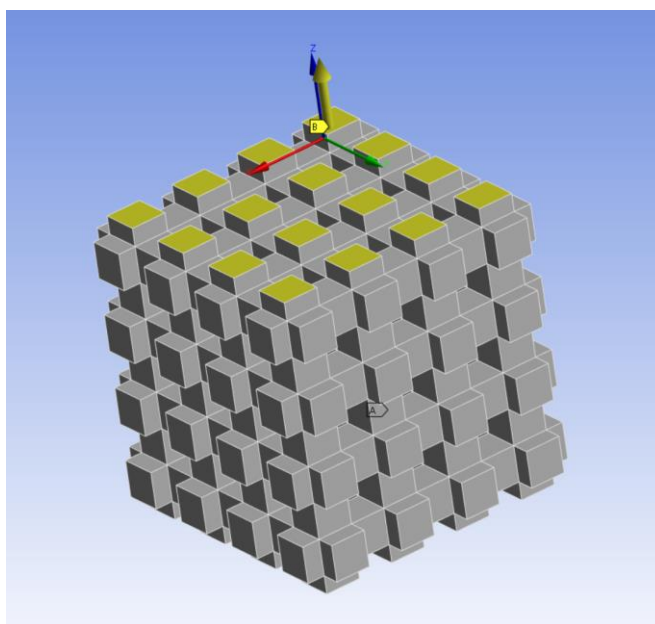


Рис. 5.8 – Расчетная модель кубической ячейки

Данную сетку необходимо варьировать по плотности, рассматривая физические свойства. Диапазон варьирования $(0; 1)$. То есть предельные значения приниматься не могут, так как при 0 у нас образуются стенки нулевой толщины, а при 1 сетка становится закрытой, то есть там сплошной материал, что не удовлетворяет требованиям.

На рисунке 5.9. представлены результаты исследования на растяжение, где синяя линия – это коэффициент запаса прочности, оранжевая деформации, серая напряжения.

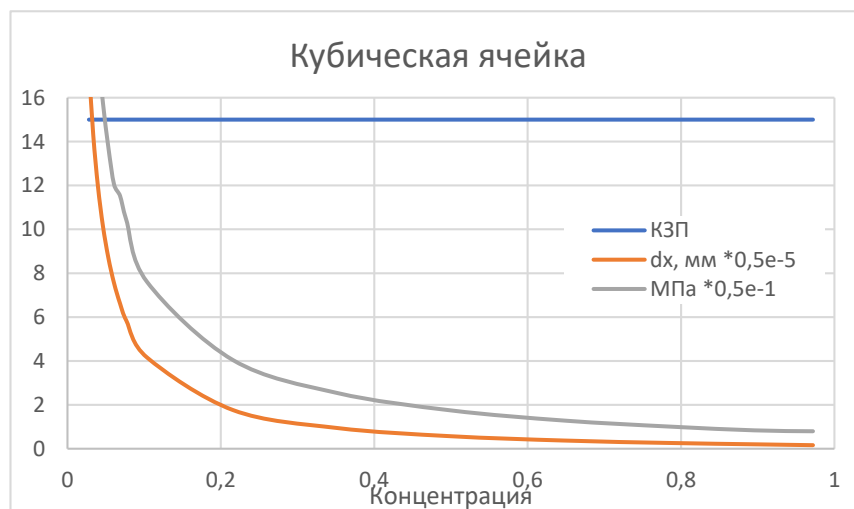


Рис. 5.9. – Растяжение кубической ячейки

На рисунке 5.10. представлены результаты исследования на сдвиговые деформации, где синяя линия – это коэффициент запаса прочности, оранжевая деформации, серая напряжения.

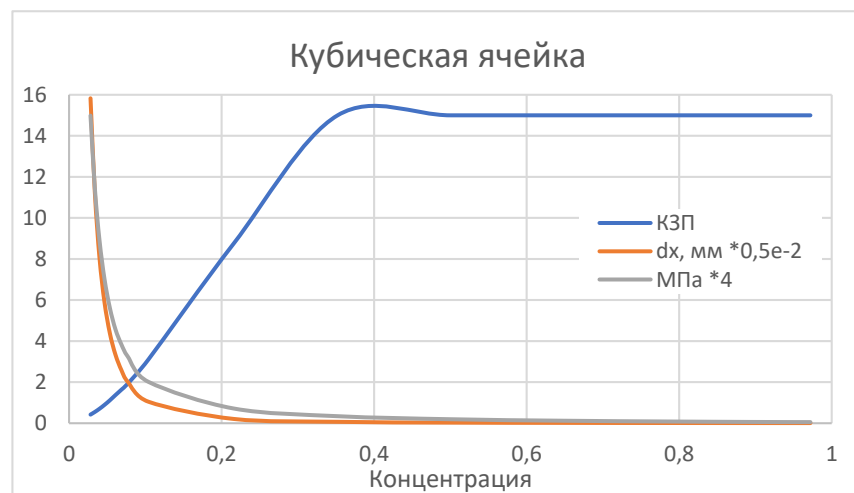


Рис. 5.10. – Сдвиг кубической ячейки

В таблице 5.1. представлены численные результаты этих же исследований.

Таблица 6.1. – численные результаты

С	Сдвиг			Растяжение		
	КЗП	Деформация, мм	Напряжения, МПа	КЗП	Деформация, мм	Напряжения, МПа
2,8%	0,42	3,17E-01	59,95	15	8,68E-04	1,14E+00
3,4%	0,54	2,25E-01	45,92	15	7,15E-04	1,05E+00
4,0%	0,70	1,61E-01	35,66	15	5,98E-04	9,07E-01
4,6%	0,89	1,18E-01	28,12	15	5,07E-04	7,88E-01
5,3%	1,11	8,78E-02	22,50	15	4,35E-04	6,87E-01
6,1%	1,37	6,66E-02	18,27	15	3,78E-04	6,03E-01
6,9%	1,63	5,15E-02	15,31	15	3,31E-04	5,76E-01
7,7%	1,93	4,03E-02	12,98	15	2,92E-04	5,21E-01
10,4%	3,11	2,09E-02	8,05	15	2,08E-04	3,80E-01
21,6%	8,76	4,08E-03	2,85	15	8,82E-05	2,02E-01
35,2%	15,00	1,30E-03	1,36	15	4,73E-05	1,27E-01
50,0%	15,00	5,41E-04	0,77	15	2,87E-05	8,75E-02
64,8%	15,00	2,70E-04	0,48	15	1,88E-05	6,43E-02
78,4%	15,00	1,55E-04	0,33	15	1,33E-05	5,07E-02
89,6%	15,00	1,01E-04	0,24	15	1,02E-05	4,21E-02
97,2%	15,00	7,44E-05	0,21	15	8,22E-06	4,00E-02

5.2.2. Шаровая ячейка

Следующая модель для проверки на нагрузки была с шаровой ячейкой. Нагрузки на данную сетку идентичны кубической сетки. Пример данной ячейки представлены на рисунке 5.11.

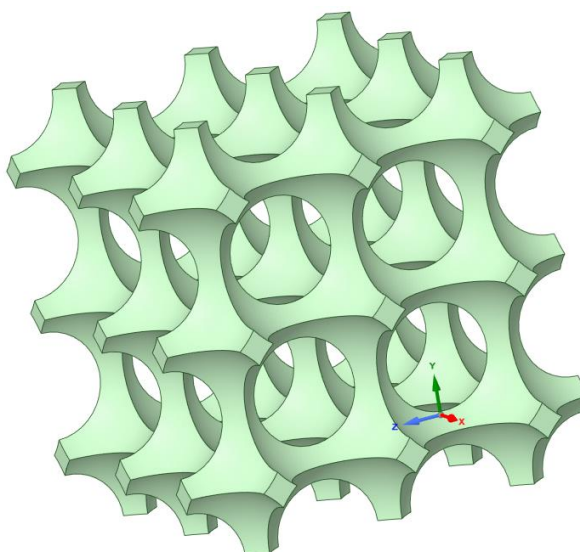


Рис. 5.11. – Шаровая ячейка

Расчеты произведены в ANSYS. Верхние поверхности подвергаются растяжению на $0,09H$, нижние поверхности жестко закреплены, остальные поверхности остаются свободными, материал используется ABS по той же причине.

Сетка варьируется по плотности. Рассматриваются физические свойства данного материала. Диапазон варьирования $(0,1; 0,475)$. То есть предельные значения приниматься не могут, так как при $0,1$ образуются стенки нулевой толщины, а при $0,475$ и более сетка становится, что не удовлетворяет требованиям исходной задачи

На рисунке 5.12. представлены результаты исследования на растяжение, где синяя линия – это коэффициент запаса прочности, оранжевая деформации, серая напряжения.

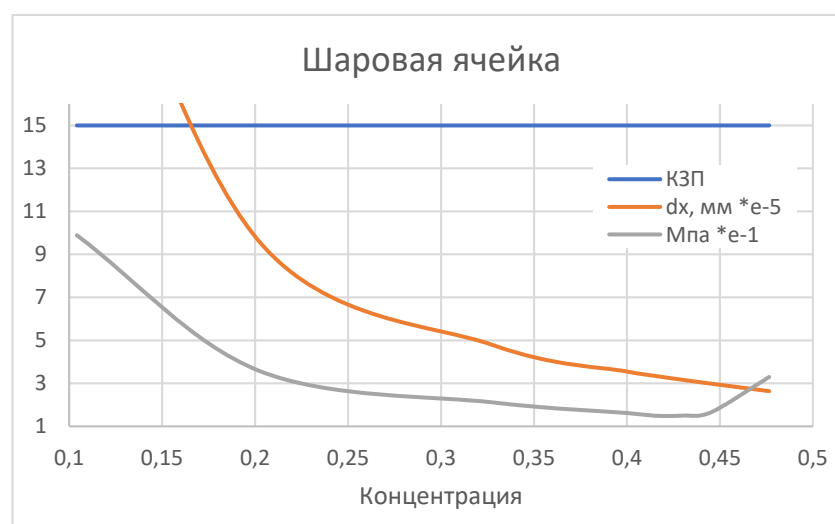


Рис. 5.12. – Растяжение шаровой ячейки

На рисунке 5.13. представлены результаты исследования на сдвиговые деформации, где синяя линия – это коэффициент запаса прочности, оранжевая деформации, серая напряжения.

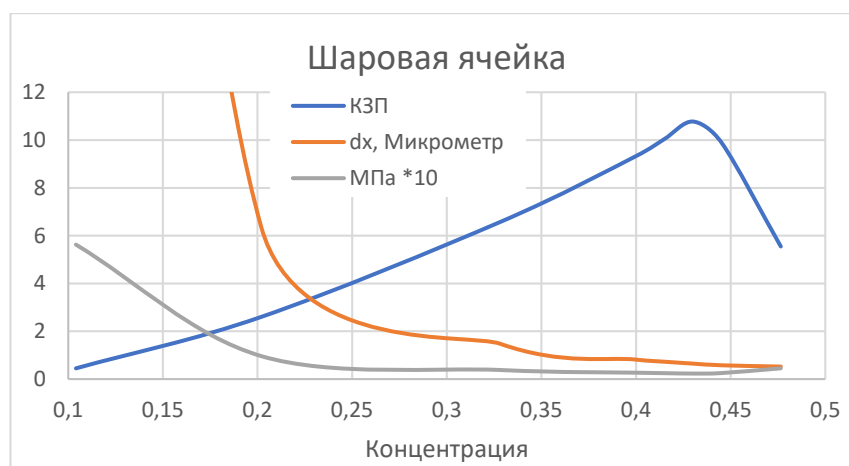


Рис. 5.13. – Сдвиг шаровой ячейки

В таблице 5.2. представлены численные значения этих же исследований.

Таблица 5.2. – численные результаты

С	Сдвиг			Растяжение		
	КЗП	Деформация, мм	Напряжения, МПа	КЗП	Деформация, мм	Напряжения, МПа
47,6%	5,55	5,20E-04	4,51	15,00	2,64E-05	3,29E-01
44,5%	9,88	5,75E-04	2,53	15,00	2,98E-05	1,65E-01
43,0%	10,78	6,45E-04	2,32	15,00	3,16E-05	1,50E-01
41,5%	10,04	7,24E-04	2,49	15,00	3,35E-05	1,49E-01
40,0%	9,33	8,13E-04	2,68	15,00	3,55E-05	1,62E-01
32,8%	6,57	1,48E-03	3,80	15,00	4,77E-05	2,11E-01
20,2%	2,60	6,35E-03	9,62	15,00	9,62E-05	3,59E-01
10,4%	0,44	6,49E-02	56,25	15,00	2,77E-04	9,89E-01

5.3. Получение модели в реальности

Для исследования физико-механических свойств, были созданы модели с разной структурой заполнения одним материалом.

На рисунке 6.14. представлена кубическая сетка в прямом направлении.

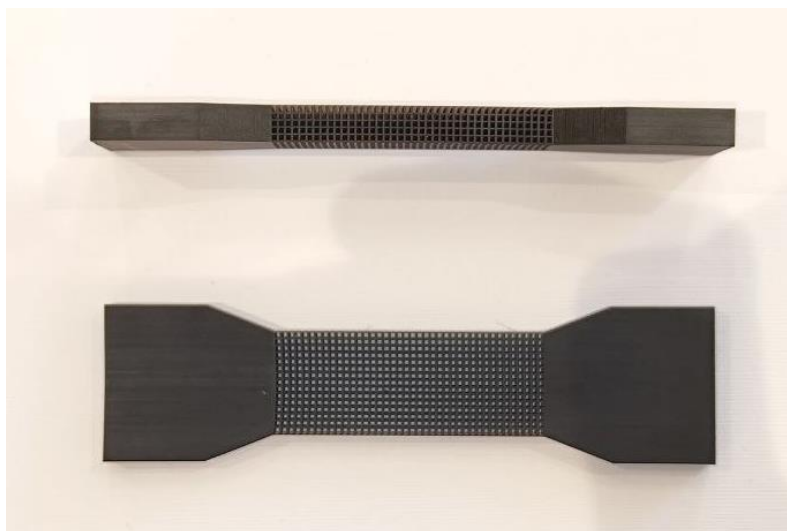


Рис. 5.14. – Кубическая сетка в прямом направлении

На рисунке 5.15. представлена кубическая сетка в повернутом направлении

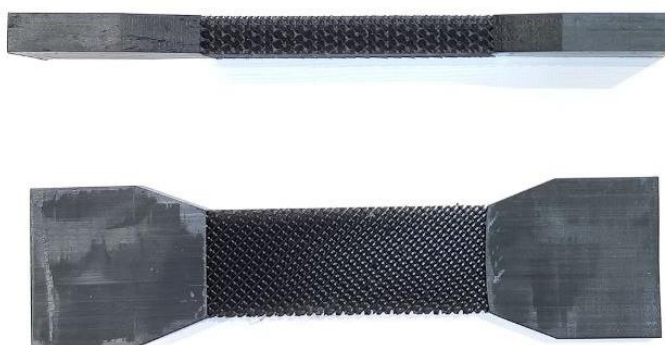


Рис. 5.15. – Кубическая сетка в повернутом направлении

5.3.1. Результаты исследований

Данные структуры были исследованы. Их можно будет проверить и сравнить с численным моделированием.

На рисунке 5.16. приложены зависимости напряжений от деформаций для кубической ячейки в 90° . Где пунктирная линия – это численные эксперименты, а сплошная это экспериментальные данные лопатки.

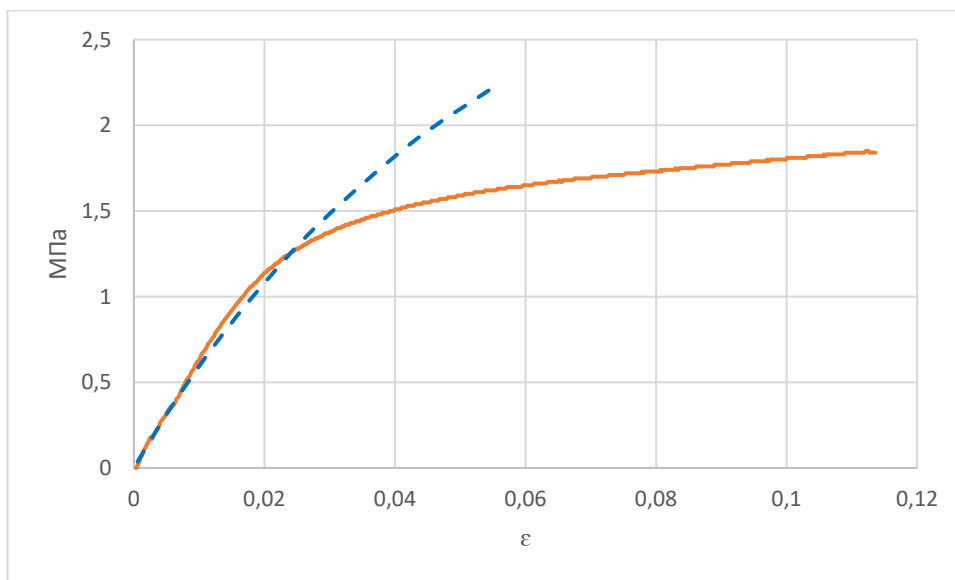


Рис. 5.16. – Кубическая сетка 90°

На рисунке 5.17 приложены зависимости напряжений от деформаций в сравнении для кубической ячейки в 90°(Серая линия) с 45°(Оранжевая линия).

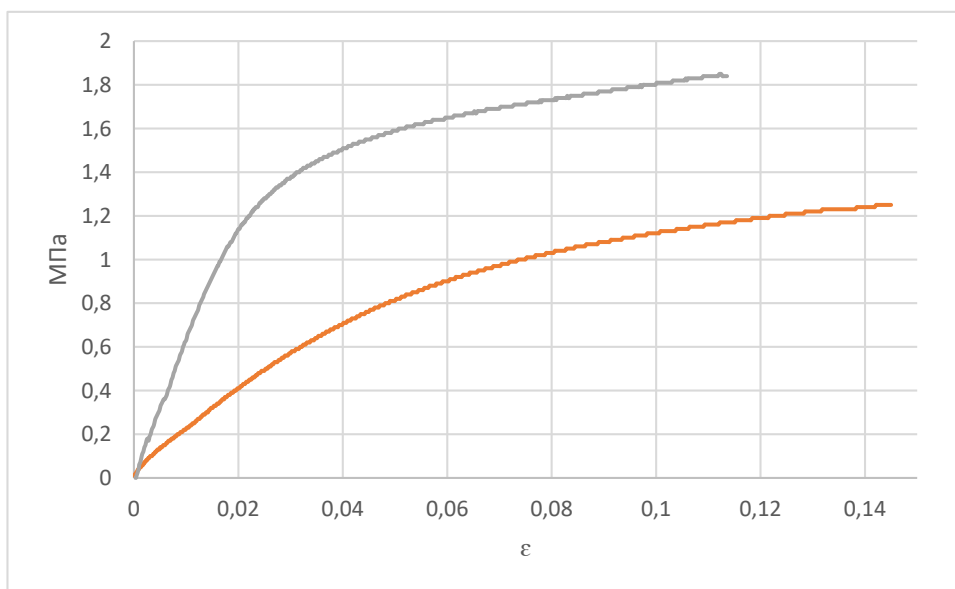


Рис. 5.17. – Сравнение экспериментальных данных

В таблице 5.3 представлены результаты экспериментальных данных

Таблица 5.3. – численные результаты

Скорость, мм/мин	Модуль Юнга, МПа	Предел текучести (Верхний), МПа	Деформация при разрушении, %	Прочность, МПа	Максимум Нагрузка, Н
5	20,16	1,25	14,57	1,24	93,63
5	66,71	1,85	11,38	1,83	138,48

5.4. Выводы по сеткам

Сетки были напечатаны одним материалом, но с разными направлениями: 90° и 45°. Как оказалось, второй вариант, где направления сетки был 45°, имел другие свойства:

1. Был более гибким;
2. Меньше подвергалась разрушениям.

Лопатка с прямым направлением в 90°:

1. Была более жесткая;
2. Легче разрушалась.

По экспериментальным данным внешние характеристики совпадают с результатами экспериментов, а результаты эксперимента совпадает с численным моделированием в ANSYS. Однако результаты рознятся при больших деформациях, связано это с тем, что материал при численном моделировании был линейными и других данных и моделей не было.

Второй вывод – это то что в итоге такая сетка не дает изотропный материал, он анизотропный, то есть свойства отличаются по направлениям. Что говорит, что необходимо усложнять численную модель.

ГЛАВА 6. ПОИСК СОЕДИНЕНИЙ

Для производства импланта для последующего внедрения необходимо решить следующие задачи:

1. Найти варианты закрепления двух трубчатых тел;
2. Сопоставить все плюсы и минусы;
3. Определить наилучший вариант соединения.

6.1. Типы соединений

На данный момент в конструкторской сфере существуют разные виды соединений. Ниже представлены все виды соединений, которые известны[22][4]:

1. Разъёмные соединения:

1. резьбовое соединение;
2. штифтовое соединение;
3. шпоночное соединение;
4. шлицевое (зубчатое) соединение;
5. байонетное соединение;
6. клеммовое соединение;
7. конусное соединение;
8. профильное соединение;
9. бугельное соединение;
- 10.соединение шип-проушина;
- 11.Защелки.

2. Неразъёмные соединения:

1. сварное соединение;
2. паяное соединение;
3. клееное соединение;
4. заклёпочное соединение (клёпаное соединение).

3. Условно разъёмное соединение:

1. соединение с натягом.

Многие варианты уже используются полностью или частично в медицине, в особенности в стоматологии, такие как пп 1.1, 1.2, 1.3, 1.6, 1.11, 2.3, 2.4.

При операциях на ребрах часто используют тип соединения 1.1 и его разновидности. Такой тип соединения для ребра не самый лучший, так как изначально в таком соединении при таких обстоятельствах появляются концентраторы напряжения, что не лучшим образом сказывается на лечении: высок риск того что соединение не выдержит и может разрушить часть костной ткани.

Самыми перспективными являются разновидности и модификации п1.6, п1.11, п2.3 и п3.1. Рассмотрим каждое отдельно

6.1.1 Клеммовое соединение

Клеммовое соединение, представленное на рисунке 6.1, изначально не предназначено чтобы соединять в торцах 2 трубы, но и в таком варианте тоже используется иногда, но само изделие, которое соединяет торами 2 трубы, используется очень широким, чтобы увеличить площадь контакта.

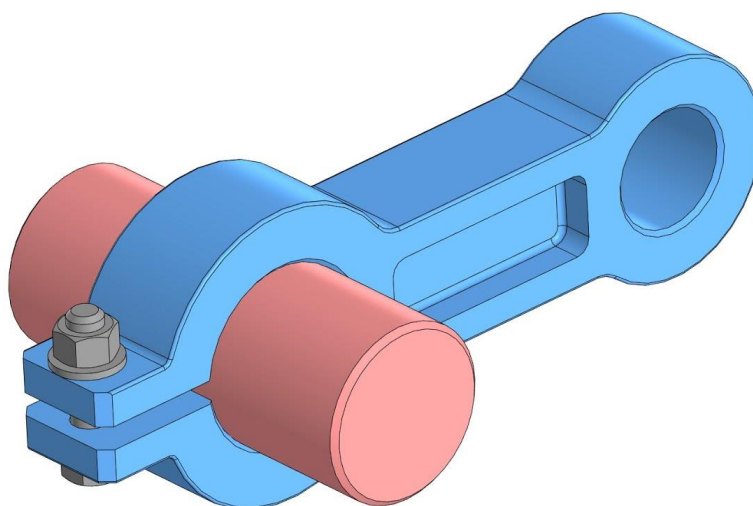


Рис. 6.1. – Клеммовое соединение

6.1.2. Защелки

Защелки, представленные на рисунке 6.2., используются чаще всего для изделий, сделанных из пластика, что ничуть не препятствует для решения данной задачи, так как основной материал будет использоваться пластик.

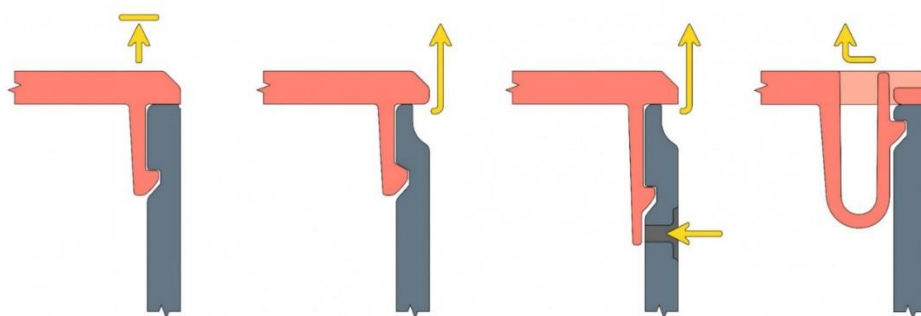


Рис. 6.2. – Защелки

Защелки очень сильно зависят от самой конструкции. Часто защёлки делают маленькими и узкими, от чего они не держаться долго, и легко ломаются.

6.1.3. Клеевое соединение

Клеевое соединение, представленное на рисунке 6.3., не является разъемным, что стоит учитывать, но для решения такой задачи такой тип соединения тоже подходит, так как изначально не предполагается что данное соединение будет извлекаться из тела человека.

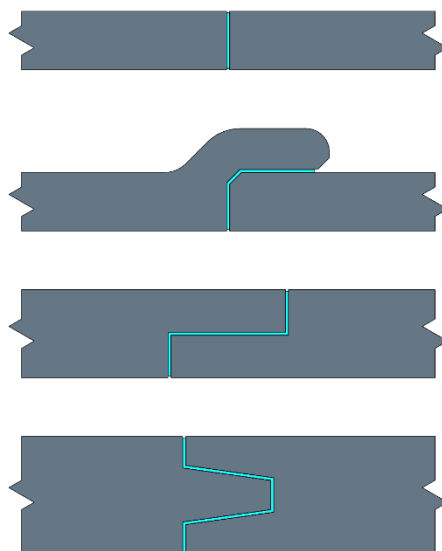


Рис. 6.3. – Клеевое соединение

Основное преимущество данного соединения является то, что оно не зависит от формы деталей в местах соединения. Также сами формы обязательно должны идеально совпадать в месте склейки. Что безусловно является преимуществом. Но есть и свои минусы.

Такое соединение очень сильно зависит от клеевого состава, от его свойств: как долго склеивает, за счет чего склеивает, токсичность, степень прочность закрепления.

Такое соединение сохнет не быстро, а долго, что уже является минусом, бывает токсичным, в особенности, когда это используется внутри человека, и часто является не крепким.

Хотя на данный момент и существует биологический клей[20], который удовлетворяет требованиям применения внутри человека, но не может быть крепким. Поэтому такое соединение используется как дополнение к основному соединению, что так же можно использовать.

6.1.4. Соединение с натягом

Соединение с натягом на пластиковых корпусах для электроники этот способ используется нечасто. Он требует достаточной точности изготовления, требователен к поверхностям и материалам, что может быть большой сложностью в реализации на практике во время операций на человеке.

7.2. Выводы по соединениям

Все рассмотренные подробно типы соединений можно использовать и проводить реальные опыты на живом существе за исключением соединения с натягом. К такому типу соединения надо подходить с осторожностью и прорабатывать технологию. Для данного соединения придется тратить много ресурсов, что не целесообразно, поэтому лучше направить основной упор на другие типы соединений.

ГЛАВА 7. РАЗРАБОТКА КРЕПЕЖА

Разработку модели крепежа будет происходить на основе всех ранее изложенных данных: с учетом тех результатов, которые были получены, как и в качестве экспериментальных данных, так и исследовательских результатов из ANSYS.

7.1. Этап 1. Получение меш модели ребра

С использованием ранее написанного ПО просмотрим снимки пациента. Из соблюдения конфиденциальности данные пациента не могут быть раскрыты.

На рисунке 7.1., представлен вид с программы: как читается сам снимок.

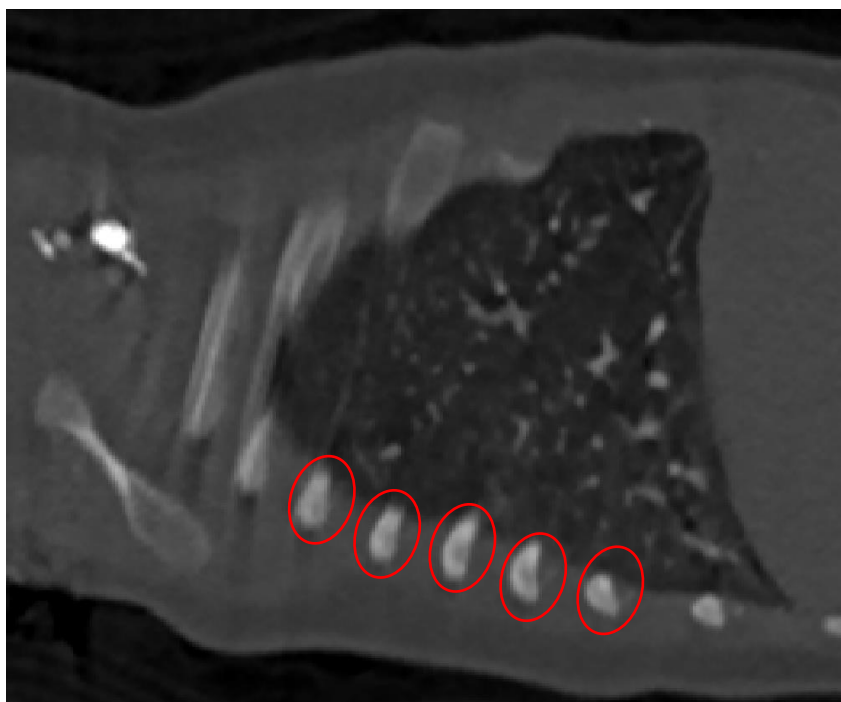


Рис. 7.1. – Снимок из программы

Красным выделены часть участков с ребрами. Их и необходимо будет выделить из снимка. Воспользуемся декомпозицией, которая была добавлена в данную программу. На рисунке 7.2. представлен этот же участок.

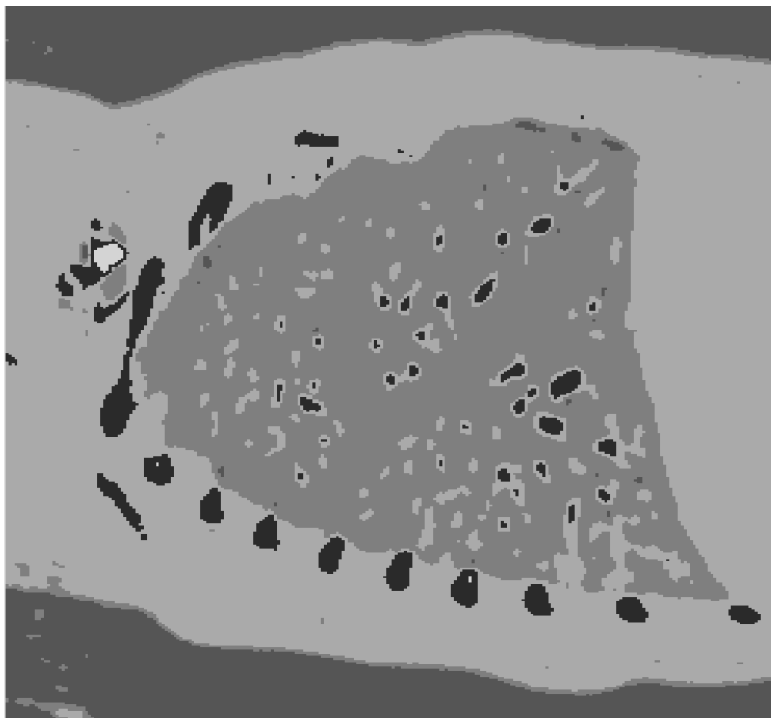


Рис. 7.2. – Снимок после декомпозиции

Можно обратить внимание, что ребра видны теперь четко, их границы имеют точное определение, даже сохраняется внутренняя полость кости, что соответствует реальности, так как сами кости похожи на трубы.

На рисунке 7.3. оставлено только 1 ребро, которое будет участвовать в исследовании.

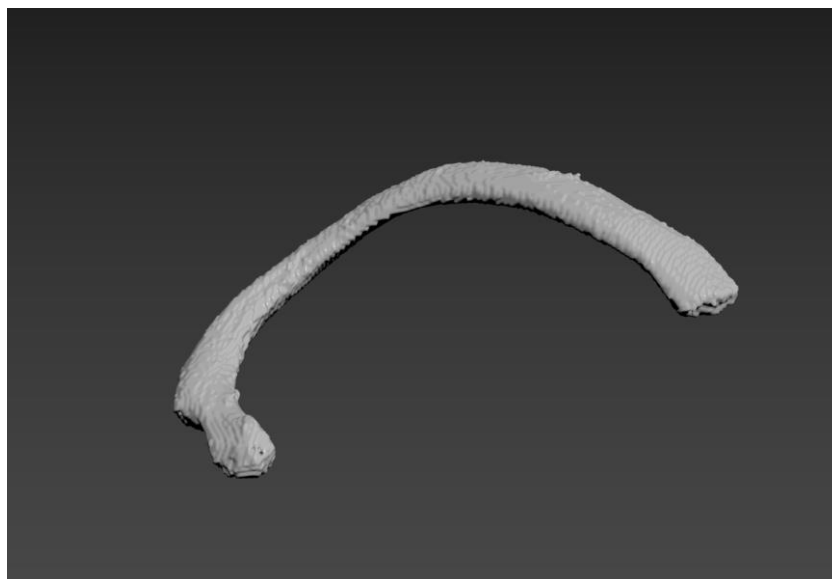


Рис. 7.3. – Выделенное ребро для исследования

Данное ребро необходимо подготовить для исследования на крепление.

7.2. Этап 2. Модель крепежа

Чтобы спроектировать крепеж, была использована программа 3ds Max. Использование именно этой программы обусловлено тем, что данная программа отлично работает именно с полигональными.

На рисунке 7.4. представлено подготовленное ребро с креплением с посадкой, а на рисунке 7.5. представлено ребро, подготовленное для классического соединения, то есть на болты[8][19].

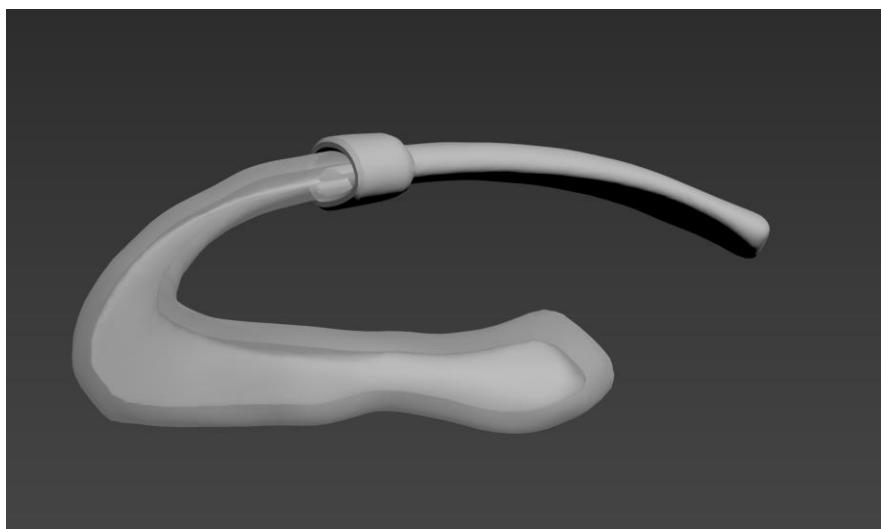


Рис. 7.4. – Модель крепления с посадкой

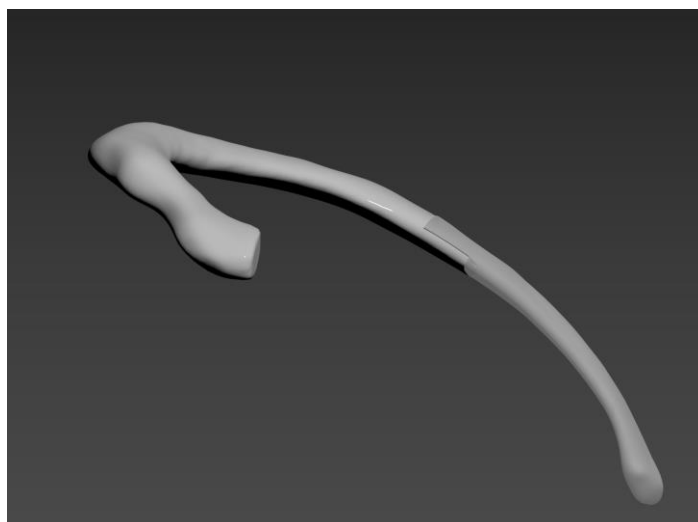


Рис. 7.5. – Модель с болтовым соединением

В новом крепеже предполагается, что есть 2 больших пятна контакта: снаружи кости и внутри кости. Также, новое ребро должно будет надеваться в натяг на кость. Общая длина контакта 10мм

7.3. Исследование

Чтобы определить какой тип соединениям будет лучше, все расчёты будут проводиться одинаково с одинаковыми граничными условиями в программе ANSYS.

7.3.1. Болтовое соединение

На рисунке 7.6. представлена подготовленная модель с болтовым соединением.

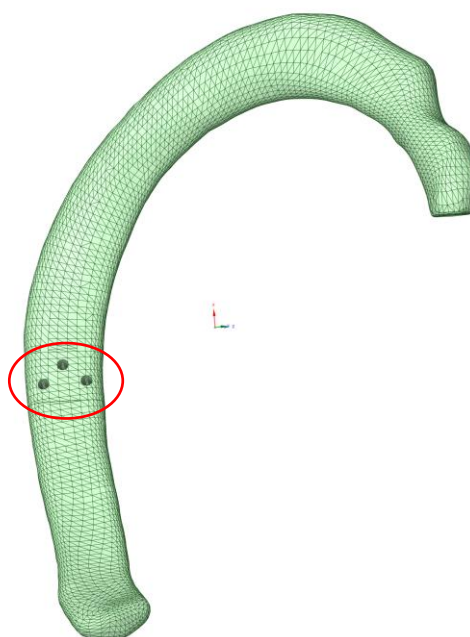


Рис. 7.6. – Подготовленная модель

На данной картинке красным овалом обведено место где были построены отверстия для крепления, а также в нем есть упрощение болтового соединения.

На рисунке 8.7. представлены граничные условия данной модели.

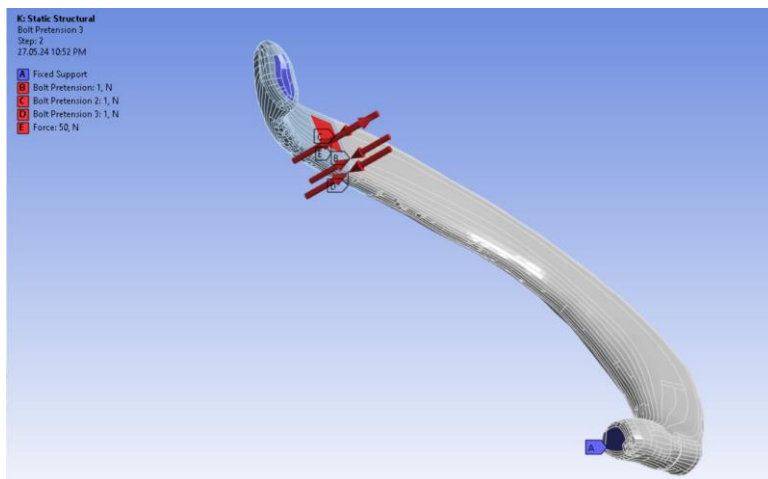


Рис. 7.7. – Граничные условия модели с болтовым соединением

Концы ребра жёстко закреплены, в местах болтового соединения установлена граница болтового соединения, рядом с местом соединения производится нагрузка в 5 кг.

На рисунке 7.8. проиллюстрирована зона контакта между костью и имплантом. Коэффициент трения взят 0,5 в справочнике как между пластиком гладким и камнем.

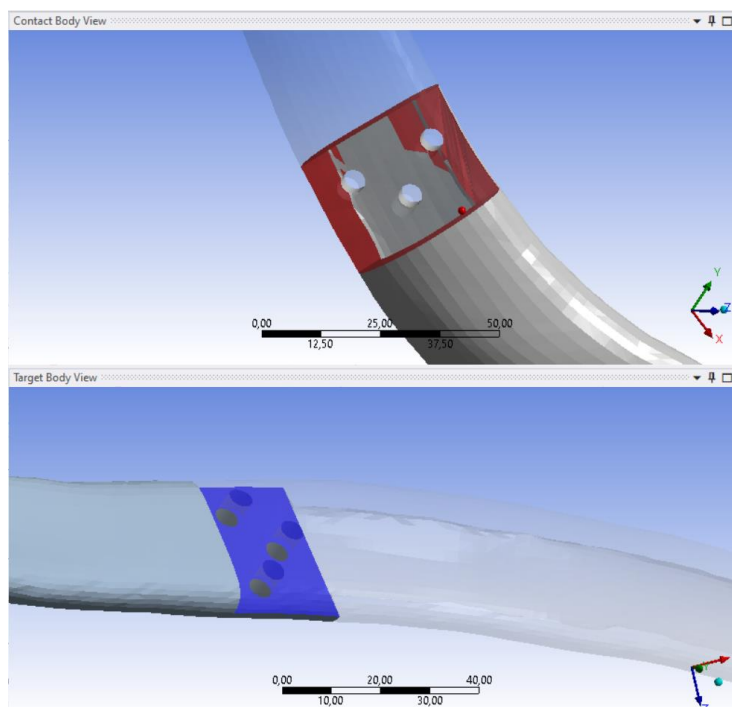


Рис. 7.8. – Зона контакта

В качестве материала был использован линейный материал кости из справочника. Расчетная модель с учетом больших деформаций.

7.3.2. Крепление с посадкой

Была подготовленная модель крепления с посадкой на клей и/или с натягом. На рисунке 7.9. представлены граничные условия.

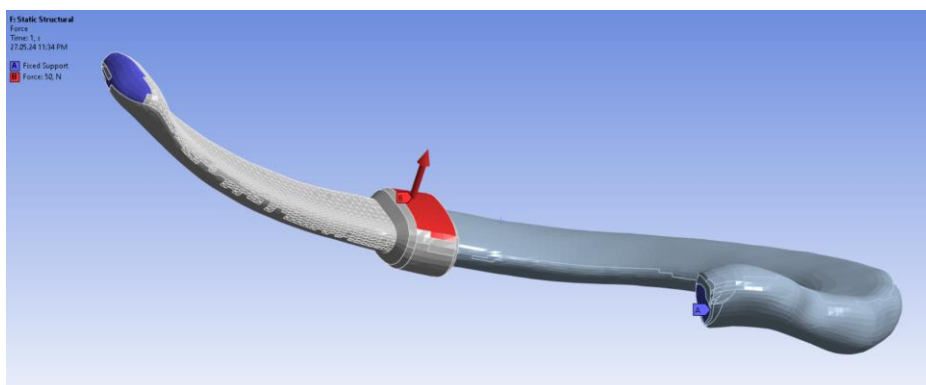


Рис. 7.9. – Граничные условия модели крепления с посадкой

Концы ребра жёстко закреплены, рядом с местом соединения производится нагрузка в 5 кг.

На рисунке 7.10. проиллюстрирована контактная зона, где и должен быть биологический клей соединяя кость с имплантом.

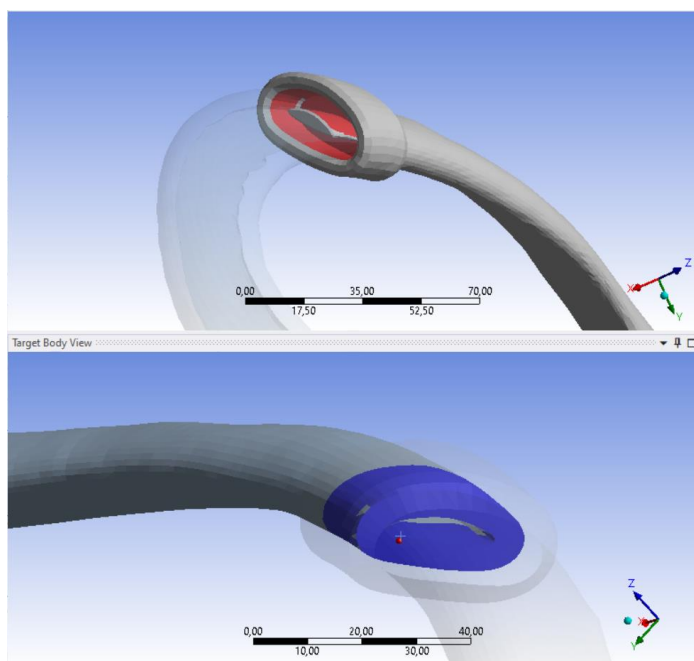


Рис. 7.10. – Зона контакта

Сразу можно обратить внимание на то, что контактная зона гораздо больше чем при предыдущем способе, но в предыдущем способе не применяется биологический клей вовсе.

В качестве материала был использован линейный материал кости из справочника. Расчетная модель с учетом больших деформаций.

7.4. Сравнение и анализ

Результаты исследования стандартного крепления представлены на рисунках 7.11. и 7.12.

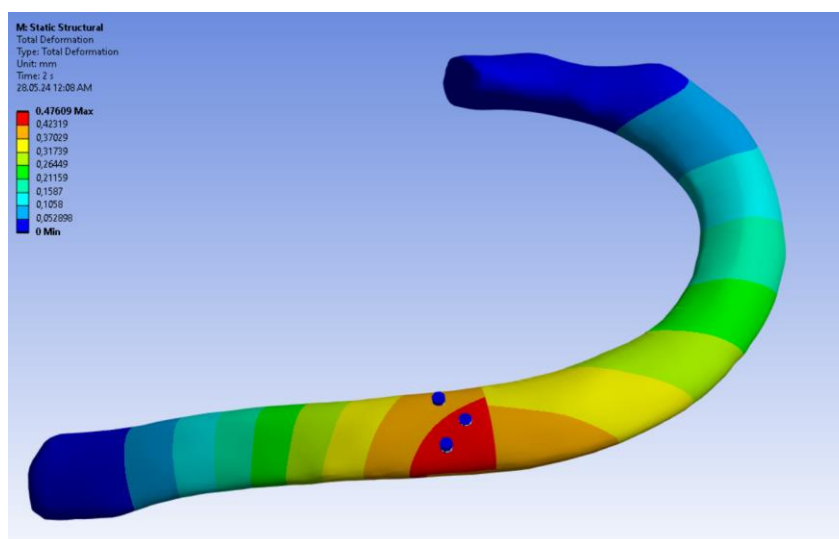


Рис. 7.11. – Деформации

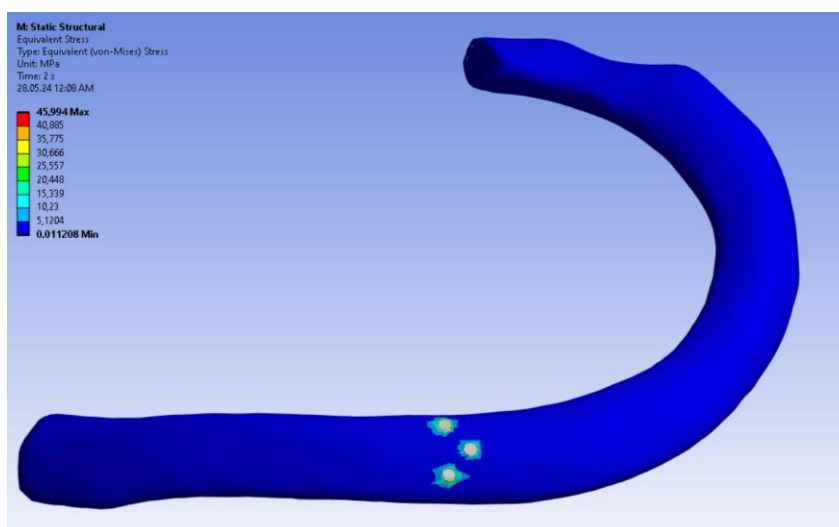


Рис. 7.12 – Напряжения

Самые максимальные напряжения как раз возникли в местах болтового соединения.

Результаты исследований крепления с посадкой представлены на рисунках 7.13. и 7.14.

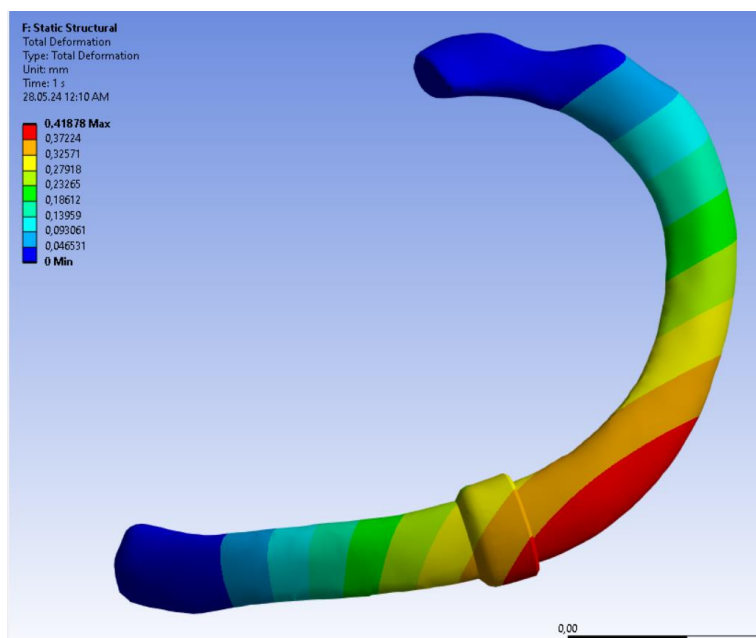


Рис. 7.13. – Деформации

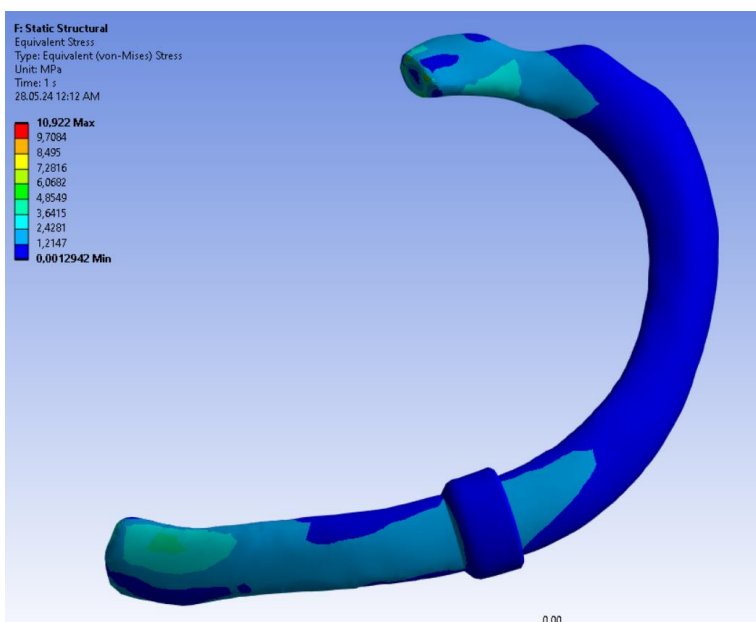


Рис. 7.14. – Напряжения

Численные результаты представлены в таблице 7.1.

Таблица 7.1. – результаты исследований

Тип крепления	КПЗ, мин	мм	МПа
Стандартный	0,8262	0,47609	45,994
С посадкой	3,4793	0,41878	10,922

7.5. Анализ результатов

При стандартном креплении самые максимальные напряжения и минимальный коэффициент запаса прочности находятся в болтовом соединении. В то же время как при соединении с посадкой максимальное напряжение и минимальное значение коэффициента запаса прочности сосредоточены только у зон с жестким закреплением.

Также стоит обратить внимание на численные результаты. При сравнительно одинаковой нагрузке максимальные напряжения гораздо меньше у крепления с посадкой, а максимальные деформации у этого же крепления немного меньше чем у стандартного крепления. Это все обусловлено не только наличием большего количества материала, но и более равномерно распределенной нагрузки между костью и имплантом.

ЗАКЛЮЧЕНИЕ

В процессе работы были решены следующие задачи:

1. С использованием инструментов машинного обучения был разработан метод первичной подготовки КТ для получения 3D моделей, пригодных для печати и компьютерных экспериментов. При этом увеличивается контрастность снимков, для более точного и качественного отделения костных тканей от других.
2. Исследованы пористые структуры на пропускную способность и упругие характеристики. По упругим характеристикам был определен диапазон плотности структуры, это помогает подобрать композитный материал для конкретных случаев. Определено, что тип структуры не влияет на физико-механические свойства. Однако кубическая сетка имеет разные свойства в разных направлениях.
3. Были рассмотрены типы крепления импланта с костной тканью (с ребром). Определены типы соединений, которые можно использовать: защелки, соединение в натяг и клееное. На основе всех этапов работы было разработано крепление, дающее коэффициент прочности больше, чем в болтовом соединении, в 4,2 раза при одинаковых условиях.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Купфер З. – Основы компьютерной томографии 29.05.2018 [Электронный ресурс], Medach, URL: <https://medach.pro/post/1348> (дата обращения: 15.12.2020).
2. Прокоп, Матиас. "Михаэль Галански." Спиральная многослойная компьютерная томография. 2-е издание, Методика исследования, страницы 29-31
3. Ту Дж., Гонсалес Р. – Принципы распознавания образов», страницы 109-112
4. anvos – Как выбрать оптимальный способ соединения деталей в пластиковых корпусах. 21 марта 2023 [Электронный ресурс], Хабр, URL: <https://habr.com/ru/articles/723684/> (дата обращения: 18.04.2024).
5. Bauermeister, A. J., Zuriarrain, A., Newman, M. I. – (2016). Three-dimensional printing in plastic and reconstructive surgery: A systematic review. *Annals of Plastic Surgery*, 77, 569–576.
6. Fan, B., Chen, H., Sun, Y. J., Wang, B. F., Che, L. et al. – (2017). Clinical effects of 3-D printing-assisted personalized reconstructive surgery for blowout orbital fractures. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 255(10), 2051–2057
7. Ghai, S., Sharma, Y., Jain, N., Satpathy, M., Pillai, A. K. – (2018). Use of 3-D printing technologies in craniomaxillofacial surgery: A review. *Oral and Maxillofacial Surgery*, 22(3), 249–259
8. Kang Jianfeng, Wang Ling, Chuncheng Yang, Lei Wang, Cao Yi, Jiankang He, Dichen L – Custom design and biomechanical analysis of 3D-printed PEEK rib prostheses, 2018
9. kornveits – 3D-печать, битва технологий, FDM vs SLA 19.07.2021 [Электронный ресурс], URL: <https://habr.com/ru/articles/568446/> (дата обращения: 28.01.2024).

10. Kosulin, A. V., Elyakin, D. V. – (2017). The use of a physical 3D model as a reference object in the surgical treatment of congenital kyphoscoliosis. *Pediatrician*, 8, M163–M164.
11. Kulczycki I.A., Loboda O.S., Suvorov V.V. – Algorithmic modelling of the heart based on CT scans, *Advanced Problem in Mechanics III*, p. 341-346
12. Li, C., Cheung, T. F., Fan, V. C., Sin, K. M., Wong, C. W. et al. – (2017). Applications of three-dimensional printing in surgery. *Surgical Innovation*, 24(1), 82–88
13. Martelli, N., Serrano, C., van den Brink, H., Pineau, J., Prognon, P. et al. – (2016). Advantages and disadvantages of 3-dimensional printing in surgery: A systematic review. *Surgery*, 159, 1485–1500
14. Seysen – FDM живее всех живых 26.12.2019 [Электронный ресурс], Хабр, URL: <https://habr.com/ru/articles/473390/> (дата обращения: 15.12.2020).
15. Shenaq, D. S., Matros, E. – (2018). Virtual planning and navigational technology in reconstructive surgery. *The Journal of Surgical Oncology*, 118(5), 845–852
16. Stürzer Dominik – FDM Vs. SLA 3D Printer: Choose The Right Technology 12.07.2021 [Электронный ресурс], bigrep, URL: <https://bigrep.com/posts/fdm-vs-sla-3d-printer/> (дата обращения: 13.02.2024).
17. Suvorov Vitaliy, Loboda Olga, Balakina Maria and Kulczycki Igor – A New Three-Dimensional (3D) Printing Prepress Algorithm for Simulation of Planned Surgery for Congenital Heart Disease p. 491-505"
18. Suzuki S, Abe K. – Topological Structural Analysis of Digitized Binary Images by Border Following, учебник, CVGIP. 1985, глава 1, страницы 32-46.
19. Wang Lei, Chuncheng Yang, Changning Sun, Xiaolong Yan, Jiankang He, Changquan Shi, Chaozong Liu, Dichen Li, Tao Jiang, Lijun Huang – Fused Deposition Modeling PEEK Implants for Personalized Surgical Application:

From Clinical Need to Biofabrication, International Journal of Bioprinting (2022)–Volume 8, Issue 4

20. Биологический клей [Электронный ресурс], MST, URL: [https://www.mst.ru/products/biomaterials/bioglue/#:~:text=BioGlue%C2%AE%20%2D%20%D0%B1%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9%20%D0%B4%D0%B2%D1%83%D1%85%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%BB%D0%B5%D0%B9,%D0%B4%D0%B2%D1%83%D0%BC%D1%8F%20%D1%85%D0%B8%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%20%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%BC%D0%B8%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC%D0%B8%20%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0%D0%BC%D0%B8\)](https://www.mst.ru/products/biomaterials/bioglue/#:~:text=BioGlue%C2%AE%20%2D%20%D0%B1%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9%20%D0%B4%D0%B2%D1%83%D1%85%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%BB%D0%B5%D0%B9,%D0%B4%D0%B2%D1%83%D0%BC%D1%8F%20%D1%85%D0%B8%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%20%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%BC%D0%B8%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC%D0%B8%20%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0%D0%BC%D0%B8)). (дата обращения: 18.04.2024).
21. Метод k-средних [Электронный ресурс], Wikipedia, URL: https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_k-%D1%81%D1%80%D0%B5%D0%B4%D0%BD%D0%B8%D1%85 (дата обращения: 15.12.2023).
22. Соединение [Электронный ресурс], Wikipedia, URL: <https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5> (дата обращения: 15.04.2024).
23. Технологии 3D печати. Разновидности [Электронный ресурс], 3DiY, URL: <https://3d-diy.ru/wiki/3d-printery/tekhnologii-3d-pechati-kakie-raznovidnosti/> (дата обращения 15.05.2024)

- 24.FDM технология. Как это работает. [Электронный ресурс], 3DTOOL, URL: <https://3dtool.ru/stati/fdm-tekhnologiya-kak-eto-rabotaet/> (Дата обращения 16.05.2024)
- 25.International standard to transmit, store, retrieve, print, process, and display medical imaging information [Электронный ресурс], Digital Imaging and Communications in Medicine, URL: <https://www.dicomstandard.org> (дата обращения: 18.12.2020).
- 26.NIfTI-1 format description [Электронный ресурс], Neuroimaging Informatics Technology Initiative, URL: <https://nifti.nimh.nih.gov> (дата обращения: 18.12.2020).
- 27.NumPy [Электронный ресурс], NumPy, URL: <https://numpy.org/> (дата обращения: 24.01.2021).
- 28.pybind11 documentation – [Электронный ресурс], pybind11, URL: <https://pybind11.readthedocs.io/en/stable/> (дата обращения: 15.12.2020).
- 29.PyDicom [Электронный ресурс], PYDICOM, URL: <http://pydicom.org/> (дата обращения: 24.01.2021).
- 30.PyQt5 [Электронный ресурс], Wikipedia, URL: <https://ru.wikipedia.org/wiki/PyQt> (дата обращения: 01.12.2023).
- 31.The Marching Cubes [Электронный ресурс], The Marching Cubes, URL: <http://users.polytech.unice.fr/~lingrand/MarchingCubes/algo.html>, (дата обращения 22.03.2021).
- 32.Ultimate Cura [электронный ресурс], <https://ultimaker.com/software/ultimaker-cura/> (Дата обращения 01.04.2024)
- 33.Wavefront .obj file [Электронный ресурс], Wikipedia, URL: https://en.wikipedia.org/wiki/Wavefront_.obj_file (дата обращения 18.04.2021).