

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки /специальность 09.04.04 – Программная инженерия
Направленность (профиль) образовательной программы Управление
разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой

_____ А.В. Бушманов
«_____» _____ 2023 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Моделирование напряженно-деформированных состояний аппаратов
внешней фиксации

Исполнитель

студент группы 157-ом

_____ А.Ф. Мышалов
(подпись, дата)

Руководитель

доцент, канд. техн. наук

_____ А.В. Бушманов
(подпись, дата)

Руководитель научного

содержания программы

магистратуры

профессор, доктор техн. наук

_____ И. Е. Еремин
(подпись, дата)

Нормоконтроль

Доцент, канд. техн. наук

_____ Л. В. Никифорова
(подпись, дата)

Рецензент

_____ Р.А. Кузьменко
(подпись, дата)

Благовещенск 2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
подпись
« _____ » _____ 2023 г.

ЗАДАНИЕ

К магистерской диссертации студента группы 157-ом Мышалова Александра Федоровича

Тема магистерской диссертации: Моделирование напряженно-деформированных состояний аппарата внешней фиксации

(Утверждено приказом от 21.02.2023 № 442-уч)

1. Срок сдачи студентом законченной работы (проекта) 20.06.2023 г.
 2. Исходные данные к выпускной квалификационной работе: предметная область, отчеты по практической подготовке, результаты выступления на научной конференции
 3. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): анализ предметной области проводимого исследования, разработка алгоритмической составляющей системы исследования напряженно-деформированных состояний аппаратов внешней фиксации, практическое исследование напряженно-деформированных состояний аппаратов внешней фиксации
 4. Дата выдачи задания: 30.01.2023 г.
 5. Руководитель выпускной квалификационной работы: Бушманов А.В., доцент, канд. техн. наук
(фамилия, имя, отчество, должность, уч. степень, уч. звание)
- Задание принял к исполнению (31.01.2023) _____

РЕФЕРАТ

Магистерская диссертация содержит 65 с., 19 рисунков, 2 приложения, 18 источников.

АППАРАТ ВНЕШНЕЙ ФИКСАЦИИ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, НАПРЯЖЕННО-ДЕФОРМИРОВАННОЕ СОСТОЯНИЕ, ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Цель данной работы – разработать программные средства, позволяющие за счет автоматизации уменьшить время расчетов напряженно-деформированного состояния аппаратов внешней фиксации при их непосредственном применении на теле человека.

Задачи выпускной работы:

- исследовать особенности применения аппаратов внешней фиксации для разнообразных задач и выделить главные параметры этих процессов;
- выделить нерешенные части задачи моделирования напряженно-деформированного состояния аппаратов внешней фиксации и разработать необходимые проектные решения по созданию соответствующего программного обеспечения;
- выполнить программную реализацию соответствующего обеспечения и провести его тестирование, оценить эффективность и сделать выводы о возможностях его применения в реальных практических исследованиях и на производстве.

СОДЕРЖАНИЕ

Введение	7
1 Анализ отрасли производства аппаратов внешней фиксации для тела человека	10
1.1 Предназначение аппаратов внешней фиксации и особенности конструкций	10
1.2 Особенности проектирования аппаратов внешней фиксации и обзор программных продуктов для данной предметной области	14
1.3 Выделение нерешенных ранее частей проблемы и постановка задачи на данное исследование	18
Выводы	20
2 Разработка алгоритмической составляющей системы исследования напряженно-деформированных состояний аппаратов внешней фиксации	22
2.1 Разработка требований к информации, обрабатываемой системой	22
2.2 Создание алгоритмов использования системы исследования напряженно-деформированных состояний аппаратов внешней фиксации	24
2.3 Обоснование выбора инструментальных средств для исследования напряженно-деформированных состояний аппаратов внешней фиксации	27
2.3.1 Средства для формирования входной информации для процесса моделирования	27
2.3.2 Программные продукты для моделирования напряженно-деформированных состояний аппаратов внешней фиксации	34
2.3.3 Инструменты для визуализации результатов моделирования	39
Выводы	45
3 Практическое исследование напряженно-деформированных	47

состояний аппаратов внешней фиксации	
3.1 Создание примера модели аппарата внешней фиксации	47
3.2 Моделирование напряженно-деформированного состояния спроектированного аппарата	48
3.3 Реализация средства визуализации результатов исследования	52
3.4 Оценка эффективности созданной программной системы исследования напряженно-деформированных состояний аппаратов внешней фиксации	54
Выводы	56
Заключение	57
Библиографический список	59
Приложение А	62
Приложение Б	63

СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

АВФ	– аппарат внешней фиксации
АРМ	– автоматизированное рабочее место
ИТ	– информационные технологии
ЛВС	– локальная вычислительная сеть
НДС	– напряженно-деформированное состояние
ПК	– персональный компьютер
ТЗ	– техническое задание
ЭВМ	– электронно-вычислительная машина
CPU	– Central Processor Unit
HDD	– Hard Disk Drive
LAN	– Local Area Network

ВВЕДЕНИЕ

Как известно, тело человека при экстремальных внешних воздействиях, к сожалению, подвержено разнообразным травмам, среди которых значительное место отводится переломам крупных костей конечностей: бедренной и берцовых костей, плечевой, локтевой и лучевой костей. Часто при переломе достаточно наложения гипса, однако, в сложных случаях, когда разрушенными могут быть целые участки костной ткани, для правильного и относительно быстрого срастания поврежденных костей больному устанавливают аппарат (аппараты) внешней фиксации (далее – АВФ). Так называют устройства, иммобилизирующие части костей в определенном пространственном взаимном расположении, которое является наиболее целесообразным на данный момент времени по мнению лечащего врача-ортопеда для задач остеосинтеза. Устройства должны допускать тонкую настройку, благодаря которой положение сращиваемых участков может постепенно меняться согласно указаниям врача. Форма этих устройств и их механические характеристики должны быть выверены и позволять четкую фиксацию частей тела в необходимых положениях с определенным запасом прочности. Следует отметить, что относительно недавно в научно-технической литературе (и гораздо раньше – в фантастической, художественной) стали описываться разработки т.н. усилителей физических способностей человека (или экзоскелетов), которые можно назвать ответвлением от устройств типа АВФ. И действительно, в них части тела человека фиксируются во вполне определенных приспособлениях (аналогично классическим АВФ), однако по своему предназначению эти устройства направлены на увеличение физической силы тела человека с помощью включенных в систему моторов. Анализируя приведенную информацию, можно сказать что во всех случаях, моделирование механических свойств АВФ, в частности их напряженно-деформированного состояния, представляет собой **актуальную задачу** для специалистов в области математического моделирования и ИТ.

Цель данной работы – разработать программные средства, позволяющие за счет автоматизации уменьшить время расчетов напряженно-деформированного состояния аппаратов внешней фиксации при их непосредственном применении на теле человека.

Задачи работы:

- исследовать особенности применения аппаратов внешней фиксации для разнообразных задач и выделить главные параметры этих процессов;
- выделить нерешенные части задачи моделирования напряженно-деформированного состояния аппаратов внешней фиксации и разработать необходимые проектные решения по созданию соответствующего программного обеспечения;
- выполнить программную реализацию соответствующего обеспечения и провести его тестирование, оценить эффективность и сделать выводы о возможностях его применения в реальных практических исследованиях и на производстве.

Объект исследования – процессы расчета напряженно-деформированного состояния аппаратов внешней фиксации.

Предмет исследования – методы и средства моделирования напряженно-деформированного состояния аппаратов внешней фиксации.

Элемент **научной новизны** работы состоит в предложенной методике исследования механических свойств аппарата внешней фиксации, представляющей собой полный цикл работ по автоматизации, начиная от создания модели, далее – самого моделирования и, наконец, представления результатов, что в целом позволяет экономить время на проведение соответствующих расчетов.

Практическое значение работы состоит в разработке примера замкнутого цикла исследования аппарата внешней фиксации с использованием трех программных продуктов: 3D редактора, среды численного моделирования и специально разработанной (в рамках данной работы) программы для визуализации результатов расчетов.

Методы, применяемые в работе: численные методы расчета напряженно-деформированных состояний, общенаучные методы анализа и синтеза, а также методы из области технологий программирования – для реализации соответствующего программного обеспечения.

1 АНАЛИЗ ОТРАСЛИ ПРОИЗВОДСТВА АППАРАТОВ ВНЕШНЕЙ ФИКСАЦИИ ДЛЯ ТЕЛА ЧЕЛОВЕКА

1.1 Предназначение аппаратов внешней фиксации и особенности конструкций

Несмотря на всё бурное развитие техники и технологий в разных сферах научного знания, наблюдающееся в последние десятилетия, человеческое тело по-прежнему остается относительно слабым (с точки зрения физической механики) и крайне уязвимым объектом.

В первую очередь, приведенный тезис, к сожалению, проявляется в разнообразных травмах, которые может получать человек в аварийных и различных нестандартных ситуациях. В данной работе речь будет идти о переломах костной системы тела человека (скелет), которая представляет собой каркас, основу для процессов механических взаимодействий с окружающей средой. Именно характеристики костей в первую очередь определяют механические возможности тела человека. Проводя аналогию с мехатронными системами, можно со всей очевидностью утверждать, например, что навешивание мощных моторов на непрочные консоли никогда не позволит создать мощные транспортирующие системы и т. п. Характеристики мышц (аналогичных двигателям робототехнических систем) являются вторичными, по отношению к костным параметрам. Кроме того, говоря о внезапных экстремальных воздействиях окружающей среды на тело человека, основную роль в таких процессах играет прочность костного остова, а влияние мышц здесь значительно ниже. Мало того, часто озвучивается мнение, что меньшим травмам способствует расслабленное состояние мышц во время ударов и падений, т.е. не мышечные усилия, а прочность костей является основным фактором, позволяющим избежать серьезных травм.

Во многих случаях при переломах человеку накладывают гипс, чего бывает вполне достаточно для восстановления костной ткани в месте разлома. Однако, при сложных переломах части кости должны быть зафиксированы

одна относительно другой на достаточно длительные периоды времени (порядка дня и более) – тут на пользу и приходят АВФ. Эти устройства могут иметь разные варианты крепления к телу человека, однако чаще всего используются временные спицы или даже штифты, прикрепляемые непосредственно к частям разломанной кости – рис. 1.1. Процесс этот является чрезвычайно болезненным и с медицинской точки зрения крайне опасным, так как в теле достаточно долго (порядка недель) присутствует фактически инородное тело и точки проникновения инфекций в организм. Конечно, использование спиц является значительно более предпочтительным по сравнению со штифтами, так как прямым образом вместе с диаметром уменьшается и степень опасности такого крепления для организма, однако слишком тонкие спицы могут стать механически неустойчивыми (подвергаться изгибам и кручениям) или вообще потерять общую прочность (переломиться).



Рис. 1.1 Примеры аппаратов внешней фиксации

Таким образом, налицо имеем задачу технической оптимизации: экстремально уменьшить размер вживляемых элементов (штифтов и спиц), но обеспечивая при этом общую и местную прочность конструкции АВФ. Следует отметить, что АВФ могут применяться не только для конечностей, а и для тазовой кости, позвоночника и других вариантов, однако, частота таких

применений в целом ниже, в сравнении с фиксацией длинных «линейных» костей. Классическим примером АВФ является аппарат Илизарова (схематически показан на рис. 1.2), который часто применяется не только для сращения сложных переломов, требующих длительного остеосинтеза, но и для вытягивания конечностей (процесс, при котором нагрузки на аппарат на порядок выше, чем в случае простого удержания отломков кости на заданных позициях). Именно такого типа АВФ будут рассматриваться далее в работе.



Рис. 1.2 Схема применения аппарата Илизарова, являющегося наиболее известным примером АВФ, для вытягивания целой кости (тонкие спицы вживляются в кость)

Переходя от сферы медицины к более утилитарным вопросам, связанным с АВФ, можно отметить, что даже физически развитый человек не может оперировать грузами, значительно превышающими его вес. Чемпионом в этом вопросе является муравей, поднимающий груз, пятидесятикратно превышающий собственную массу. Однако, и крупные млекопитающие, к тому же близкие к человеку, также достаточно эффективны в этом вопросе: горилла может нести вес, в 10 раз превышающий ее собственный. Средний же человек с

большим трудом справляется с весом, равным своему собственному. Тем не менее, на многочисленных работах, связанных с перемещениями тяжелых грузов, никто иной, кроме человека, по понятным причинам работать не может, поэтому целесообразен поиск путей увеличения силы человека, одним из которых является применение экзоскелетов – рис. 1.3, а (показана отечественная разработка Ростеха). Основное их предназначение – на погрузочных и транспортировочных работах, однако часто применение таким «костюмам» можно видеть и в военной сфере – рис. 1.3, б.

Экзоскелеты, в отличие от медицинских АВФ, фиксируются на теле без интрузионных элементов, а только лишь с учетом пространственной геометрии и локомоции тела человека. Конечно, на экзоскелетах имеются и компрессионные элементы, удерживающие всю конструкцию на теле, однако, такого типа крепления следует минимизировать, опираясь в большей степени на естественные упоры и различные ограничители. В любом случае, какой бы ни была конструкция экзоскелета, очевидно, что его элементы находятся в достаточно сложных механических условиях (так как это собственно и является целью их применения – перенос механических нагрузок на экзоскелет, вместо скелета человека), поэтому расчет их напряженно-деформированного состояния чрезвычайно важен и даже важнее (в экономическом смысле), чем для аппаратов, создаваемых для медицинской отрасли.



Рис. 1.3 Примеры экзоскелетов: *а* – российская разработка ProEXO для промышленных работ, находящаяся на этапе запуска в серийное производство; *б* – возможный вид экзоскелета военного назначения

1.2 Особенности проектирования аппаратов внешней фиксации и обзор программных продуктов для данной предметной области

Очевидно, проектирование АВФ, особенно медицинского типа, должно вестись междисциплинарными группами исследователей, в первую очередь, медиками и/или биологами, отлично понимающими особенности крепления конкретного аппарата к элементам тела человека. Однако, когда точки крепления определены, далее работать могут специалисты по механике, определяющие прочностные характеристики аппарата и его внешний вид в общих чертах. Далее следует привлечь специалистов по эргономическому и эстетическому (особенно, если решение планируется для широкого коммерческого использования) дизайну медицинской техники. Таким образом, будет установлен предпочтительный внешний вид АВФ, однако финальный расчет механического состояния все же должен завершаться работой

специалистов механиков, а конечное решение о завершении проектирования аппарата должны принимать проектанты от медицины.

Таким образом, можно сказать, что расчет напряженно-деформированного состояния является одной из основных стадий проектирования аппаратов внешней фиксации и должен выполняться специалистами по механике деформируемого твердого тела (прочнистами).

Наилучшие результаты по предварительным расчетам характеристик деформируемого тела дают методы математического моделирования, поэтому кратко рассмотрим инструменты, которые можно применить для таких целей.

В целом следует отметить, что спектр программного обеспечения, применяемого для задач моделирования аппаратов внешней фиксации, оказывается достаточно широк. Так, в работе [1] описывается моделирование напряжений при фиксации переломов дистального метаэпифиза плечевой кости, которое выполняется на базе программного продукта своей собственной разработки. Пример результатов работы программы показан на рис. 1.4.

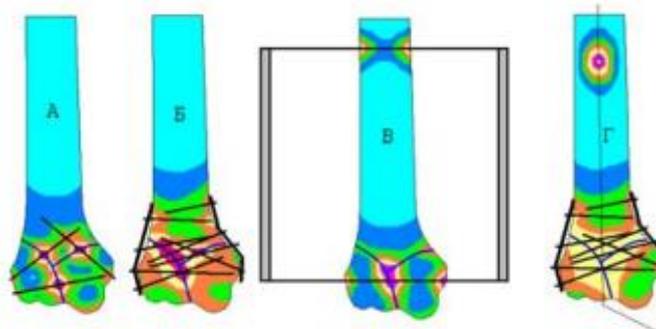


Рис. 1.4 Пример использования программного продукта собственной разработки для моделирования аппаратов внешней фиксации

Как видим, основным результатом тут является поле распределения напряжений, возникающее при фиксации перелома разными вариантами аппаратов. Математической основой разработанного ПО является метод конечных элементов (далее – МКЭ, англ. FEM – Finite Elements Method, FEA – Finite Elements Analysis). Аналогично и в [2], где производится моделирование АВФ для таза, использован МКЭ, а результатом должна являться картина

распределения напряжений. В качестве программы для расчетов здесь применяется MSC Nastran. Это отличная программа для мультидисциплинарного структурного анализа – как утверждают сами ее разработчики. Однако, стоимость продукта достаточно высокая (375 долларов США в месяц), а студенческие бесплатные версии в ней отсутствуют. Интерфейс продукта чрезвычайно насыщен разнообразными элементами, что говорит о богатой функциональности программы.

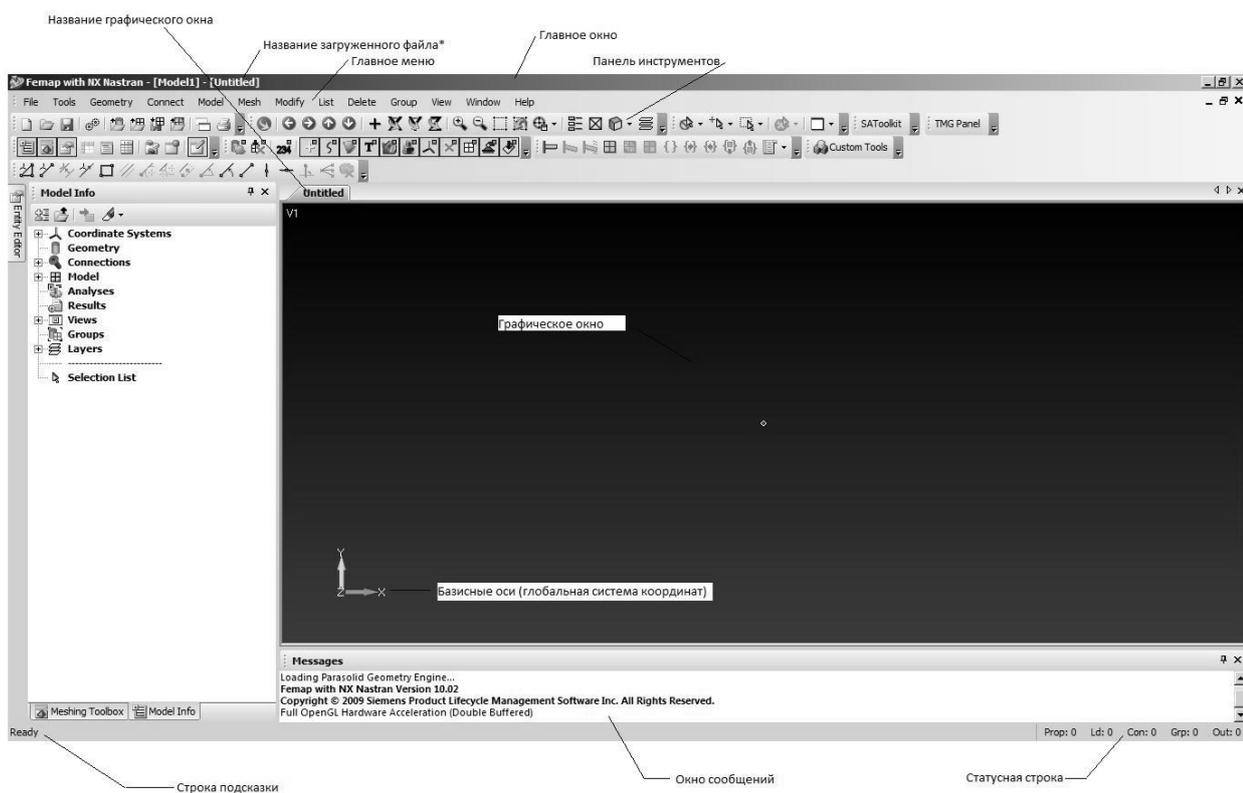


Рис. 1.5 Главное окно программы MSC Nastran для расчетов методом конечных элементов

Наконец, в [3] описано моделирование напряженно-деформированного состояния аппарата внешней фиксации для предплечья, выполненное на базе продукта ANSYS – рис. 1.6. Этот программный продукт является общепризнанным лидером в области расчетов с помощью МКЭ и будет принят за основу также и в рамках данной работы.

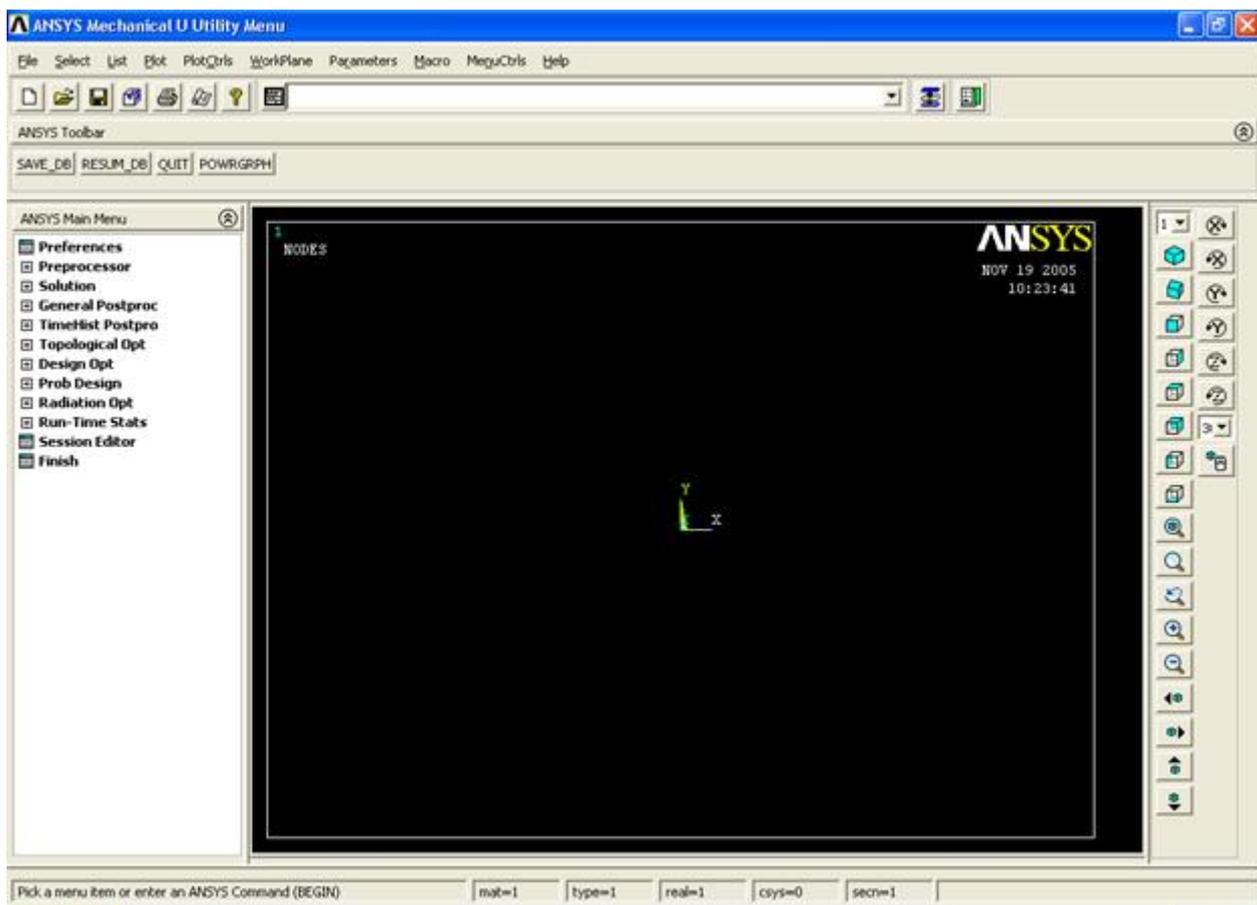


Рис. 1.6 Главное окно системы ANSYS для моделирования методом конечных элементов

Важной деталью, относящейся к данному программному продукту, является тип лицензии, согласно которому он распространяется: это платное программное обеспечение, причем его стоимость достаточно высока. Это, конечно, является серьезным недостатком и препятствует его использованию небольшими и даже средними конструкторскими компаниями, ведущими проектирование на полностью легальном программном обеспечении (например, ввиду плотного сотрудничества с западными партнерами).

1.3 Выделение нерешенных ранее частей проблемы и постановка задачи на данное исследование

Анализ научных публикаций показывает, что основная цель, которую обычно ставят себе авторы, состоит в анализе эффективности фиксации костных отломков с помощью АВФ и моделирование напряженно-

деформированного состояния именно костных тканей. При этом наличие достаточно высоких значений механических напряжений, показываемых моделью, позволяет уберечь пациента от дальнейшего разрушения кости, которое является возможным при наличии опасных значений механических нагрузок на и так травмированную костную ткань. Это направление чрезвычайно важно и обоснованно, однако, в работах практически отсутствует инженерный анализ самих АВФ, что на самом деле также является важной проблемой (хотя, конечно, и не такой, как недопущение дальнейшего травмирования пациента).

Действительно, как уже упоминалось ранее, спицы и штифты, вживляемые в костную ткань, представляют собой серьезную угрозу для инфицирования мягких тканей организма и места их введения в тело подлежат постоянной обработке и специальному уходу. Уменьшение размеров (в частности, диаметра) этих объектов уменьшает потенциальные зоны проникновения инфекций, поэтому вполне целесообразно с медицинской точки зрения. Однако, с механической точки зрения, уменьшение размеров ограничено соображениями прочности, поэтому моделирование самих АВФ (а не костей с ними связанных) и является также важным процессом, требующим решения, что и выполняется в данной работе.

При рассмотрении второго аспекта АВФ, связанного с усилением силы человека с помощью экзоскелетов, необходимость расчетов напряженно-деформированного состояния их составляющих проявляется еще значительно, т.к. в данном случае АВФ будет сам нести значительную внешнюю нагрузку.

Таким образом, необходимо разработать методику исследования напряженно-деформированного состояния аппаратов внешней фиксации, направленную на исследование физических величин, характеризующих сам аппарат и определяющих его прочностные характеристики.

Для этого следует выделить части такой методики, и предложить пути решения каждой из возникающих подзадач, а также указать обеспечение, с помощью которого это решение может быть достигнуто. Для подтверждения

реализуемости методики необходимо рассмотреть пример ее применения на конкретном АВФ.

Выводы

Таким образом, в разделе рассмотрена предметная область разработки аппаратов внешней фиксации. Рассмотрено их традиционное медицинское применение и рассмотрено такое расширение данного класса устройств, как экзоскелеты, позволяющие увеличить силу человека. Проанализирован спектр программного обеспечения, применяемого для целей моделирования и установлено, что наиболее лучшим вариантом является применение продукта ANSYS, наиболее распространенного для задач, решаемых методом конечных элементов. В то же самое время результаты расчетов, вероятно потребуются анализировать и на других ЭВМ, где данный, достаточно сложный продукт не будет установлен. Так же само и подготовку трехмерной модели АВФ целесообразнее вынести на отдельную ЭВМ, где наличие программы ANSYS также не должно быть обязательным. Эти требования обуславливают использование дополнительного ПО: трехмерных редакторов – для создания моделей АВФ, и средств отображения информации о результатах моделирования их напряженно-деформированного состояния (возможно, собственной разработки). Эти соображения свидетельствуют о том, что целесообразной является проработка отдельной методики исследования АВФ на предмет их напряженно-деформированного состояния, учитывающей указанные особенности. Разработка и анализ такой методики будет выполняться в дальнейшей работе.

2 РАЗРАБОТКА АЛГОРИТМИЧЕСКОЙ СОСТАВЛЯЮЩЕЙ СИСТЕМЫ ИССЛЕДОВАНИЯ НАПРЯЖЕННО-ДЕФОРМИРОВАННЫХ СОСТОЯНИЙ АППАРАТОВ ВНЕШНЕЙ ФИКСАЦИИ

2.1 Разработка требований к информации, обрабатываемой системой

Итак, согласно материалам предыдущего раздела, можно предложить трехуровневую схему обработки информации при проектировании АВФ – рис. 2.1.

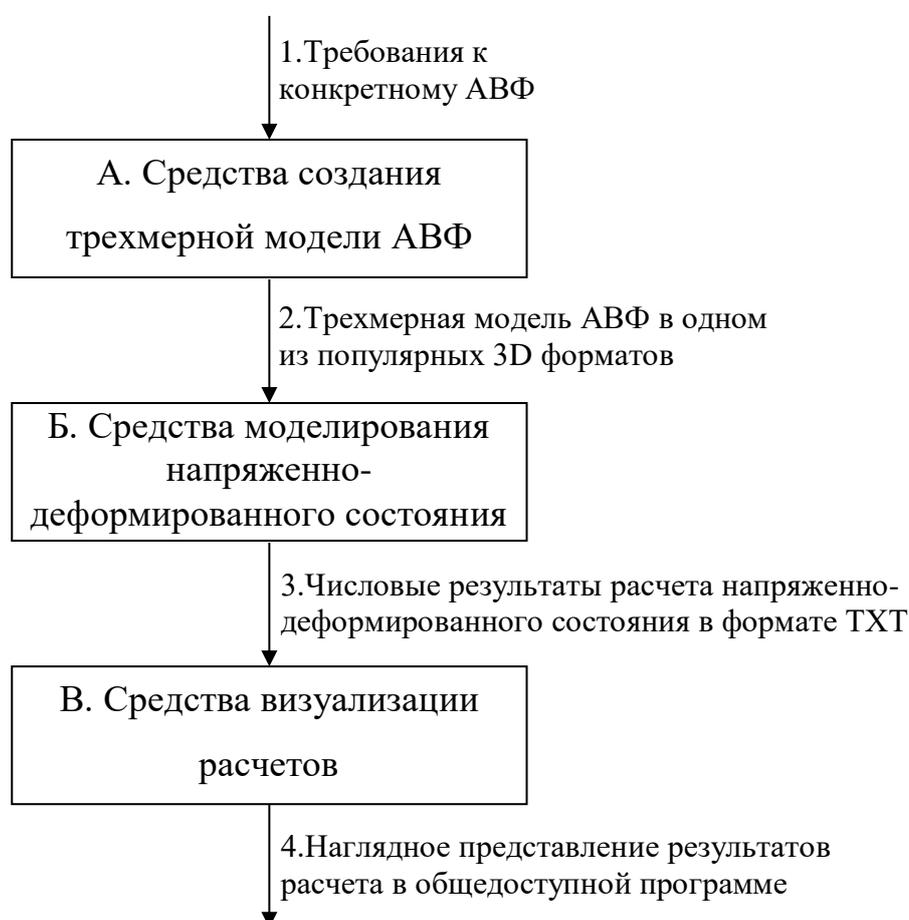


Рис. 2.1 Схема обработки информации при предложенном методе проектирования АВФ

На схеме возле каждого информационного потока приписана суть информации, которая его составляет:

- на вход подаются изначальные требования, предъявляемые к конструкции конкретного типа АВФ, подлежащего проектированию. Согласно этим данным инженер (возможно, с функциями дизайнера) формирует его трехмерную модель. При этом используются программные средства для создания таких моделей. Это может быть практически любой популярный и свободный к использованию трехмерный редактор, например, Blender. Выходной формат может быть одним из свободных к использованию вариантов: STL (расшифровывается как Standard Triangle Language или Standard Tessellation Language или STereoLithography) или OBJ;

- на вход программного средства, моделирующего напряженно-деформированное состояние (и вероятнее всего другие параметры АВФ, хотя бы общий вес и/или расход материала), подается созданный на предыдущем шаге трехмерный файл. Если в качестве средства моделирования используется именно ANSYS (как говорилось об этом выше), то он может импортировать файлы формата STL. Выходной на этом этапе является информация о результатах расчета НДС, которую для универсальности можно сохранять в обычном текстовом формате (т.е. например, экспортировать значения вектора перерезывающей силы и/или изгибающего момента в обыкновенном текстовом формате TXT). Данный этап выполняется инженером-прочнистом;

- числовые результаты расчета НДС, продуцированные на предыдущем этапе, могут быть переданы третьему инженеру, например, осуществляющему общий надзор и управление процессом проектирования (в частном случае все этапы выполняет одно и то же лицо). Для визуализации этих результатов используются средства, не имеющие глубокой специализации (как, например, пакет ANSYS), и поэтому распространяющиеся свободно. Целесообразным вариантом здесь является разработка программного продукта собственного производства, предназначенным строго для визуализации НДС в аппаратах внешней фиксации. Выходом этой программы являются графики, представляющие результаты расчетов в наглядной визуализированной форме.

Учитывая специфику информации о механическом состоянии объекта, это должны быть традиционные линейные графики или столбиковые диаграммы.

Таким образом, подробно описаны требования к информации, передающейся между составляющими разрабатываемой системы на разных этапах ее использования.

2.2 Создание алгоритмов использования системы исследования напряженно-деформированных состояний аппаратов внешней фиксации

Разработка любого ПО должна включать в себя один из таких обязательных (начальных) этапов, как разработка алгоритмов работы программы. Как известно, алгоритмы могут описываться в виде:

- подробного текстового описания;
- наглядного представления в виде блок-схемы;
- таблиц значений выходных переменных программы в зависимости от значений входных переменных, подающихся на ее вход, и т.д.

В данном случае разработке подлежит трехуровневая схема, в которой можно выделить общий алгоритм ее использования и алгоритмы работы с каждой из ее составляющих. Эти частные алгоритмы определяются особенностями того программного продукта, который используется для указанной стадии (например, Blender, ANSYS и т.д.).

Общее представление процесса работы по предлагаемой схеме показано на рис. 2.2 в виде диаграммы языка UML.

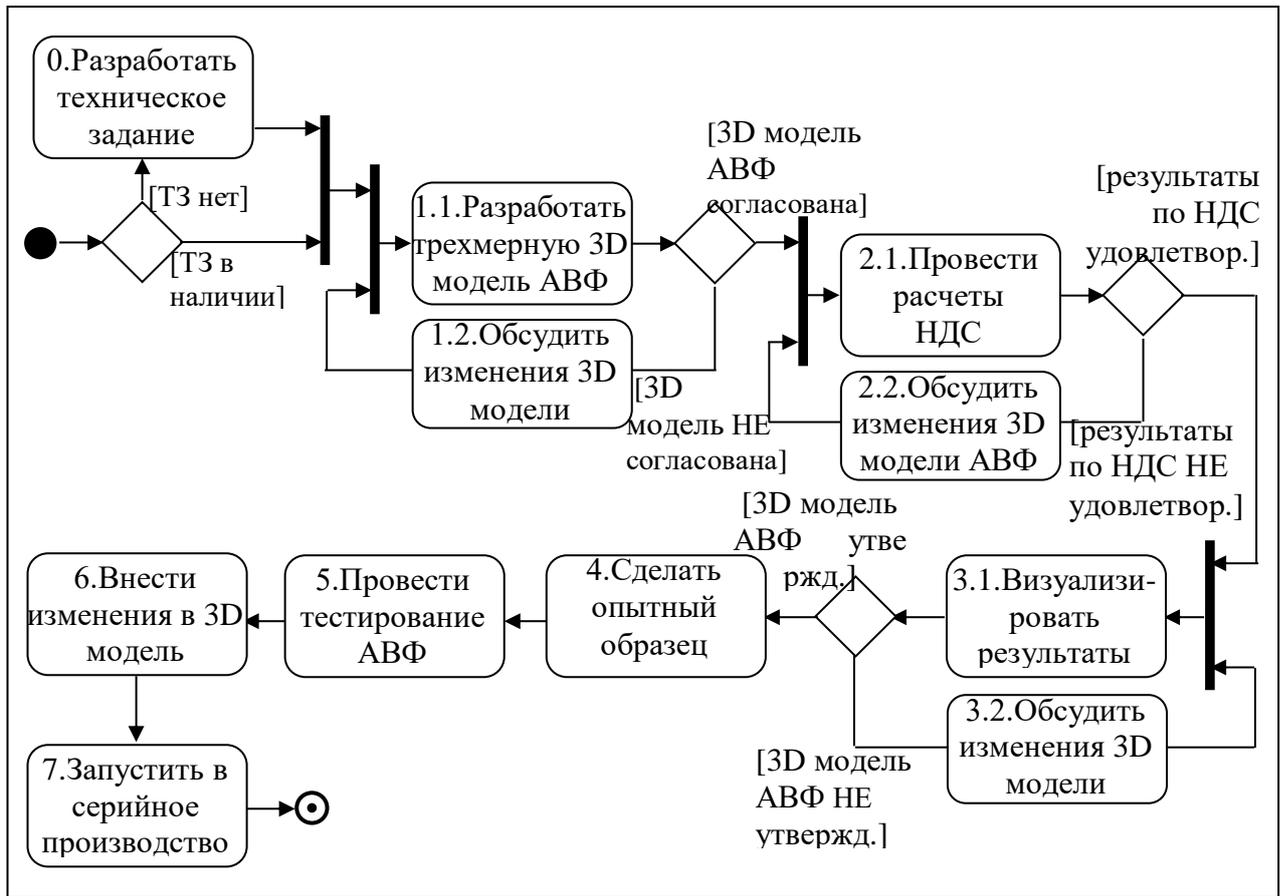


Рис. 2.2 Схема алгоритма процесса проектирования АВФ по предлагаемой в данной работе трехэтапной схеме

В данной работе программное обеспечение первых двух этапов процесса не разрабатывается, а только создается ПО для визуализации результатов расчетов НДС. Это ПО имеет простую пакетную схему работы – рис. 2.3.

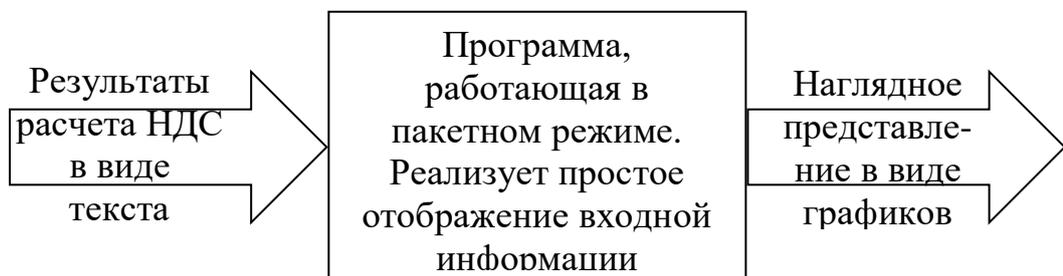


Рис. 2.3 Пакетная схема работы визуализирующей результаты расчета НДС программы

Следует отметить, что программы, имеющие пакетные режимы работы могут строиться по различным принципам и, соответственно, быть

ориентированными на различные операционные системы. Так, простейшим примером программы с пакетным режимом работы является консольное приложение, работающее в текстовом режиме интерфейса пользователя. Такое приложение может быть сделано полностью кроссплатформенным, то есть один и тот же код можно оттранслировать под разные целевые платформы (для этого, конечно, в коде не должны использоваться специфические возможности, присущие только одной какой-нибудь платформе, например, любые API-функции ОС Windows, которые очевидно отсутствуют в ОС семейства *nix). При использовании текстового режима интерфейса пакетной программы все необходимые для ее работы сведения она может получать через аргументы командной строки (в частности, простые числовые настройки, короткие ключи или опции запуска, а также названия файлов с обширными входными данными).

Организация работы пакетных программ через аргументы командной строки позволяет также легко переходить к использованию графического интерфейса пользователя. При этом не требуется повторная реализация логики работы приложения в рамках новой программы с графическим интерфейсом, так как вся суть использования этой графической программы состоит в сборе необходимых сведений, передаваемых в дальнейшем как аргументы командной строки в консольную программу. Такая программа, предназначенная только для сбора входных данных в визуальном режиме, называется фронтендом. Подобная организация программ (консольное приложение, содержащее всю логику обработки данных и получающее все необходимые сведения в виде аргументов командной строки, а также программа с графическим интерфейсом для сбора этих аргументов - фронтенд), в частности, широко распространена для ОС Linux.

Используя приведенные в данном подразделе схемы, можно переходить к вопросам выбора инструментальных средств для различных этапов предложенного процесса.

2.3 Обоснование выбора инструментальных средств для исследования напряженно-деформированных состояний аппаратов внешней фиксации

2.3.1 Средства для формирования входной информации для процесса моделирования

Для создания непосредственно трехмерной модели может использоваться любая программа для твердотельного моделирования. Если разработчик обладает лицензией на соответствующий программный продукт, это может быть один из проприетарных мощных пакетов наподобие SolidWorks или AutoCad, а также 3D Studio Max или Maya. Если такой лицензии нет, можно ограничиться использованием одного из бесплатных продуктов, например, Blender.

Элементарнейшим, но популярным способом использования трехмерного содержимого является его создание в специализированных программах (типа 3D Studio Max, Maya, Blender 3D) с последующим рендерингом, то есть созданием обычного статического изображения (например, в формате JPG) или динамической видеозаписи (например, в обычном формате AVI). При этом сами исходные трехмерные сцены хранятся во внутреннем формате соответствующей программы (например, 3ds – 3D Studio Max) и их не может читать пользователь, а тот результат, который ему передается (JPG или AVI), допускает просмотр только с того ракурса, который указал при рендеринге автор модели, а главное, не позволяет организовать никакого интерактивного взаимодействия с человеком. Очевидно, что такой подход не подходит для целей данной работы, поскольку является крайне ограниченным и больше предназначен для создания анимаций для YouTube (TikTok, Likee и других сервисов по обмену видеозаписями собственного производства) или красивых статических обоев для рабочего стола, но не для промышленного применения.

Для целей создания трехмерных объектов разработан, например, формат VRML, который рассмотрим подробнее.

Данный формат предназначен как промышленной, так и для развлекательной (непроизводственной) сферы. Об этом также говорит его расшифровка, которая переводится как «Язык моделирования виртуальной реальности» (Virtual Reality Modeling Language), ведь на сегодняшний день вся виртуальная реальность, которая уже реализована в ограниченных масштабах, направлена как на развлекательную, так и на промышленную, научную сферу.

Технология VRML [4] представляет собой открытый стандарт виртуальной реальности в сети Internet. Виртуальная реальность - это созданное компьютером среда («мир», «окружение»), которое доступно пользователю и может включать в себя трехмерную графику, объемный звук и тому подобное. VRML можно использовать для создания миров, программирования компьютерных игр и объемного представления информации. VRML является открытым стандартом, поэтому каждый пользователь может использовать язык VRML для написания программ и создания собственного виртуального мира, не делая при этом никаких лицензионных отчислений. При использовании VRML можно создавать виртуальный мир, части которого будут размещены по всему миру, например, стул на экране монитора находится во Франции, сад, что его окружает, находится в Японии, записанная фонограмма - в Англии и т.д.

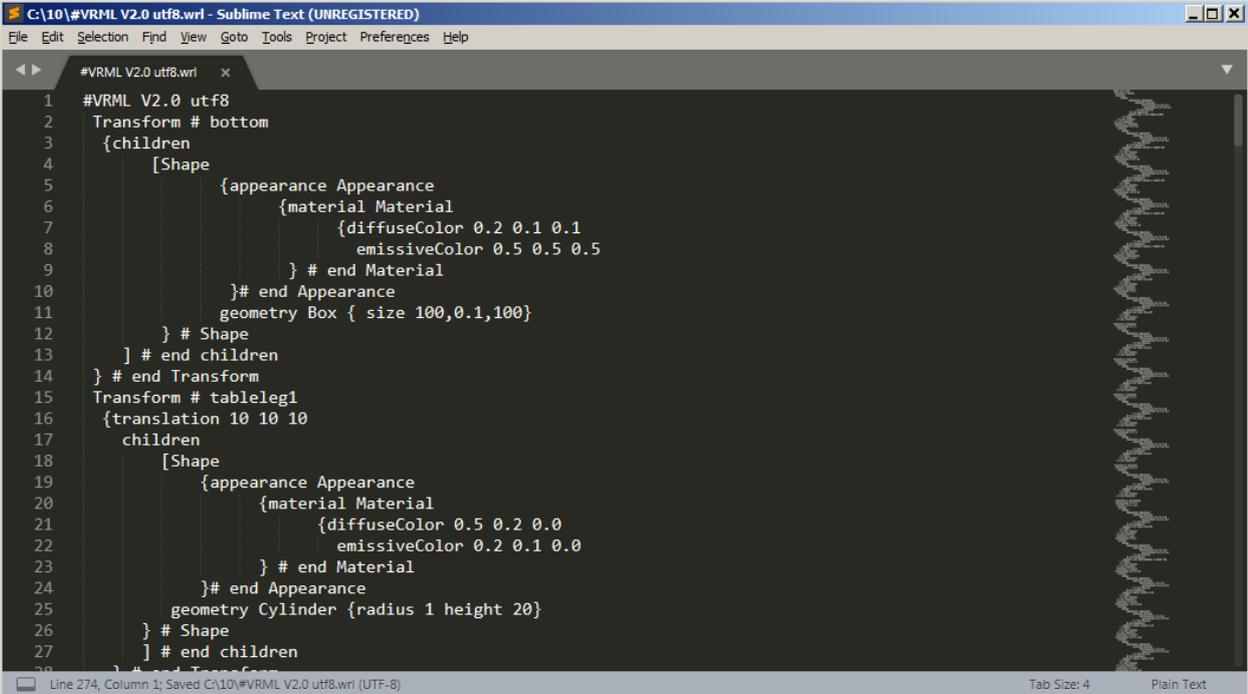
Существует много приложений использования VRML, которые используют открытый трехмерный формат файлов VRML, сетевые возможности языка моделирования виртуальной реальности и мультимедийную структуру. Приведем некоторые сферы использования VRML, заявляемые его разработчиками:

- автоматизированное проектирование;
- моделирование;
- программирование компьютерных игр;
- визуализация данных;
- распределенный многопользовательское среду;
- специальные программы;
- интерфейс пользователя;

- финансовые приложения;
- маркетинг и реклама продукции;
- образование;
- развлечения.

Файлы формата VRML имеют расширение *.WRL (от англ. «WoRLd», т.е. «мир») и являются текстовыми документами, т.е. сам язык относится к классу интерпретируемых [5]. Пример кода на этом языке, загруженный в популярный редактор Sublime Text Editor, приведен на рис. 2.4.

Из анализа рисунка можно сделать вывод, что во-первых, язык VRML является простым текстовым, что характерно практически для всех 3D форматов. Во вторых, «программа», или лучше сказать трехмерная разметка имеет достаточно простую и понятную структуру, в результате чего писать код VRML можно уже через несколько десятков минут после начала изучения языка. Также в правой части окна редактора видно, что весь текст имеет сильно иерархичную структуру, чем напоминает разметку на языке HTML (XML).



```
#VRML V2.0 utf8
1 #VRML V2.0 utf8
2 Transform # bottom
3 {children
4   [Shape
5     {appearance Appearance
6       {material Material
7         {diffuseColor 0.2 0.1 0.1
8           emissiveColor 0.5 0.5 0.5
9         } # end Material
10      } # end Appearance
11     geometry Box { size 100,0.1,100}
12   } # Shape
13 ] # end children
14 } # end Transform
15 Transform # tableleg1
16 {translation 10 10 10
17   children
18   [Shape
19     {appearance Appearance
20       {material Material
21         {diffuseColor 0.5 0.2 0.0
22           emissiveColor 0.2 0.1 0.0
23         } # end Material
24       } # end Appearance
25     geometry Cylinder {radius 1 height 20}
26   } # Shape
27 ] # end children
28 }
```

Рис. 2.4 Структура программы на языке VRML

Для просмотра кода VRML в браузере пользователь должен предварительно установить себе дополнительный плагин, например, такой как Cortona 3DViewer или использовать одну из программ твердотельного моделирования, как например Maya.

На этом языке даже достаточно простой код, написанный непрофессионалом, позволяет строить относительно полезные 3D объекты, а в случае работы опытного промышленного дизайнера, язык VRML вполне может быть использован как профессиональное средство разработки моделей деталей и устройств.

Основными примитивами традиционно считаются параллелепипед, шар, в меньшей степени цилиндр, конус, и т.д. Коротко рассмотрим процесс создания первых двух из этих объектов.

Параллелепипед создается с помощью кода:

```
#VRML V2.0 utf8
DEF Box001 Transform {
  translation 0 0 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.8353 0.6039 0.898
        }
      }
      geometry Box { size 2 3 2 }
    }
  ]
}
```

Результат приведен на рис. 2.5.

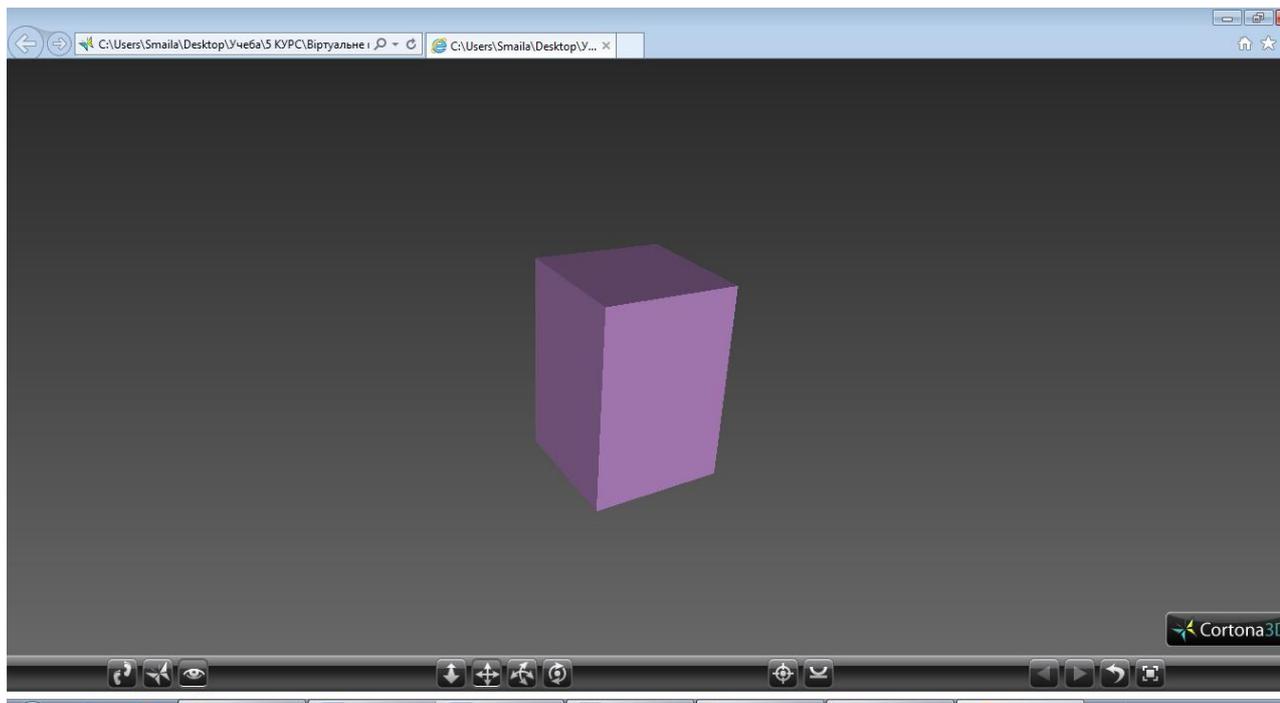


Рис. 2.5 Трехмерное изображение параллелипипеда

Следующим шагом является создание сферы. Пишем код:

```
#VRML V2.0 utf8
DEF Sphere001 Transform {
  translation 0 0 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.5804 0.6941 0.1059
        }
      }
      geometry Sphere { radius 4 }
    }
  ]
}
```

Полученный результат на рис. 2.6.

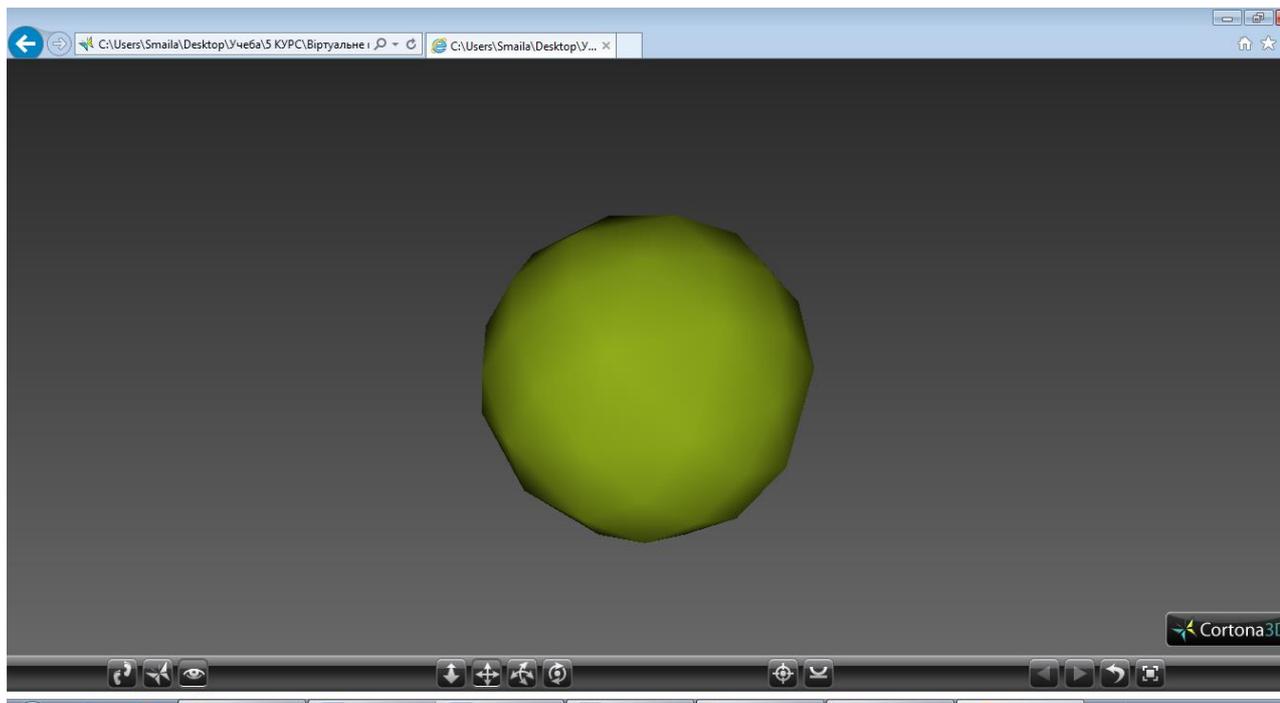


Рис. 2.6. Трехмерное изображение сферы

Следующим шагом является создание цилиндра. Пишем код:

```
#VRML V2.0 utf8
DEF Cylinder001 Transform {
translation 0 0 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 0.8392 0.8941 0.6
}
}
geometry Cylinder { radius 1 height 3 }
}
]
}
```

Получаем результат, показанный на рис. 2.7.

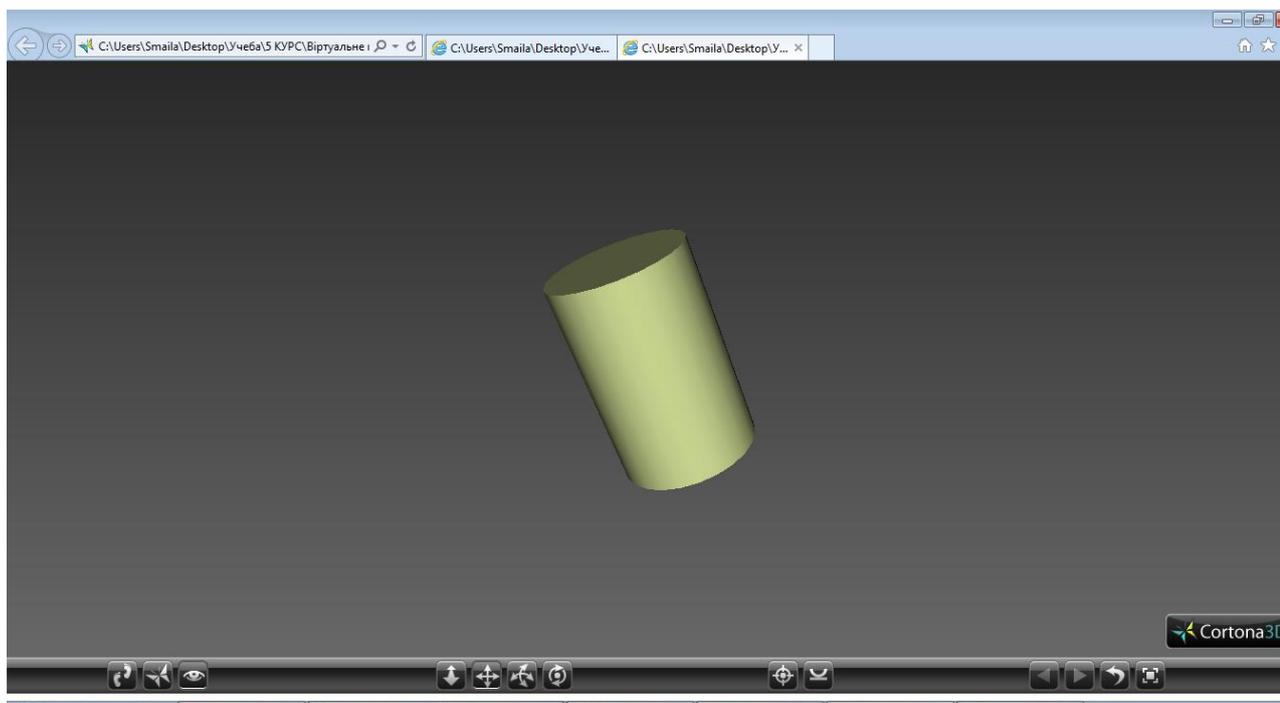


Рис. 2.7 Трехмерное изображение цилиндра

Таким образом, видим, что язык VRML представляет собой простое и эффективное средство, удобное для применения при создании трехмерных объектов, в том числе и вариантов АВФ. Анализ свойств данного языка и других средств представления трехмерных объектов позволяет сказать, что его свойства являются в достаточной степени типовыми, характерными для практически всех способов представления трехмерных моделей (из-за чего данный формат и был рассмотрен так подробно).

2.3.2 Программные продукты для моделирования напряженно-деформированных состояний аппаратов внешней фиксации

В первую очередь, следует отметить, что аппараты внешней фиксации характеризуются значительной простотой конструкции (несколько кольцевых элементов и штифтов либо спиц, крепящихся к ним), поэтому по характеристикам их напряженно-деформированного состояния не будут сложнее других механических устройств, т.е. не имеют какой-либо особой специфики. Наоборот, если в других устройствах к конструкции могут прилагаться значительные усилия, то в данном случае они используются для

фиксации костей, поэтому не испытывают чрезмерных нагрузок. В основном нагрузки, действующие на них находятся на уровне весовых параметров тела человека (т.е. на АВФ действуют веса и моменты от частей тела человека, которые поэтому не могут быть чрезмерно высокими). Таким образом, условия эксплуатации АВФ даже проще, чем для каких-либо схожих по структуре балок, ферм и элементов в прочностных конструкциях.

С другой стороны, процедура одевания (вживления) АВФ является крайне болезненной, поэтому недопустима ситуация, когда вживленный АВФ механически выходит из строя ввиду нарушения местной или общей прочности (в результате чего его следует снимать и одевать другой, новый). Поэтому к АВФ предъявляются требования повышенного уровня надежности (если обычно инженерами реализуется двух- или трехкратный запас по прочности, то в данном случае целесообразно применять пятикратный и более запас).

Еще одним специфическим вопросом при проектировании АВФ является ограничение на применение для их изготовления исключительно гипоаллергенных материалов. При этом они должны обеспечивать достаточный уровень механической прочности и не быть хрупкими. Также важным параметром является плотность материала, из которого изготавливают АВФ. Обычно, чем более плотный материал, тем более прочным он является, однако, в последние годы область материаловедения ступила далеко вперед и, в частности, в области композиционных материалов. Такие материалы могут иметь достаточную прочность, но быть значительно легче, чем сплошное металлическое изделие. Данное направление является чрезвычайно популярным, ввиду широкого внедрения новых материалов (особенно композиционных – на базе различных комбинаций металлов с неметаллическими включениями и т.п.).

Все вышесказанное свидетельствует о том, что для моделирования НДС АВФ вполне можно применять те же программные продукты, которые используются для моделирования НДС в самых общих случаях. Такими продуктами являются, например, ANSYS, Nastran, FEM models, Mathcad,

Matlab, GenIDE32, SolidWorks, Stress, Plaxis, вычислительный комплекс SCAD и др.

Некоторые из перечисленных продуктов в большей степени приспособлены для расчетов именно НДС, некоторые в меньшей. Например, пакет Mathcad, представляя собой универсальную среду для инженерных расчетов, тем не менее вообще не имеет встроенных средств для моделирования непосредственно НДС. Тем не менее, он часто применяется для этих целей, однако необходимые расчетные формулы при этом задаются самим исследователем вручную. Несколько более высокий уровень предоставляет пакет Matlab, имеющий многочисленные расширения и средство Simulink, позволяющее более просто моделировать различные режимы работы нагруженных конструкций, по сравнению с MathCad, где все нужно прописывать вручную.

Некоторые средства расчетов НДС имеются и в CAD системе твердотельного моделирования SolidWorks. Аналогичные элементы имеются и в отечественной системе КОМПАС и (в несколько меньшей степени) в американской системе AutoCAD. Однако, если необходим полноценный инженерный анализ при различных вариантах нагружения конструкции, причем этот анализ нужно производить не единично, а на постоянной основе для разных конструкций, то следует переходить к использованию систем Nastran или ANSYS, которые обладают достаточно продвинутыми возможностями по расчету НДС.

По заявлению разработчиков, Nastran – это лучшая на рынке конечно-элементная система. Постоянно развиваясь уже в течение 40 лет, она аккумулирует в себе достоинства новейших технологий, методов, алгоритмов и поэтому остается ведущей системой конечно-элементного анализа в мире.

MSC Nastran обеспечивает полный набор расчетов, включая расчет напряженно-деформированного состояния, запасов прочности, собственных частот и форм колебаний, анализ устойчивости, исследование установившихся и неустойчивых динамических процессов, решение задач

теплопередачи, акустических явлений, нелинейных статических и нелинейных переходных процессов, анализ сложного контактного взаимодействия, расчет критических частот и вибраций роторных машин, анализ частотных характеристик при воздействии случайных нагрузок и импульсного широкополосного воздействия, исследование аэроупругости на дозвуковых и сверхзвуковых скоростях. Предусмотрена возможность моделирования практически всех типов материалов, включая композитные и гиперупругие. В состав расширенных функций входит технология суперэлементов (подконструкций), включая продвинутые методы динамических конденсаций, модальный синтез и развитые методы анализа динамики сложных структур на основе суперэлементов и формулировок метода Крейга-Бемптона.

MSC Nastran располагает эффективным аппаратом автоматической оптимизации параметров, формы и топологии конструкций. Широкие возможности оптимизации позволяют использовать MSC Nastran для автоматической идентификации компьютерной расчетной модели и планирования экспериментов.

В составе MSC Nastran имеются специальные возможности моделирования динамики роторных машин, что обуславливает незаменимость этого программного продукта в отраслях, связанных с разработкой турбомашин.

MSC Nastran предоставляет возможности расчёта характеристик работы конструкций из композиционных материалов.

MSC Nastran широко используется для планирования экспериментов (определение мест расположения датчиков) и оценки полноты полученных экспериментальных данных. С помощью MSC Nastran решаются задачи моделирования систем управления, систем терморегулирования с учетом воздействия этих систем на конструкцию.

Специальная опция MSC Nastran SOL 600 - по сути «встроенный» решатель Marc, обеспечивающий проведение углубленного анализа

существенно нелинейного поведения конструкций и решения задач теплопередачи, включая связанные теплопрочностные задачи.

MSC Nastran - это современная расчетная суперсистема. В то же время, тесная интеграция MSC Nastran через Patran и SimManager с другими системами высокого уровня: Adams, Fatigue, FlightLoads and Dynamics, Marc, Mvision, Dytran, Easy5, SimDesigner и Sofy, а также интеграция со всеми известными системами CAD/CAM/CAE - реализует совершенно новый по своей широте и глубине системный уровень моделирования и многодисциплинарного анализа. Практически, на компьютере создается точная виртуальная модель изделия и, еще до начала производства, всесторонне исследуется его функционирование в рабочих и экстремальных условиях, тем самым, совершенствуя изделие, повышая его качество, надежность, безопасность, технологичность и экономичность на основе "компьютерных испытаний". Этот совершенно новый уровень интегрированной среды наукоемких инженерных компьютерных систем реализует современные VPD технологии.

Системой, которая конкурирует с Nastran и чаще применяется для научных расчетов, но также и достаточно часто – для инженерных, является ANSYS.

На сегодняшний день возможности Ansys Mechanical включают следующие наиболее основные пункты:

- Статические расчеты.

Определение прочности конструкции — самый распространенный тип расчетов в рамках механики деформируемого твердого тела, который проводится инженерами всех отраслей. Ansys Mechanical обеспечивает возможность вычисления напряженно-деформированного состояния (НДС) конструкции любой сложности.

- Динамические расчеты.

Резонанс является одним из опаснейших явлений в технике. Возможность узнать состав собственных частот конструкции позволит избежать появления резонанса, обеспечив высокую надежность конструкции и длительный срок

службы. Для решения данной задачи в Ansys Mechanical реализована возможность проведения модального анализа — анализа собственных частот и форм колебаний конструкций.

- Тепловые расчеты.

Как правило, основной целью задач теплообмена является определение полей температур и тепловых потоков как в отдельных деталях, так и целых агрегатах или конструкциях. В Ansys Mechanical возможно смоделировать 3 вида теплообмена: теплопроводность изотропного или ортотропного материала, конвективный теплообмен и теплообмен излучением с окружающей средой и/или между поверхностями тел.

- Акустические расчеты.

Ansys Mechanical позволяет выполнять акустические расчеты с целью определения собственных частот акустических объемов и отклика среды на акустическое возбуждение. Одним из распространенных применений акустических расчетов является эхолокация — определение акустического поля рассеяния объектов различной формы.

Как видим возможности данной системы включают не только механику, но и другие отрасли.

Учитывая все позитивные стороны и широкое распространение программы ANSYS, выбираем ее в качестве основы для дальнейшей разработки.

2.3.3 Инструменты для визуализации результатов моделирования

Как упоминалось ранее, средства визуализации целесообразно разрабатывать собственными силами в виде отдельного программного продукта. Логичнее всего использовать один из свободных языков программирования, допускающих разработку бесплатных программных продуктов для коммерческого использования.

Итак, необходимо установить, какой из многочисленных имеющихся на рынке или в свободном доступе языков программирования предоставляет достаточные возможности для реализации проектируемой программы,

связанной с визуализацией результатов расчета напряженно-деформированного состояния АВФ, в программных кодах. Следует отметить, что разные средства разработки или даже целые языки программирования с самого начала своего создания имели разное назначение и некоторые из них являются специализированными, то есть приспособленными к выполнению только определенных задач отдельных прикладных отраслей, а некоторые – средствами общего назначения. Среди последних можно выделить содержащие отдельные уже готовые, более или менее проработанные компоненты для решения любой прикладной задачи (в частности, такой, как отображение разнообразных графиков), а также такие языки, в которых таких компонентов нет.

Также важным фактором при выборе средств разработки выступает его популярность (в частности, языка программирования), т.е. распространенность фактов применения этого средства в профессиональной деятельности. Не следует недооценивать этот момент, поскольку в процессе разработки нового программного обеспечения более-менее сложной функциональности у любого программиста будут возникать вопросы по ее реализации, причем как раз с помощью выбранного средства разработки. В настоящее время самым первым и эффективным местом, где можно получить информацию о вопросах разработки программного обеспечения является глобальная сеть Интернет. Сегодня программисты сравнительно легко и быстро находят ответы на практически любые вопросы, связанные с профессиональной деятельностью в Интернет (среднее время поиска составляет порядка нескольких минут, максимум – до 10). Такая, практически идеальная, ситуация реализуется только при условии текущей популярности используемого языка программирования (или среды разработки, библиотеки, фреймворка, и т.п.). Например, существуют десятки (если не сотни) тысяч ресурсов, посвященных языку программирования PHP, причем некоторые из них имеют как подробную теоретическую базу в виде пакетов документации, информационных статей, видеолекций и т.д., так и средства для «живого» общения программистов

между собой – начиная от форумов и заканчивая онлайн-чатами. В то же время, учитывая, что всего в мире существует уже 500-2000 (по данным разных источников) более или менее активно используемых языков программирования, очевидно то, что некоторые из них используются чрезвычайно редко и найти необходимую информацию о них в Интернет иногда не представляется возможным или чрезвычайно трудно.

Отметим, что на сегодняшний день достаточно популярны такие языки программирования общего назначения как: C++, Java, C#.

Первый вариант, язык C++, является чрезвычайно мощным, имеющим стабильную популярность выше среднего уровня, но и достаточно сложным для восприятия; фактически, это инструмент для профессионалов высочайшего уровня. В то же время на сегодняшний день в среднем более популярны языки Java и C#.

Таким образом, при реальном процессе выбора языка программирования для профессиональной разработки программистом среднего уровня на сегодняшний день следует рассматривать два основных варианта – Java и C#. Очень много полемики было посвящено сравнению этих двух языков, но безусловным фактом является значительное преимущество продукта от Microsoft над конкурентом в части возведения пользовательского интерфейса. В этом вопросе среда Visual Studio предоставляет практически тот же уровень удобства, что и Delphi, но использует популярные языки, а не Паскаль, к которому традиционно укоренилось отношение как к учебному языку программирования для новичков. Итак, ввиду необходимости создания программного средства для операционной системы Windows, выбираем язык программирования C#, что и соответствует задаче разработки в рамках данной работы. Кратко рассмотрим особенности этого языка программирования.

C# (произносится как «си шарп») - простой, современный объектно-ориентированный и типобезопасный язык программирования. C# относится к широко известному семейству C-подобных языков, и будет казаться хорошо знакомым каждому, кто работал с C, C++, Java или JavaScript.

C# является объектно-ориентированным языком, но поддерживает также компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в форме автономных пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов – это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, которые дают декларативные сведения о нем, а также встроенные элементы документации. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы, и благодаря этому язык отлично подходит для создания и применения программных компонентов.

Приведем несколько функций языка C#, обеспечивающих надежность и устойчивость программ, разработанных с помощью нее:

- уборка мусора автоматически освобождает память, занятую уничтоженными и неиспользуемыми объектами;
- обработка исключений предоставляет структурированный и расширяемый способ выявлять и обрабатывать ошибки;
- строгая типизация языка не позволяет обращаться к неинициализированным переменным, выходить за пределы индексированных массивов или выполнять неконтролируемое приведение типов.

В C# существует единая система типов: все типы, включая типы-примитивы, такие как `int` и `double`, наследуются от одного корневого типа `object`. Таким образом, все типы используют общий набор операций, и значения любого типа можно хранить, передавать и обрабатывать подобным образом. Кроме того, C# поддерживает определенные пользователем ссылочные типы и типы значений, позволяя как динамически выделять память для объектов, так и сохранять упрощенные структуры в стеке.

Чтобы обеспечить совместимость программ и библиотек C# при дальнейшем развитии, при разработке C# большое внимание было уделено управлению версиями. Многие языки программирования обходят вниманием

этот вопрос, и в результате программы на этих языках ломаются чаще, чем хотелось бы, особенно при выходе новых версий зависимых библиотек. Вопросы управления версиями оказали существенное влияние на такие аспекты разработки C#, как:

- отдельные модификаторы `virtual` и `override`;
- правила разрешения перегрузки методов;
- поддержка очевидного объявления членов интерфейса.

Файлы исходного кода C# обычно имеют расширение `.cs`.

Команда `csc` на самом деле и является компилятором языка C# (`csc.exe`), другими словами выполняет компиляцию на полной версии платформы, и она может существовать не для всех программно-аппаратных систем (изначально предназначалась для программирования для ОС Windows).

Программирование на языке C# довольно простое и удобное. В целом, можно констатировать, что язык программирования C# на сегодняшний день является наилучшим для программирования современных приложений общего назначения, поэтому именно он выбран в качестве основного средства для реализации проектируемой программы для визуализации результатов расчета напряженно-деформированного состояния аппаратов внешней фиксации.

Определившись с языком программирования, следует провести выбор оставшихся средств разработки (конкретный компилятор и компоновщик, текстовый редактор, система помощи, наладчик и т.д.). Лучше всего использовать для этого некую интегрированную среду разработки (Integrated Development Environment, или IDE).

Учитывая, что выбранный язык программирования C# появился уже около 20 лет назад (в 2002 году), для него за это время появились разнообразные альтернативные компиляторы и среды разработки, потребность в которых сильно уменьшилась, когда Microsoft начала на постоянной основе (а не эпизодически, под какую-нибудь временную акцию) издавать бесплатные версии собственного продукта Microsoft Visual Studio.

Этот инструмент, безусловно, является самой мощной платформой для программирования приложений, предлагаемой корпорацией Microsoft, которая уже более 20 лет развивает свой продукт Microsoft Visual Studio (далее – MVS). Следует отметить, что основой среды MVS является .NET Framework – программная платформа, выпущенная корпорацией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), подходящая для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду, однако основным языком с самого начала здесь позиционировался именно C#. Считается, что платформа .NET Framework стала ответом компании Microsoft на платформу Java компании Sun Microsystems (ныне принадлежит Oracle), набравшей в то время очень большую популярность. Действительно, подробный анализ показывает, что Microsoft использует очень схожие по своей сути технологии, но собственной разработки, и, конечно, с другими названиями.

Большим преимуществом продукта от Microsoft стало на порядок большее удобство использования среды MVS, по сравнению с продуктами для Java-разработки. К примеру, среда Eclipse требует многочисленных настроек для каждого более или менее специфического проекта. Очевидно, что при наличии опыта использования в несколько лет все эти действия выполняются за несколько секунд, однако сегодня часто встречается ситуация (особенно в небольших компаниях), когда один программист в разные периоды использует разные инструменты, причем значительное их количество. Соответственно, не только для новичков актуален вопрос как можно меньшего количества настроек, инсталляций дополнительного программного обеспечения (библиотек, модулей, пакетов и т.п.), низкого порога вхождения – по всем этим показателям Microsoft Visual Studio значительно опережает своих конкурентов, в том числе продукты для разработки на платформе Java.

И еще одним немаловажным фактором, также действующим в плюс продукта от Microsoft, являются требования к скорости работы в данной среде

разработки при заданном объеме оперативной памяти и характеристиках процессора. MVS, например, выполняет такие базовые вещи, как компиляция проекта в разы быстрее (и за вполне приемлемое время порядка секунд или десятков секунд для проектов средней сложности), по сравнению с действительно удобными средами разработки для платформы Java типа IntelliJ IDEA (компиляция проекта в которой может занимать время порядка минут и даже их десятков).

Существенным преимуществом Java первоначально была открытость средств разработки, но на сегодняшний день существуют полностью свободные для использования (даже при создании коммерческих программ) версии MVS, поэтому соответственно в этом показателе две данные платформы равны. Также хотя .NET является патентованной технологией корпорации Microsoft и официально рассчитана на работу под операционными системами семейства Microsoft Windows, существуют независимые проекты (прежде всего Mono и Portable.NET), позволяющие запускать программы .NET на некоторых других операционных системах. В настоящее время .NET Framework получила развитие в виде .NET Core, которая изначально предполагает кроссплатформенную разработку и эксплуатацию. Таким образом, и по показателю кроссплатформенности две этих платформ стали примерно равными в последние годы.

Учитывая все особенности (удобства) среды MVS, выбираем ее за основу для дальнейшей разработки проектируемой программы визуализации напряженно-деформированного состояния аппаратов внешней фиксации.

Выводы

Таким образом, в данной главе рассмотрены основные проектные решения, необходимые для практического использования предлагаемой методики исследования механических свойств аппаратов внешней фиксации. Установлены требования к информации, циркулирующей в системе, а также разработана алгоритмическая составляющая всего процесса. С учетом разнообразных факторов выбраны инструментальные средства, которые

целесообразно использовать на той или иной стадии предлагаемого трехэтапного процесса.

3 ПРАКТИЧЕСКОЕ ИССЛЕДОВАНИЕ НАПРЯЖЕННО-ДЕФОРМИРОВАННЫХ СОСТОЯНИЙ АППАРАТОВ ВНЕШНЕЙ ФИКСАЦИИ

3.1 Создание примера модели аппарата внешней фиксации

Для проведения процесса моделирования была выбрана довольно стандартная схема аппарата внешней фиксации Илизарова. Важным фактором выбора именно данной конструкции является ее широкая вариативность – как в части диаметра колец, так и расстояния между их плоскостями. То есть целесообразно иметь электронную модель такой системы, чтобы вводить в нее разные размеры на входе и получать результат расчетов НДС на выходе.

Указанная модель была реализована в системе SolidWorks 2013 – рис. 3.1.

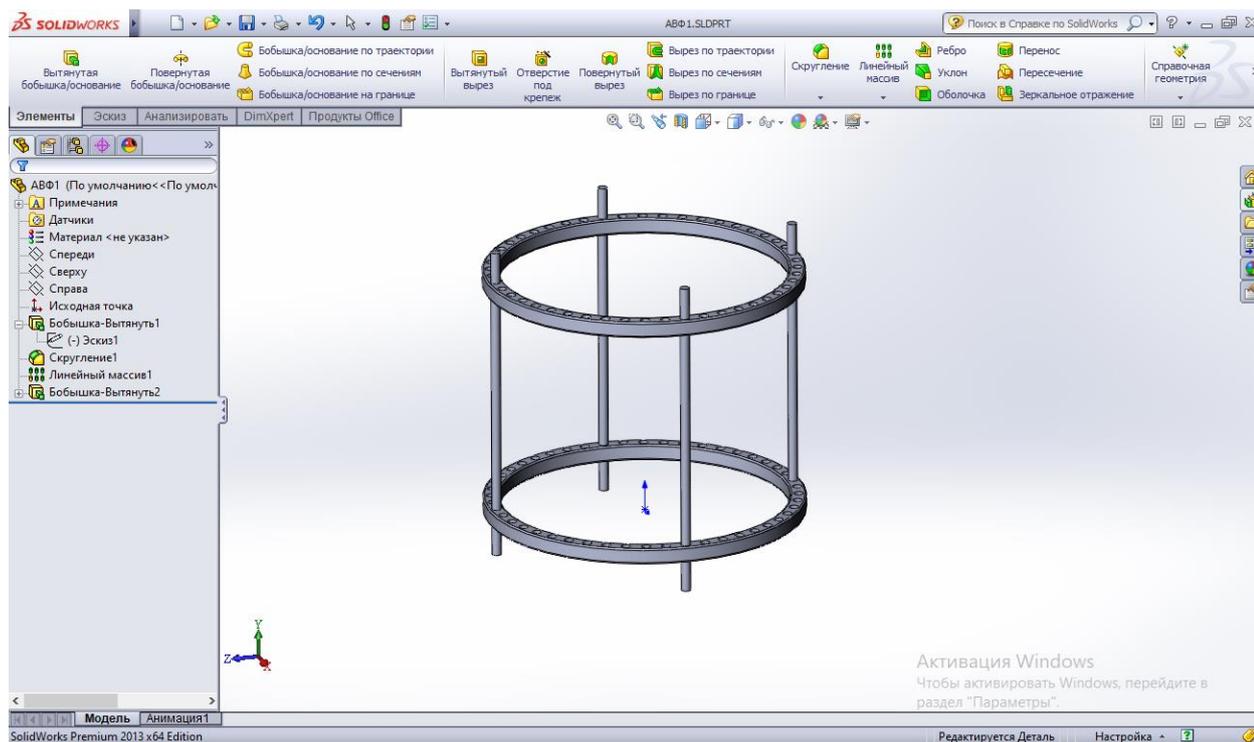


Рис. 3.1 Реализованная модель АВФ в CAD-системе SolidWorks 2013

Указанная модель настраивается с помощью параметров, задаваемых в численной форме с использованием панели «Модель» системы SolidWorks – рис. 3.2.

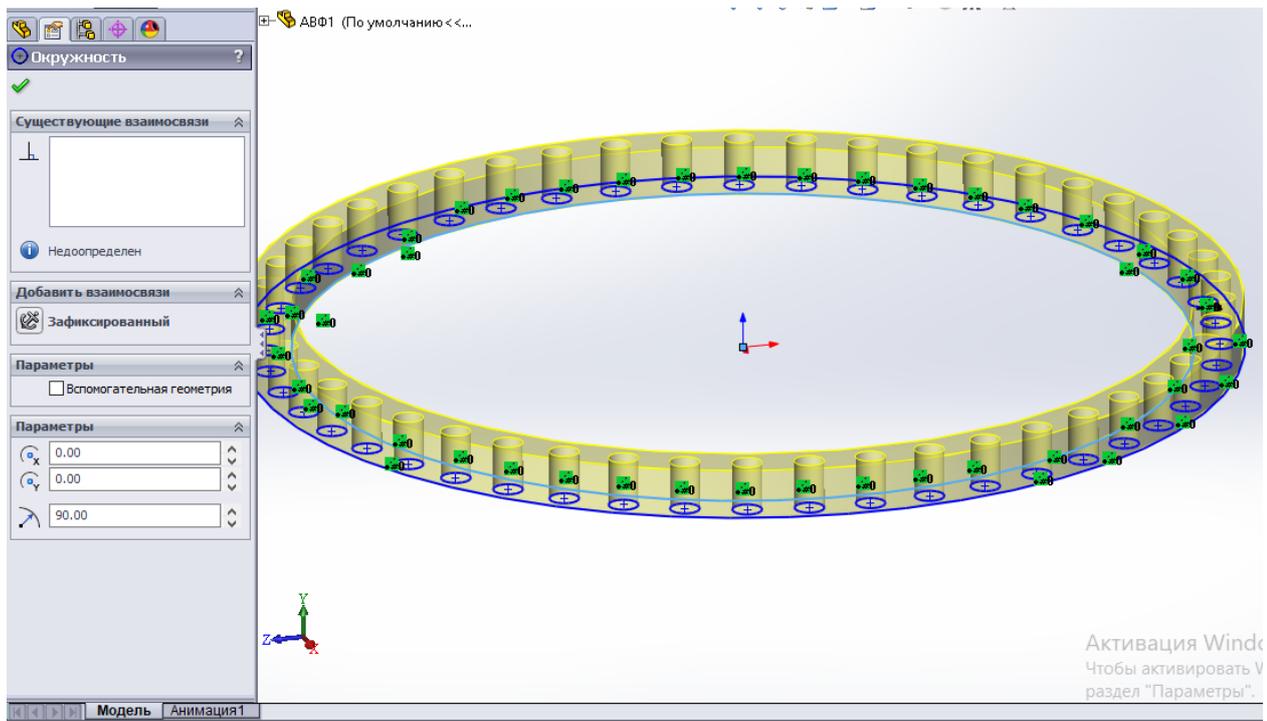


Рис. 3.2 Пример настройки параметров созданной модели АВФ

3.2 Моделирование напряженно-деформированного состояния спроектированного аппарата

Разработанная на предыдущем этапе модель была импортирована в среду ANSYS, в которой производились числовые расчеты напряженно-деформированного состояния – рис. 3.3.

Следует отметить, что аппарат Илизарова изначально является по своему предназначению дистракционно-компрессионным, т.е. может выполнять как сдавливающее, так и растягивающее воздействие. Поэтому в данной работе можем сконцентрироваться на одном виде воздействия, например сдавливающем. При этом согласно третьему закону Ньютона сам аппарат испытывает растяжение. На рис. 3.3 показаны две силы, действующие на конструкцию аппарата в местах, где выплнено закрепление спиц к сжимаемой кости (показано две позиции).

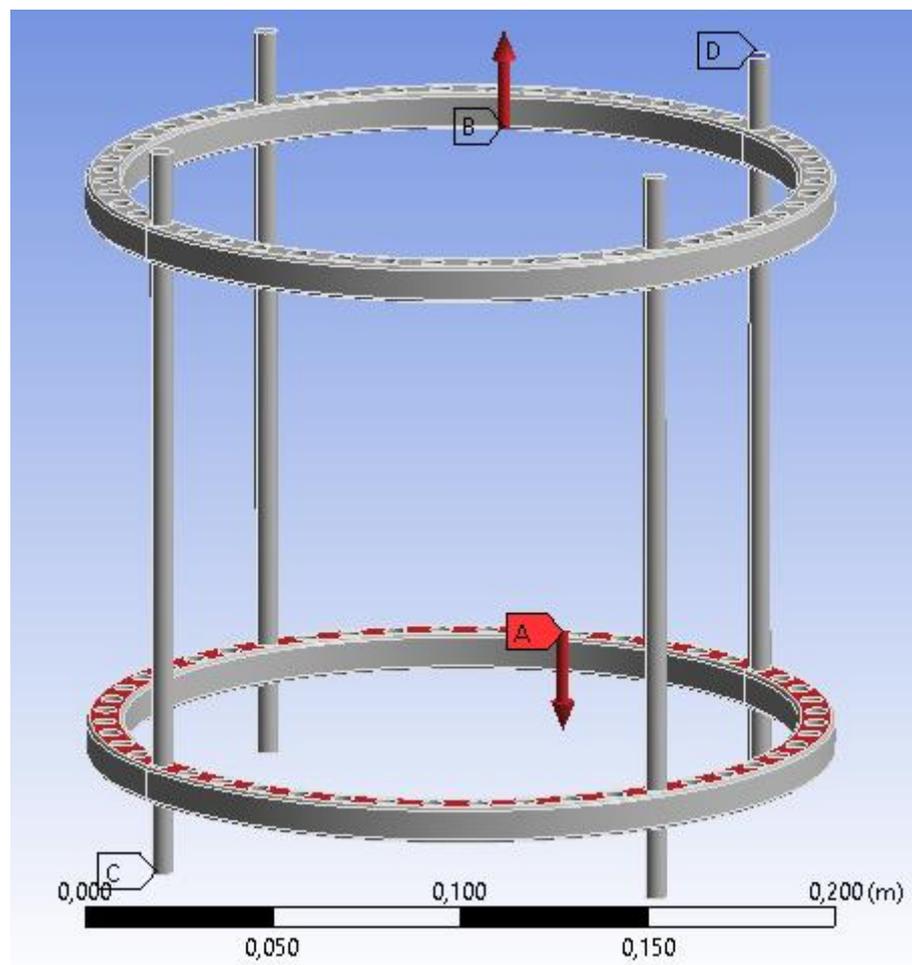


Рис. 3.3 Окно среды ANSYS, которая использовалась для моделирования напряженно-деформированного состояния АВФ, с указанием сил, действующих на аппарат

Первоначально был произведен статический механический анализ (Static Structural – по терминологии ANSYS), который позволяет просматривать результаты моделирования общих деформаций и действующих напряжений в форме привычной для специалистов по прочности – с преувеличенными деформациями, чтобы наглядно видеть проблемные места – рис. 3.4 и рис. 3.5 соответственно.

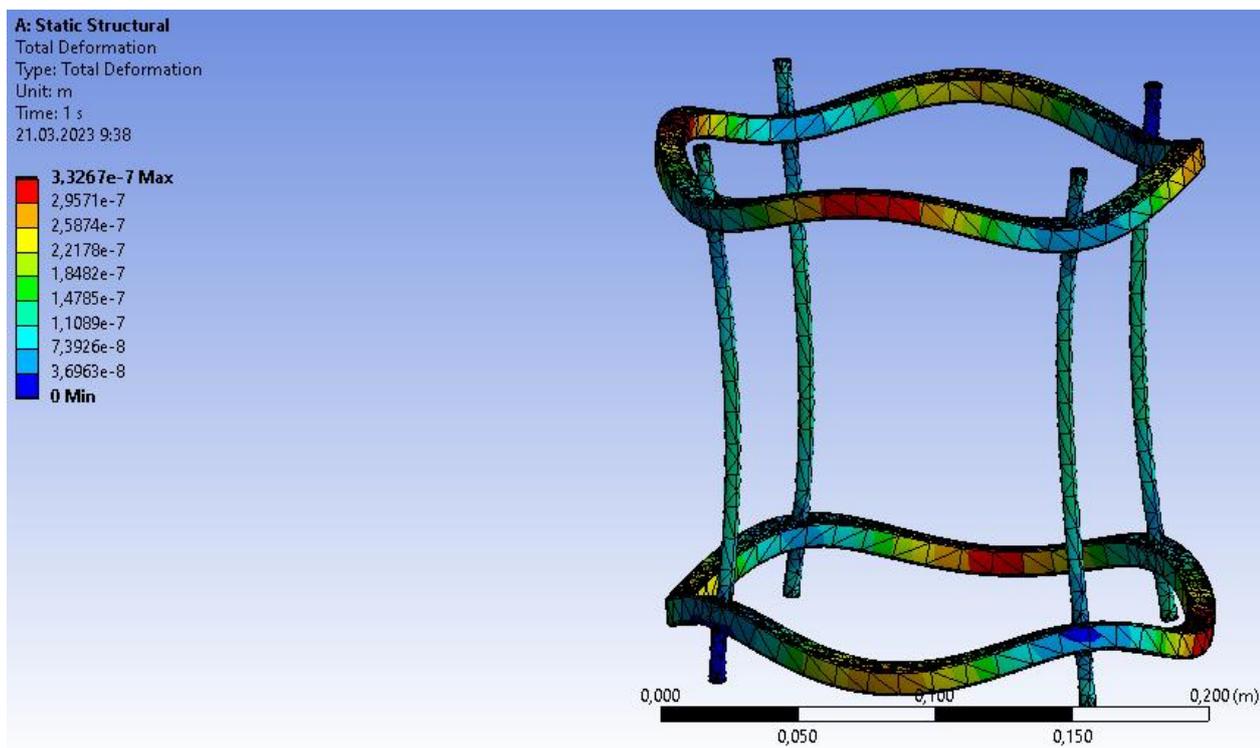


Рис. 3.4 Пример отображения результатов моделирования общей деформации аппарата Илизарова в среде ANSYS

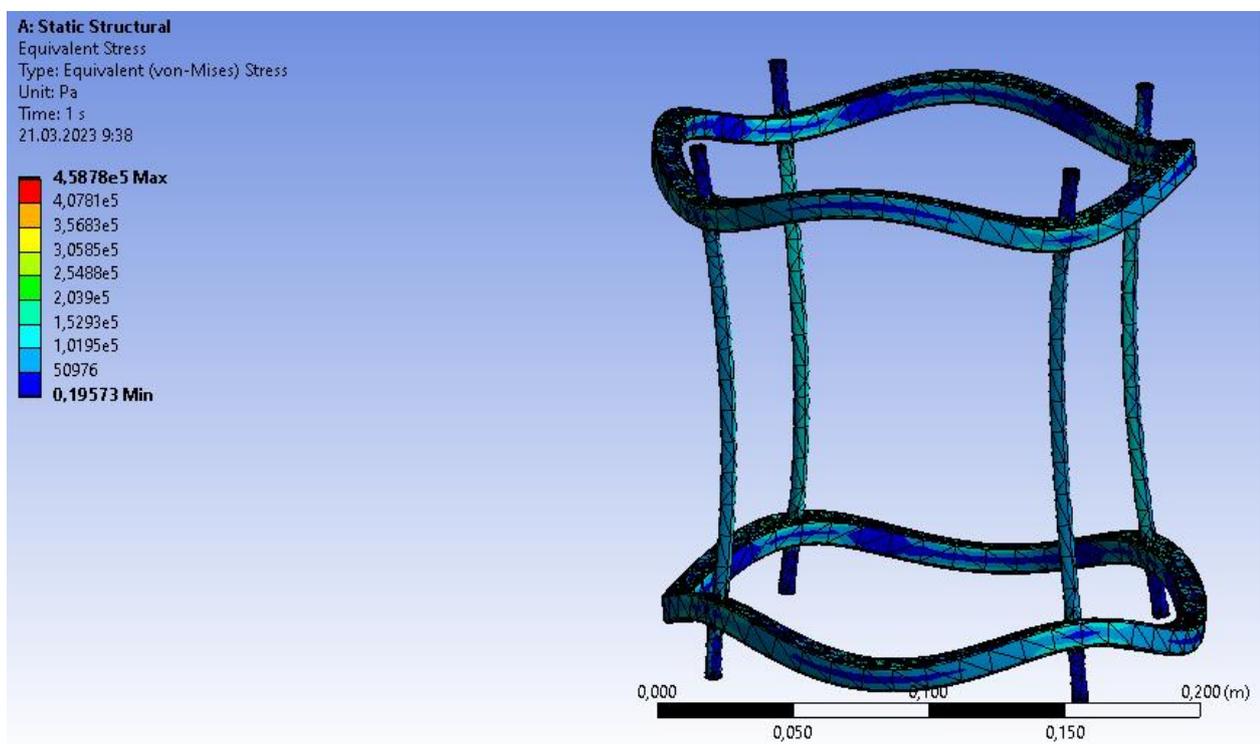


Рис. 3.5 Пример отображения результатов моделирования напряжений в конструкции аппарата Илизарова в среде ANSYS

На рисунках информация приведена также в числовой форме, где видно, что на самом деле максимальная деформация аппарата при прикладываемых нагрузках по 10 Н каждая составляет всего лишь $3,327 \cdot 10^{-7}$. Аналогично и максимально напряжение (достигаемое лишь в малых отдельных областях модели!), составляет $4,588 \cdot 10^5$ Па, что составляет ничтожно малую долю от предела текучести, находящегося в пределах $2 \cdot 10 \cdot 10^8$ Па для большинства конструкционных сталей. Таким образом, из анализа рис. 3.4-рис. 3.5 можно сделать вывод, что среда ANSYS отображает результаты в наглядной и понятной для профессионалов форме (гиперболизировано, чтобы нагляднее были видны проблемные места модели), но для обеих аудиторий такая форма представления результатов может быть неудобной (особенно с целями маркетинга, так как на рисунках кажется, что аппарат сильно деформируется). Именно поэтому, чтобы иметь более гибкие инструменты отображения данных, на рис. 2.1 отображена необходимость использования дополнительного программного обеспечения – средств визуализации результатов.

Чтобы иметь возможность коммуникации со сторонними программами, можно, например, сохранить результаты расчетов в файл (текстовый или другого формата), который затем открывать внутри средств визуализации. Итак, результаты расчетов были экспортированы в текстовый файл, пример которого показан на рис. 3.6. Для визуализации полученных данных согласно предложенной методики эти данные подавались на вход программы, описанной в следующем подразделе.

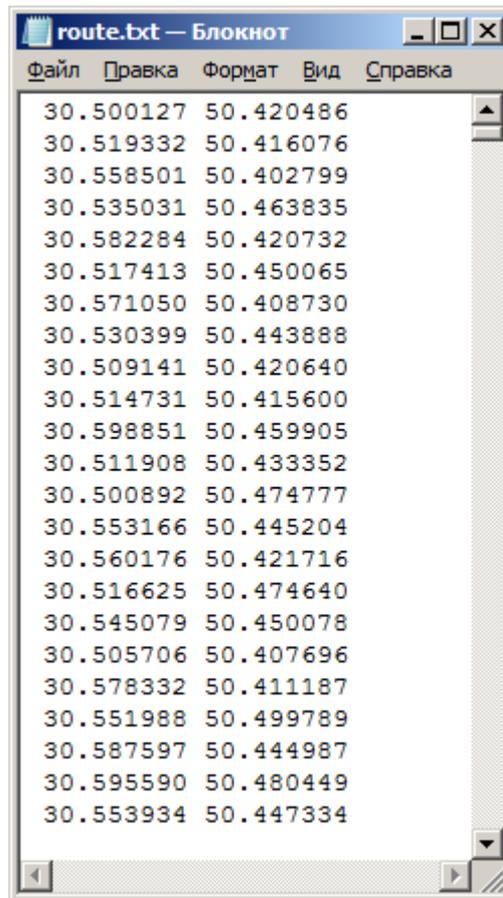


Рис. 3.6 Пример результатов расчета среды ANSYS, экспортированные в текстовый документ

3.3 Реализация средства визуализации результатов исследования

В программной реализации на языке C# использованы следующие решения:

а) считывание информации из TXT файла осуществляется на основе стандартного диалогового окна системы Windows с помощью кода:

```
if (openFileDialog1.ShowDialog() == DialogResult.Cancel)
    return;
string filename = openFileDialog1.FileName;
string fileText = System.IO.File.ReadAllText(filename);
fileText = fileText.Replace(".", ",");
string[] sep = { "\r\n" }, sep2 = { " " };
string[] lines = fileText.Split(sep,
System.StringSplitOptions.RemoveEmptyEntries);
```

Далее следует перевод полученных построчно значений в числовую форму. Для этого используется блок try-catch, поскольку, вообще говоря, файл с числами может иметь неверный формат и, например, там может располагаться текст, неконвертируемый в число:

```
try
{
    coord[i, 0] = Convert.ToDouble(co[0]);
    coord[i, 1] = Convert.ToDouble(co[1]);
    coord[i, 2] = Convert.ToDouble(co[2]);
}
```

Рисование графиков производится с помощью стандартного компонента языка C# Chart. Сначала он инициализируется кодом вида:

```
chart1.Series.Add("Перерезывающая сила, Н");
chart1.Series.Add("Расстояние, мм");
chart1.Series[0].ChartType = SeriesChartType.Spline;
chart1.Series[0].Color = Color.Red;
chart1.Series[1].ChartType = SeriesChartType.Spline;
chart1.Series[1].Color = Color.Blue;
```

Далее точки графика следует очистить, так как пользователь может по несколько раз подряд перестраивать график с помощью программы (а не только один раз после запуска – в этом случае данный код можно было бы опустить):

```
chart1.Series[0].Points.Clear();
chart1.Series[1].Points.Clear();
```

Для добавления новой точки на график используется код вида:

```
chart1.Series[0].Points.AddXY(coord[i, 0], coord[i, 1]);
```

Таким образом, рассмотрены основные особенности программного кода настольного приложения собственной разработки, предназначенного для визуализации результатов расчетов напряженно-деформированного состояния АВФ.

3.4 Оценка эффективности созданной программной системы исследования напряженно-деформированных состояний аппаратов внешней фиксации

Для разработанной программы на языке С# было проведено тестирование, которое проводилось по следующей методике:

- функциональное;
- системное;
- свободное;
- расширенное;
- ручное;
- бета-тестирование;
- по схеме белого ящика;
- позитивное и негативное.

Тестирование показало следующие результаты:

- работа созданного программного обеспечения для визуализации результатов расчетов проходит стабильно, без возникновения системных ошибок, аварийных завершений программы и других нештатных ситуаций;

- производительность созданного программного обеспечения находится на достаточном уровне: зависания, остановки или заметные паузы в работе программы отсутствуют при проведении отображения произвольных наборов данных, генерируемых системой ANSYS;

- в целом программный продукт адекватно выполняет поставленную перед ним задачу отображения данных, зафиксированную в задании на данную работу;

- работать с продуктом удобно, информация хорошо воспринимается (удобно размещена в окне приложения).

Переходя к оценке эффективности всей методики, можно сказать, что высокую эффективность обеспечивает ей ее модульность. Точки сопряжения между тремя обязательными компонентами предложенной методики сделаны

максимально гибкими: в первом случае (трехмерный редактор-средство моделирования) – это формат STL, являющийся на сегодняшний день одним из самых популярных методов представления трехмерных моделей, а во втором (средство моделирования-средство визуализации) – это вообще обычный текстовый формат TXT.

Таким образом, можно сказать, что разработанная методика исследования механических свойств аппаратов внешней фиксации в целом является гибкой, допускающей различные варианты реализации отдельных ее составляющих. В итоге появляется возможность реализовать полный цикл моделирования, начиная от создания пространственной модели и до наглядного представления результатов, что в целом позволяет экономить время на проведение и анализ соответствующих расчетов.

Выводы

Таким образом, в данной главе описан пример реализации предложенной методики исследования механических свойств аппаратов внешней фиксации, основанной на модульном применении трех составляющих. В разделе рассмотрено применение системы SolidWorks в качестве трехмерного редактора, продукта ANSYS – в качестве средства моделирования и языка C# - как средства для разработки программы, визуализирующей результаты расчетов.

ЗАКЛЮЧЕНИЕ

Таким образом, в данной работе разработана методика исследования механических свойств аппаратов внешней фиксации, представляющей собой полный цикл работ по автоматизации, начиная от создания модели, далее – самого моделирования и, наконец, представления результатов, что в целом позволяет экономить время на проведение соответствующих расчетов.

В первую очередь, исследованы особенности применения аппаратов внешней фиксации для разнообразных задач и выделены главные параметры этих процессов. Далее разработаны необходимые проектные решения по созданию соответствующего программного обеспечения: установлены требования к информации, обрабатываемой системой, разработан алгоритм использования предложенной методики, выбраны средства разработки (трехмерные редакторы – для первого этапа, пакет ANSYS Mechanical – для непосредственно моделирования напряженно-деформированного состояния, язык программирования C# в интегрированной среде разработки Microsoft Visual Studio Community Edition – для создания программы визуализации результатов расчетов).

Выполнена программная реализация соответствующего обеспечения и проведено его тестирование, которое показало достаточную эффективность предложенных решений и позволяет сделать выводы о возможностях его применения в реальных практических исследованиях и на производстве.

В работе разработан пример замкнутого цикла исследования аппарата внешней фиксации (по системе Илизарова) с использованием трех программных продуктов: 3D редактора, среды численного моделирования и специально разработанной (в рамках данной работы) программы для визуализации результатов расчетов.

В перспективе возможно описание расширенного спектра применяемых для предложенной методики программных продуктов, особенно с бесплатными лицензиями, в частности для расчета напряженно-деформированного состояния

(так как именно эти продукты наиболее дороги и требовательны к ресурсам). Также целесообразно разрабатывать более усовершенствованные средства визуализации, в которых разработанный в этой работе продукт можно брать за основу.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Аппарат для репозиции предплечья / Каплан А.В. [и др.] // Новые решения актуальных проблем в травматологии и ортопедии: учеб. пособие. М., 2001. С. 206–207.
- 2 Бойко, И.В. Сравнительный анализ напряжений при различных вариантах системы «кость-аппарат внешней фиксации» в области локтевого сустава [Текст] / И.В.Бойко, О.С.Раджабов, Д.С.Носивец // Запорож. мед. журн. – 2006. – Т.2, № 5(38). – С.26-32.
- 3 Введение в численные методы / А.А.Самарский. М.:Наука, 1987. 286 с.
- 4 Виноградов В.Г., Лапшин В.Л., Халиман Е.А. [и др.] Математическое моделирование и конструирование аппаратов внешней фиксации для лечения повреждений костей конечностей. Иркутск: НЦ РВХ СО РАМН, 2010. 136 с.
- 5 Дыдыкин А.В. Минимально инвазивный остеосинтез при лечении пострадавших с переломами длинных костей конечностей и нестабильными повреждениями таза: автореф. дис. ... д-ра мед. наук. М., 2007. 48 с.
- 6 Единак О.М. Идеальный остеосинтез. Атлас малоинвазивных хирургических технологий. Киев: Укрмедкнига, 2003. 176 с.
- 7 Метод конечных элементов и его применение в инженерных расчетах: учеб. пособие для вузов по направлениям подгот. дипломир. специалистов «Энергомашиностроение», «Прикладная механика», «Трансп. машины и трансп. технол. комплексы» / О.В. Крылов. М.: Радио и связь, 2002. 104 с.
- 8 Мителева З.М., Сухинин В.П., Меллеревич Г.М. Исследование напряженно-деформированного состояния проксимального конца бедренной кости методом конечных элементов // Ортопедия, травматология и протезирование. 1984. № 12. С. 16–20.
- 9 Метод конечных элементов в технике / пер. с англ. О.Зенкевич. М.: Мир, 1975. 542 с.

10 Оценка жесткости фиксации аппаратов внешней фиксации при повреждении монтеджиа с помощью моделирования напряженно-деформированного состояния / Бубнов А.С. [и др.]. Бюллетень Восточно-Сибирского научного центра СО РАМН. 2011. № 4–1. С. 266–273.

11 Основы строительной механики стержневых систем: учеб. для строит. специальностей вузов / Н.Н. Леонтьев, Д.Н. Соболев, А.А. Амосов. М.: Изд-во Ассоц. строит. вузов, 1996. 541 с.

12 Пичхадзе И.М., Рахимов А.Т., Рой Н.Н. Применение робототехники в реализации наружного чрескостного остеосинтеза //Ортопедия, травматология и протезирование. 1989. № 6. С. 42–46.

13 Пичхадзе И.М. Системный подход к выбору и компьютеризации стабильного чрескостного остеосинтеза при переломах длинных костей: дис. ... докт. мед. наук. М., 1994. 468 с.

14 Раздельное восстановление длины костей предплечья после прекращения их роста / А.И. Афаунов [и др.] // Вестник травматологии и ортопедии. 2008. № 2. - С. 44–48.

15 Швецов В.И., Швед С.И., Сысенко Ю.М. Чрескостный остеосинтез при лечении оскольчатых переломов. Курган, 2002. 137 с.

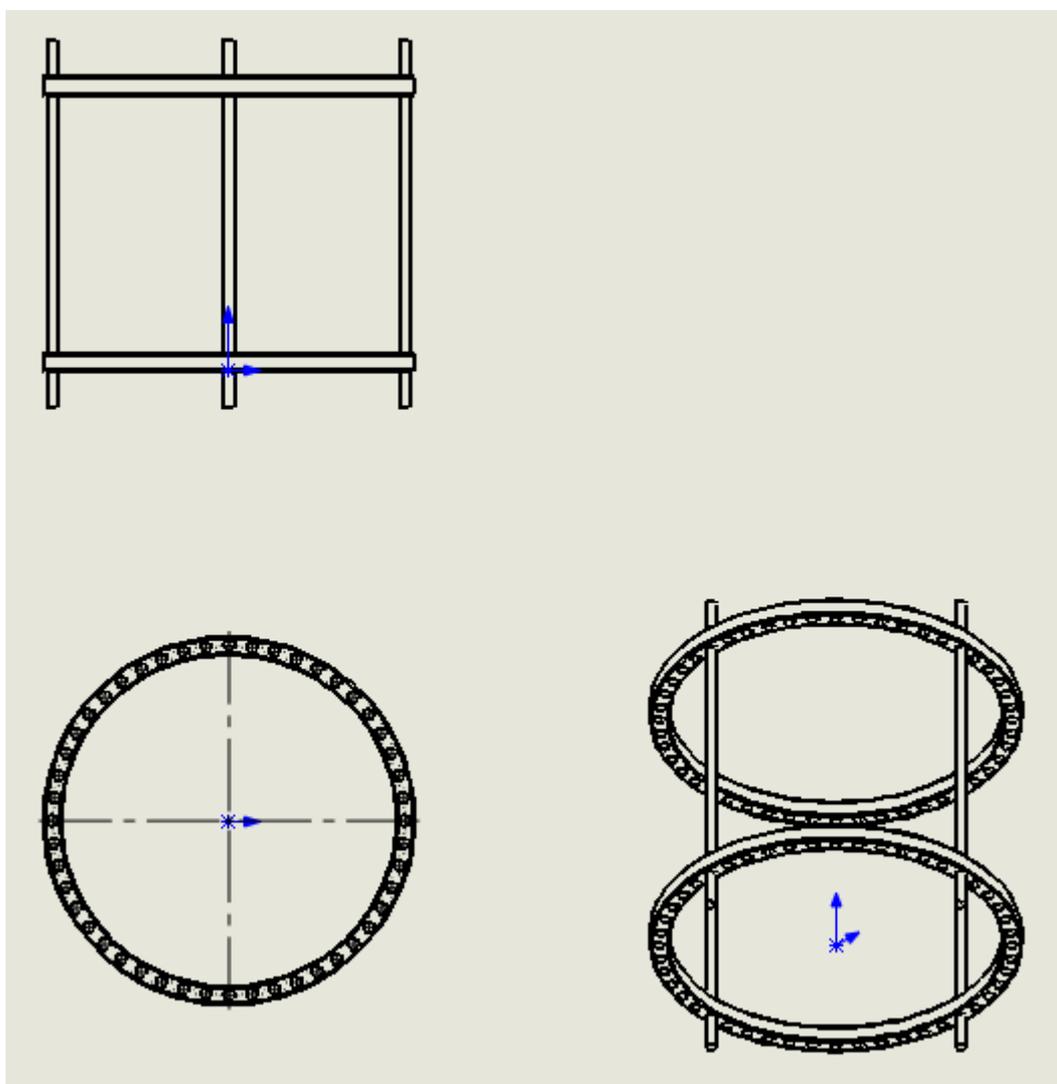
16 Oh J.K. Hybrid external fixation of distal tibial fractures: new strategy to place pins and wires without penetrating the anterior compartment / J.K. Oh, J.J. Lee, D.Y. Jung, B.J. Kim, C.W. Oh // Arch. Orthop. Trauma Surg. 2004. Vol. 124 (8). P. 542–546.

17 Oni O.O. Factors which may increase stresses at the pinbone interface in external fixation: a finite element analysis study / O.O. Oni, M. Capper, C. Soutis // Afr. J. Med. Med. Sci. 1999. Vol. 28 (1–2). P. 13–15.

18 Vazquez A.A. Finite element analysis of the initial stability of ankle arthrodesis with internal fixation: flat cut versus intact joint contours / A.A. Vazquez, H. Lauge-Pedersen, L. Lidgren, M. Taylor // Clin. Biomech. – Bristol, Avon, 2003. Vol. 18 (3). P. 244–253.

ПРИЛОЖЕНИЕ А

ОБЩИЙ ВИД РАЗРАБОТАННОГО АППАРАТА ВНЕШНЕЙ ФИКСАЦИИ



ПРИЛОЖЕНИЕ Б
ИСХОДНЫЙ ТЕКСТ ФАЙЛА FORM1.CS.

```
Файл Form1.cs.  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace EnumReducer  
{  
  
    public partial class Form1 : Form  
    {  
        protected const int MaxN = 10000;  
        protected const int MaxK = 100;  
        protected int n;  
        protected int N;  
        protected double eps;  
        protected double[,] coord = new double[MaxN,3];  
        protected int[] numinklas = new int[MaxK];  
        protected int[, ,] numpairs = new int[MaxK,MaxN*100,2];  
        //все пары точек.  
  
        public Form1()  
        {  
            InitializeComponent();  
        }  
    }  
}
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
private void выходToolStripMenuItem_Click(object
sender, EventArgs e)
{
    this.Close();
}

private void открытьToolStripMenuItem_Click(object
sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() ==
DialogResult.Cancel)
        return;
    string filename = openFileDialog1.FileName;
    string fileText =
System.IO.File.ReadAllText(filename);
    fileText = fileText.Replace(".", ",");
    string[] sep = { "\r\n" }, sep2 = { " " };
    string[] lines = fileText.Split(sep,
System.StringSplitOptions.RemoveEmptyEntries);
    string[] co;
    //string tempStr="";
    for(int i=0;i<lines.Length;i++)
    {
        //tempStr+=i.ToString()+":\t"
+lines[i)+"\r\n";
        textBox1.AppendText(i.ToString()+":\t"
+lines[i)+"\r\n");
        co = lines[i].Split(sep2,
System.StringSplitOptions.RemoveEmptyEntries);
        try
        {
            coord[i, 0] = Convert.ToDouble(co[0]);
            coord[i, 1] = Convert.ToDouble(co[1]);
            coord[i, 2] = Convert.ToDouble(co[2]);
        }
    }
}
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
    }
    catch (FormatException ex)
    {
        MessageBox.Show("Проверьте входной файл на
корректность!");
        return;
    }

    if (i >= MaxN - 1)
        break;
}
//textBox1.Text = tempStr;
button1.Enabled = true;
label6.Text = "Введите числа в левой части окна
программы (n, eps, N) и нажмите кнопку Показать";
}

private void oПрограммеToolStripMenuItem_Click(object
sender, EventArgs e)
{
    FormAbout about = new FormAbout();
    about.ShowDialog();
}

private void radioButton1_CheckedChanged(object
sender, EventArgs e)
{
    if (radioButton1.Checked)
        groupBox2.Enabled = false;
    else
        groupBox2.Enabled = true;
}

private void button1_Click(object sender, EventArgs e)
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
{
    double d1, d2, d3;
    int numfound=0;
    label6.Text = "Ожидайте окончания процесса...";

    try
    {
        n = Convert.ToInt32(textBox3.Text);
        eps = Convert.ToDouble(textBox4.Text);
        N = Convert.ToInt32(textBox5.Text);
        if (N > MaxN)
            N = MaxN;
    }
    catch (FormatException ex)
    {
        MessageBox.Show("Проверьте введенные числовые значения!");
        return;
    }
    for (int i = 0; i < MaxK; i++)
        numinklas[i] = 0;
    if (radioButton1.Checked)
    {
        textBox6.Text = "";
        int dd0 = Convert.ToInt32(DateTime.Now.ToString("dd"));
        int hh0 = Convert.ToInt32(DateTime.Now.ToString("HH"));
        int mm0 = Convert.ToInt32(DateTime.Now.ToString("mm"));
        int ss0 = Convert.ToInt32(DateTime.Now.ToString("ss"));
        for (int i=0;i<N;i++)
            for(int j=i+1;j<N;j++)
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
        for(int k=j+1;k<N;k++)
        {
            d1 = Math.Sqrt(Math.Pow(coord[i,
0] - coord[j, 0], 2) + Math.Pow(coord[i, 1] - coord[j, 1], 2) +
Math.Pow(coord[i, 2] - coord[j, 2], 2));
            d2 = Math.Sqrt(Math.Pow(coord[j,
0] - coord[k, 0], 2) + Math.Pow(coord[j, 1] - coord[k, 1], 2) +
Math.Pow(coord[j, 2] - coord[k, 2], 2));
            d3 = Math.Sqrt(Math.Pow(coord[i,
0] - coord[k, 0], 2) + Math.Pow(coord[i, 1] - coord[k, 1], 2) +
Math.Pow(coord[i, 2] - coord[k, 2], 2));
            if (Math.Abs(d1 - d2) < eps &&
Math.Abs(d2 - d3) < eps && Math.Abs(d1 - d3) < eps)
            {

textBox6.AppendText(i.ToString() + "\t" + j.ToString() + "\t" +
k.ToString() + "\r\n");
            }
        }

        int dd1 =
Convert.ToInt32(DateTime.Now.ToString("dd"));
        int hh1 =
Convert.ToInt32(DateTime.Now.ToString("HH"));
        int mm1 =
Convert.ToInt32(DateTime.Now.ToString("mm"));
        int ss1 =
Convert.ToInt32(DateTime.Now.ToString("ss"));
        label6.Text="Последнее действие заняло " +
((ss1 - ss0) + (mm1 - mm0) * 60 + (hh1 - hh0) * 3600 + (dd1 - dd0)
* 86400) + " секунд; построено полигонов: "+textBox6.Lines.Length;
    }
    else
    {
        textBox6.Text = "";
    }
}
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
int dd0 =
Convert.ToInt32(DateTime.Now.ToString("dd"));
int hh0 =
Convert.ToInt32(DateTime.Now.ToString("HH"));
int mm0 =
Convert.ToInt32(DateTime.Now.ToString("mm"));
int ss0 =
Convert.ToInt32(DateTime.Now.ToString("ss"));

//оценка расстояний в текущем наборе
координат, поиск нижней и верхней границ графика
Random rnd = new Random();
int numKlas;
double mind=1e+20, maxd=0, curd, deltad;
for (int i = 0, j, k; i < 100; i++)
{
    //int j, k;
    j = rnd.Next(N);
    k = rnd.Next(N);
    curd = Math.Sqrt(Math.Pow(coord[j, 0] -
coord[k, 0], 2) + Math.Pow(coord[j, 1] - coord[k, 1], 2) +
Math.Pow(coord[j, 2] - coord[k, 2], 2));
    if (curd > maxd)
        maxd = curd;
    if (curd < mind)
        mind = curd;
}
deltad = (maxd - mind) / (MaxK - 2);
//формирование расстояний
for (int i = 0; i < N; i++)
    for (int j = i + 1; j < N; j++)
    {
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
        d1 = Math.Sqrt(Math.Pow(coord[i, 0] -
coord[j, 0], 2) + Math.Pow(coord[i, 1] - coord[j, 1], 2) +
Math.Pow(coord[i, 2] - coord[j, 2], 2));
        if(d1<mind)
        {
            numpairs[0, numinklas[0], 0] = i;
            numpairs[0, numinklas[0], 1] = j;
            numinklas[0]++;
        } else
        if (d1 > maxd)
        {
            numpairs[MaxK-1, numinklas[MaxK -
1], 0] = i;
            numpairs[MaxK-1, numinklas[MaxK -
1], 1] = j;
            numinklas[MaxK-1]++;
        }
        else
        {
            numKlas = 1+(int)Math.Truncate((d1
- mind) / deltad);
            numpairs[numKlas,
numinklas[numKlas], 0] = i;
            numpairs[numKlas,
numinklas[numKlas], 1] = j;
            numinklas[numKlas]++;
        }
    }
    textBox2.Text = "";
    for (int i = 0; i < MaxK; i++)
        for (int j = 0; j < numinklas[i]; j++)
        {
            ns[0] = numpairs[i, j, 0];
            ns[1] = numpairs[i, j, 1];
        }
    }
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```

        d1 = Math.Sqrt(Math.Pow(coord[ns[0],
0] - coord[ns[1], 0], 2) + Math.Pow(coord[ns[0], 1] - coord[ns[1],
1], 2) + Math.Pow(coord[ns[0], 2] - coord[ns[1], 2], 2));
        for (int k = j + 1; k < numinklas[i];
k++)
        {
            ns[2] = numpairs[i, k, 0];
            ns[3] = numpairs[i, k, 1];
            if ((ns[0] != ns[2]) && (ns[0] !=
ns[3]) && (ns[1] != ns[2]) && (ns[1] != ns[3]))//это 4 разные
ТОЧКИ
                continue;
            /*if ((ns[2] == ns[0]) || (ns[2]
== ns[1]))
                ns[2] = ns[3];*/
            if (ns[2] == ns[0])
                continue;

            d2 =
Math.Sqrt(Math.Pow(coord[ns[1], 0] - coord[ns[2], 0], 2) +
Math.Pow(coord[ns[1], 1] - coord[ns[2], 1], 2) +
Math.Pow(coord[ns[1], 2] - coord[ns[2], 2], 2));
            d3 =
Math.Sqrt(Math.Pow(coord[ns[0], 0] - coord[ns[2], 0], 2) +
Math.Pow(coord[ns[0], 1] - coord[ns[2], 1], 2) +
Math.Pow(coord[ns[0], 2] - coord[ns[2], 2], 2));
            if (Math.Abs(d1 - d2) < eps &&
Math.Abs(d2 - d3) < eps && Math.Abs(d1 - d3) < eps)
                {
                    textBox6.AppendText(ns[0].ToString() + "\t" + ns[1].ToString() +
"\t" + ns[2].ToString() + "\r\n");
                }
        }
    }
}

```

```
}
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
int dd1 =  
Convert.ToInt32(DateTime.Now.ToString("dd"));  
int hh1 =  
Convert.ToInt32(DateTime.Now.ToString("HH"));  
int mm1 =  
Convert.ToInt32(DateTime.Now.ToString("mm"));  
int ss1 =  
Convert.ToInt32(DateTime.Now.ToString("ss"));  
label6.Text = "Последнее действие заняло " +  
((ss1 - ss0) + (mm1 - mm0) * 60 + (hh1 - hh0) * 3600 + (dd1 - dd0)  
* 86400) + " секунд";  
}  
}  
}
```