

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

На правах рукописи

Ефременков Иван Валерьевич

МОДЕЛИРОВАНИЕ И РАСЧЕТ ВЗАИМОДЕЙСТВИЯ
ВРАЩАЮЩИХСЯ УПРУГИХ ВИНТОВ С ГАЗАМИ

Специальность: 05.13.18 — Математическое моделирование, численные методы и
комплексы программ

Диссертация на соискание ученой степени кандидата технических наук

Научный руководитель:
доктор физико-математических наук, профессор
Леонтьев Виктор Леонтьевич

Ульяновск — 2019

ОГЛАВЛЕНИЕ

Введение.....	5
Глава 1. Обзор известных методов решения аэрогидродинамических задач в профессиональных комплексах программ. Проблематика прочности лопастей винта и резонансных состояний лопастей. Актуальность, новизна темы. Цели и задачи работы.....	9
1.1 Методы решения в ANSYS аэродинамической задачи обтекания поверхностей.....	9
1.1.1 Ограничения геометрических параметров в цифровых моделях.....	9
1.1.2 Постановка граничных условий и условий сопряжения в задачах аэрогидродинамики.....	11
1.2 Методы решения аэрогидродинамических задач в LS-Dyna.....	14
1.2.1 Подготовка исходных данных в препостпроцессоре LS-Dyna.....	14
1.2.2 Задание граничных условий для внешней воздушной среды в рамках метода SPH и условий сопряжения воздуха с лопастями.....	15
1.3 Проблематика прочности лопастей винта и методы её анализа.....	16
1.3.1 Общая характеристика лопастей винта и технологий их производства.....	16
1.3.2 Основные сведения о работе несущего винта вертолета одноосной схемы.....	18
1.4 Проблематика резонансных колебаний лопастей винта и методы их исследования.....	22
1.5 Актуальность, новизна темы. Цели и задачи работы.....	23
Глава 2. Математические модели взаимодействия упругих вращающихся лопастей винта с воздушной средой в комплексе программ ansys. Континуальная модель и дискретная модели	26
2.1 Математическая модель лопастей винта вертолета.....	26
2.1.1 Форматы различных CAD-систем в модуле Geometry.....	27
2.1.2 Возможности редактирования цифровой модели.....	28
2.1.3 Построение сетки конечных элементов в лопастей винта.....	28
2.1.4 Определение частот и форм свободных колебаний лопастей винта вертолета.....	31
2.2 Математическая модель воздушной среды.....	33
2.2.1 Определение в модуле Geometry размеров и формы области, занимаемой воздушной средой	34
2.2.2 Построение и перестройка сетки конечных элементов.....	35
2.3 Континуальная модель взаимодействия вращающихся упругих лопастей винта с воздушной средой (связанная задача).....	36

2.4 Дискретизация континуальной модели средствами метода конечных элементов в комплексе программ ANSYS.	38
2.5 Постановка граничных условий в задаче о взаимодействии лопастей винта с воздушной средой.....	39
2.5.1 Загрузка цифровых моделей лопастей и воздуха в модуле CFX и проверка их совместимости	39
2.5.2 Задание граничных условий на внешней границе воздушной среды.....	40
2.5.3 Установка параметров проведения расчета, форматов вывода результатов, а также точек мониторинга параметров процессов.....	42
2.5.4 Запуск модуля CFX и проверка исходных данных задачи.....	44
2.5.5 Учет частот свободных колебаний лопастей винта в расчетах динамического взаимодействия лопастей винта с воздушной средой.....	45
2.5.6 Расчеты. Анализ результатов расчета.....	46
Глава 3. Разработка нового конечного элемента на основе ортогональных финитных функций и его реализация в программной среде ANSYS. Применение модифицированного МКЭ, связанного с новым конечным элементом в решении связанной задачи аэроупругости.....	48
3.1 Общая структура пользовательского конечного элемента.....	48
3.2 Ортогональные финитные функции.....	60
3.3 Модификация структуры пользовательского конечного алгоритма. Формирование функций формы конечного элемента на основе ортогональных финитных функций.....	62
3.4 Компиляция динамической библиотеки модифицированных конечных элементов.....	66
3.5 Использование новых КЭ при решении тестовых задач теории упругости. Сравнительный анализ результатов расчетов с применением нового и стандартного конечных элементов.	66
3.6 Выводы.....	88
Глава 4. Программная среда решения задач динамики винтов и ее использование при проектировании.....	89
4.1 Обзор способов внедрения программной среды расчета динамики винтов в программный комплекс ANSYS.....	89
4.2 Общая структура программной среды решения технических задач динамики упругих винтов.....	91
4.3 Обзор функций и возможностей программной среды и её детализация.....	93
4.5 Алгоритм и блок-схема разработанной программной среды.....	105
4.6 Выводы. Инструкция по использованию программной среды.....	109

Глава 5. Возможности применения созданной программной среды. Решение актуальных технических задач.....	111
5.1 Обзор типов технических задач динамики взаимодействия винта с окружающей газовой средой	111
5.2 Техническая задача 1. Моделирование динамики лопастей винта, находящихся в воздушном потоке. Анализ достоверности.....	115
5.3 Техническая задача 2. Моделирование аэродинамики ветро-солнечной установки.....	126
5.4 Техническая задача 3. Моделирование резонансных колебаний лопастей воздушного винта, анализ образования вихрей.....	128
ЗАКЛЮЧЕНИЕ.....	131
СПИСОК ЛИТЕРАТУРЫ.....	133
ПРИЛОЖЕНИЕ А.....	139
ПРИЛОЖЕНИЕ Б.....	153
ПРИЛОЖЕНИЕ В.....	162

ВВЕДЕНИЕ

Моделирование представляет собой одну из основных форм отражения действительности, тех или иных свойств реальных объектов. Модель отображает основные, интересующие проектировщика и расчетчика, особенности объекта. Компьютерное моделирование — метод решения задачи анализа сложной системы на основе использования ее компьютерной модели. Суть компьютерного моделирования заключается в создании адекватной модели, в получении на ее основе количественных и качественных результатов. Цели компьютерного моделирования могут быть различными, однако наиболее часто моделирование является центральной процедурой системного анализа. Модель должна быть достаточно универсальной, чтобы описывать близкие по назначению объекты, и в то же время достаточно простой, чтобы позволить выполнить необходимые исследования с разумными затратами.

В современном мире компьютерное моделирование приобретает все большее значение в различных сферах производства. Для универсального моделирования объекта используются определенные программные пакеты, относящиеся к CAD-системам, которые обладают ограниченными способностями в расчете напряженно-деформированных задач. Помимо CAD-систем существуют и специализированные для инженерного анализа CAE-системы, такие как ANSYS, NASTRAN, DEFORM, Ls-Dyna и другие. Но даже они обладают ограничениями, связанными с их специализацией и с их возможностями, связанными с методом конечных элементов. Данные ограничения приводят к низкой точности расчетов, например, для моделей взаимодействия деформируемых лопастей винтов с воздушной средой на больших угловых и линейных скоростях. Повышение точности расчетов вызывает значительное увеличение времени расчета, недопустимый рост требуемой памяти ЭВМ и др. Для применения сложных программных продуктов требуется обученный персонал, способный работать не только с CAE-системами, но и с CAD.

Создание программных оболочек, содержащих препроцессоры, решатели и постпроцессоры и направленных на решение определенных классов технических задач с использованием существующих инженерных программных продуктов, снижает требования к инженерному персоналу, выполняющему расчеты, и уменьшает затрачиваемое время на расчеты. Такие программные оболочки позволяют получать решения актуальных технических задач за обозримое время с приемлемыми вычислительными затратами. Они могут содержать элементы научной новизны, позволяющие создавать требуемые модели и проводить необходимые исследования на основе этих моделей с целью проектирования новых эффективных технических конструкций.

Увеличение точности получаемых результатов, а также увеличение производительности вычислений возможно с помощью внедрения новых конечных элементов с разработкой специализированных программных оболочек для решения инженерных задач.

В связи с этим, математическое моделирование взаимодействия вращающихся упругих винтов с воздушной средой и создание численных методов исследования созданных моделей является **актуальной задачей**.

Цель данной работы состоит в создании и исследовании математических моделей динамического взаимодействия упругих винтов с окружающей средой с использованием конечно-элементного расчета со встроенными ортогональными финитными функциями, а также программной среды, реализующей эти метод и алгоритм.

Для достижения поставленной цели в работе решаются следующие **задачи**:

1. Проведение анализа современных способов моделирования взаимодействия исследуемого объекта с воздушной средой.

2. Разработка математических моделей динамики взаимодействия упругих лопастей винта с окружающей средой в программном комплексе ANSYS WorkBench.

3. Построение метода конечных элементов двухмерной и трехмерной теории упругости с использованием ортогональных финитных функций и соответствующих конечно-элементных моделей трехмерных упругих элементов конструкций.

4. Разработка программного комплекса конечно-элементного анализа, учитывающий взаимодействие с внешними программными библиотеками решателей систем алгебраических уравнений, с существующими препроцессорами и с комплексом ANSYS.

5. Применение созданного программного продукта к решению актуальных технических задач динамики упругих лопастей винтов с окружающей средой.

Объектом исследования являются процессы, возникающие при взаимодействии вращающихся лопастей винтов с газовой средой, и их изучение с помощью конечно-элементных моделей, а также реализация этих моделей и алгоритмов метода конечных элементов в программной среде.

Предметом исследования являются конечно-элементные математические модели и алгоритмы численных методов, а также приложение их программной реализации в технических задачах.

Научная новизна заключается в создании и исследовании конечноэлементного метода математического моделирования динамических процессов аэрогидроупругости, связанного с использованием ортогональных финитных функций, в его программной реализации, верификации и в получении решений актуальных технических задач с более высокими характеристиками, создающими возможности для повышения уровня проектирования лопастей воздушных винтов.

Теоретическая и практическая значимость работы заключается в возможности, с учетом применения ортогональных финитных функций, использования конечно-элементных моделей больших размерностей для решения технических задач динамического взаимодействия упругих тел с окружающей средой.

Основные положения, выносимые на защиту

1. Новые конечно-элементные модели упругих двумерных и трехмерных тел в динамике с окружающей средой, порождаемые алгоритмами методов конечных элементов;

2. Алгоритмы новых методов конечных элементов для двумерных и трехмерных задач напряженно-деформированного состояния упругих тел, связанные с использованием ортогональных финитных функций;

3. Комплекс программ конечно-элементного анализа, включающий в себя подпрограммы, реализующие алгоритмы метода конечных элементов, решатели, верификаторы, средства обработки входных и выходных данных и среду визуализации результатов.

Результаты основных положений диссертации докладывались и обсуждались на международной научно-практической конференции «XVIII Международной научно-практической конференции. Центр научной мысли.» (Таганрог, 2015 г.), на мероприятии по программе «IX Международной научно-практической конференции.» (Махачкала, 2015 г.), на международной научно-практической конференции «V Международной научно-практической конференции» (Ульяновск, 2016 г.), на всероссийской научно-практической конференции «II Всероссийская научно-производственная конференция, приуроченная ко Дню науки.» (Ульяновск, 2017 г.), на всероссийская научно-практическая конференция с международным участием «Новые технологии, материалы и оборудование российской авиакосмической отрасли» (Казань, 2018 г.), на конференции XIV Международную научную конференцию «Дифференциальные уравнения и их приложения в математическом моделировании» (Саранск, 2019 г.)

Результаты работы использовались при разработке подсистемы проектирования электронных технологических процессов изготовления авиационных деталей при выполнении государственного задания Министерства образования и науки РФ №2.1816.2017/4.6 по теме «Исследование и разработка интегрированной автоматизированной системы управления производственно-технологическим планированием авиастроительного предприятия на базе цифровых технологий».

Диссертация состоит из введения, пяти глав, заключения и списка литературы, включающего 84 наименований, и приложения. Основная часть работы изложена на 216 страницах машинописного текста и содержит 93 рисунка и 35 таблиц.

В первой главе описаны и рассмотрены существующие методы решения аэрогидродинамических задач в современных инженерных программных комплексах. Современные численные методы, основанные на вариационных принципах механики деформируемого твердого тела и аэрогидродинамики, позволяют проводить анализ напряженно-деформированного состояния твердых тел в их взаимодействии с газами и жидкостями.

Проведено исследование причин выхода из строя лопастей винтов летательных аппаратов во время их работы и способы проведения натуральных экспериментов при определении их прочностных показателей. Выполнен обзор научных работ в соответствующей области аэрогидродинамики, на этой основе обоснована актуальность и новизна работы.

Во второй главе рассматриваются континуальные и дискретные математические модели динамического взаимодействия вращающихся упругих лопастей винта с воздушной средой, реализованные в комплексе программ ANSYS.

Третья глава посвящена реализации и исследованию авторского конечного элемента на основе ортогональных финитных функций и модификации алгоритма МКЭ решения плоских и трехмерных задач теории упругости в программном комплексе ANSYS.

В четвертой главе рассмотрена реализация программной среды. Раскрывается структура программной среды. Описаны механизмы межпрограммного взаимодействия и применение нового конечного элемента.

В пятой главе даются постановки и решения актуальных технических задач с применением разработанного программного комплекса на основе нового метода с использованием ОФФ.

По теме диссертации опубликовано 14 работ, в том числе 6 статей из перечня изданий, рекомендованных ВАК, 4 работы в других рецензируемых журналах, сборниках конференций, симпозиумов и семинарах, 1 статья из перечня изданий, индексируемых Scopus, 1 учебно-методическое пособие, 2 свидетельства о государственной регистрации программы для ЭВМ (№2016660994, №2018666901).

Автор выражает глубокую признательность научному руководителю профессору Леонтьеву В.Л. и кафедре ММТС Ульяновского государственного университета за оказанную помощь и поддержку в процессе подготовки работы.

1. ОБЗОР ИЗВЕСТНЫХ МЕТОДОВ РЕШЕНИЯ АЭРОГИДРОДИНАМИЧЕСКИХ ЗАДАЧ В ПРОФЕССИОНАЛЬНЫХ КОМПЛЕКСАХ ПРОГРАММ. ПРОБЛЕМАТИКА ПРОЧНОСТИ ЛОПАСТЕЙ ВИНТА И РЕЗОНАНСНЫХ СОСТОЯНИЙ ЛОПАСТЕЙ. АКТУАЛЬНОСТЬ, НОВИЗНА ТЕМЫ. ЦЕЛИ И ЗАДАЧИ РАБОТЫ

1.1. Методы решения в ANSYS аэродинамической задачи обтекания поверхностей

1.1.1 Ограничения геометрических параметров в цифровых моделях

Комплекс программ ANSYS WorkBench обладает различными встроенными модулями как проектирования и экспорта цифровых моделей исследуемых объектов, так и проведения инженерных расчетов на основе полученных моделей.

Перед началом проведения расчета требуется определить исследуемую модель существующего объекта. В современной методике решение задач обтекания поверхностей упругого тела, моделирование начинается с вычитания из созданной области газа, обычно в форме параллелепипеда, области, определяемой геометрией цифровой модели. Такой принцип позволяет сократить технические требования к вычислительной машине, так как сокращается количество конечных элементов при наложении сетки на исследуемое тело. Однако у данного метода имеются ограничения, так как в качестве исследуемой модели рассматривается объект с бесконечной длиной и проводится расчет лишь с некотором его поперечным сечении [1,2,3,4].

Развитие этого метода дает возможность вычитания уже целой модели, например, крыла самолета, из окружающего пространства, но и при этом возникают новые геометрические ограничения. При операции вычитания простой и однородной модели выполняется булевская операция вычитания, если же операция проводится со сложной конструктивной моделью, содержащей в себе более 100 элементов различной формы и размеров, то возрастает сложность этой операции и технические требования к вычислениям. Помимо этого, после вычитания сложной конструкционной модели появляются различные трудности, связанные с внутренним пространством модели, для дальнейшего задания граничных условий [5,6,7].

Рассмотрим два метода расчета обтекаемости крыла самолета, летящего со скоростью 800 километров в час. Для расчета была взята цифровая модель с официального сайта передачи геометрий крыла самолета Airbus A350 (Рисунок 1.1).

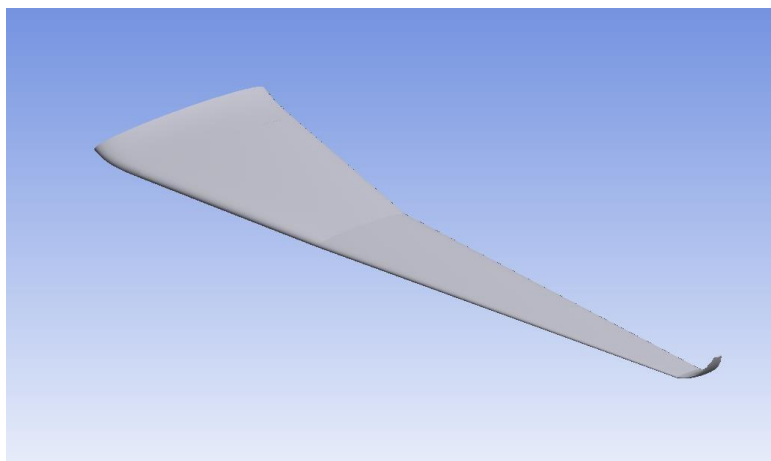


Рисунок 1.1. Цифровая модель крыла самолета Airbus A350

Первый вариант метода заключается в выборе поперечного сечения в требуемом месте крыла самолета и дальнейшая работа с ним.

В данном примере сечение было выбрано примерно посередине длины размаха полукрыла. Была построена двумерная область окружающего пространства. Метод предназначен для изучения объекта бесконечного размаха, поэтому проводится операция вычитания области сечения крыла из созданного сечения окружающего пространства (Рисунок 1.2).

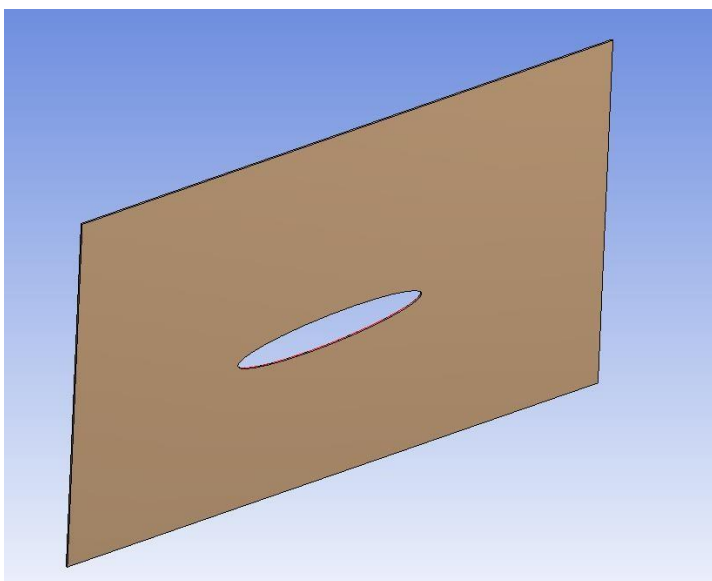


Рисунок 1.2. Окружающее пространство, окружающее сечение крыла бесконечной длины

Также рассматривался развитый метод расчета с использованием трехмерной цифровой модели крыла. Результатом обработки геометрии во втором варианте метода является трехмерная область окружающего пространства, из которой удаляется объект исследования – трехмерная модель полукрыла (Рисунок 1.3).

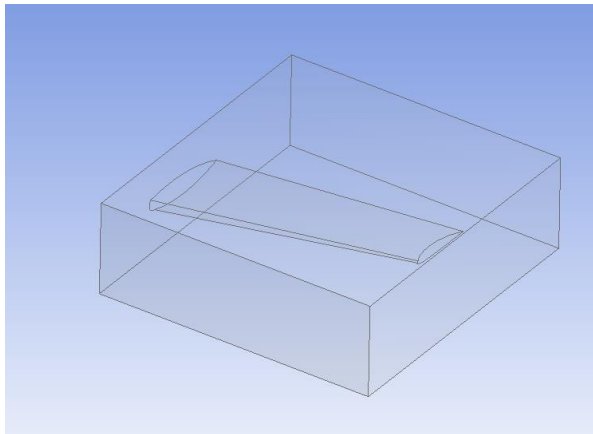


Рисунок 1.3. Трехмерная область газа с вырезанной из нее областью крыла

Так как созданная область далее используется в модуле CFX, предназначенном для проведения аэродинамических исследований, то необходимо сменить свойство области газа с «Solid» на «Fluid».

Следующим шагом является создание сетки конечных элементов. Для этого существуют два модуля: «Mesh» и «Mechanical Mesh». В первом модуле создается простая сетка, которая ориентирована на заданный материал с меньшим количеством стандартных настроек, во втором модуле перед созданием сетки программа позволяет выбрать сетку из предоставленной библиотеки или создать свою. Все рекомендуемые программой параметры настроек рекомендуется вначале оставить без изменения, они отвечают в основном за сглаженность или сгущение сетки у границ области. Далее необходимо задать граничные условия для двух видов геометрии. В результате рассмотрения этой задачи можно сделать вывод о том, что тело представляет собой абсолютно твердое тело, которое не подвергается деформации. Из результатов задачи возможно получить значения скорости потока и образования вихревых потоков вокруг исследуемого тела. Данная задача основывается на стандартном методе моделирования аэрогидродинамических задач.

1.1.2 Постановка граничных условий и условий сопряжения в задачах аэрогидродинамики

Данный этап очень важен, от качества постановки граничных условий (ГУ) зависит процесс расчета, его устойчивость и точность полученных результатов.

Постановка ГУ в ANSYS CFX, проводится на всех частях поверхности области, загруженной в препроцессор CFXPre.

Рассмотрим постановку ГУ для стационарного и нестационарного режимов движения газа. При постановке нестационарного расчета программа автоматически изменяет геометрию области газа на всех шагах по времени в соответствии с промежуточными результатами расчета. Для каждого положения газа решается задача, установившегося течения газа. Полученный массив результатов обрабатывается и

выводится для анализа в удобной табличной или графической форме. Подобные расчеты требуют огромных временных и машинных затрат, а оправдывается это увеличением адекватности модели и точности получаемых результатов, поскольку в трехмерной модели уже учитываются краевые эффекты, а также информативными и наглядными картинками течения. Это позволяет глубоко и полно изучить характер течения и оперативно обнаружить возможные негативные явления.

Далее будут использоваться созданные на предыдущих этапах два вида геометрических моделей. Сетка должна быть удовлетворительного качества.

Вначале требуется определить фрагменты поверхности области, на которых задаются различные граничные условия. В двумерном случае, где объект исследования – крыло, обладает бесконечным размахом, задаются две части границы области, соответствующих входу (скорость потока) и выходу (давление внешней среды) воздушного потока, на двух других частях внешней границы области, а также и на поверхности крыла задается граничное условие непроницаемости, (рис 1.4).

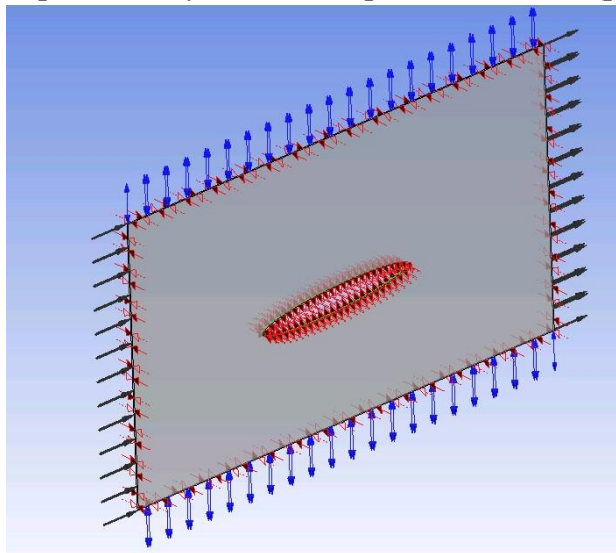


Рисунок 1.4. Отражение граничных условий, заданных для расчета обтекания крыла самолета бесконечного размаха

Во втором случае, где объектом исследования является трехмерная область, задаются, граничные плоские области входа и выхода воздушного потока и граничные условия на них, на четырех других плоских частях границы области и на криволинейной поверхности крыла задаётся граничное условие непроницаемости (Рисунок 1.5)

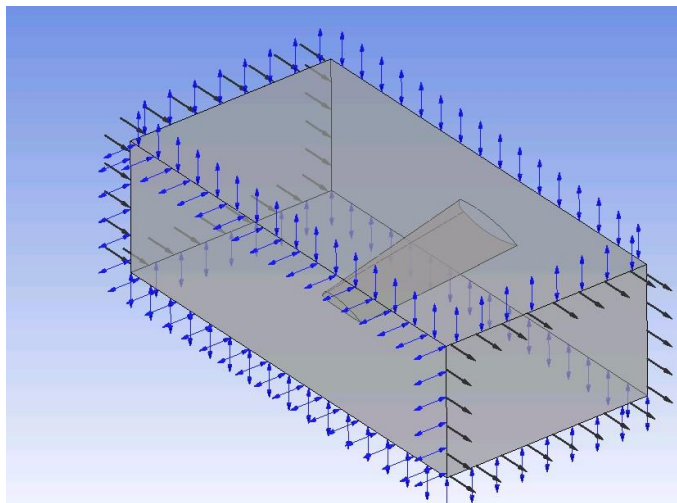


Рисунок 1.5. Заданные граничные условия для расчета обтекаемости крыла самолета

После определенных шагов в настройках решателя, можно запускать сам расчет. Просмотр результатов становится доступным после окончания расчета.

Результаты по первому варианту позволяют увидеть аэродинамические характеристики в двумерном виде (скорость воздушного потока и создаваемое им давление на поверхности в выбранном сечении) (Рисунок 1.6).

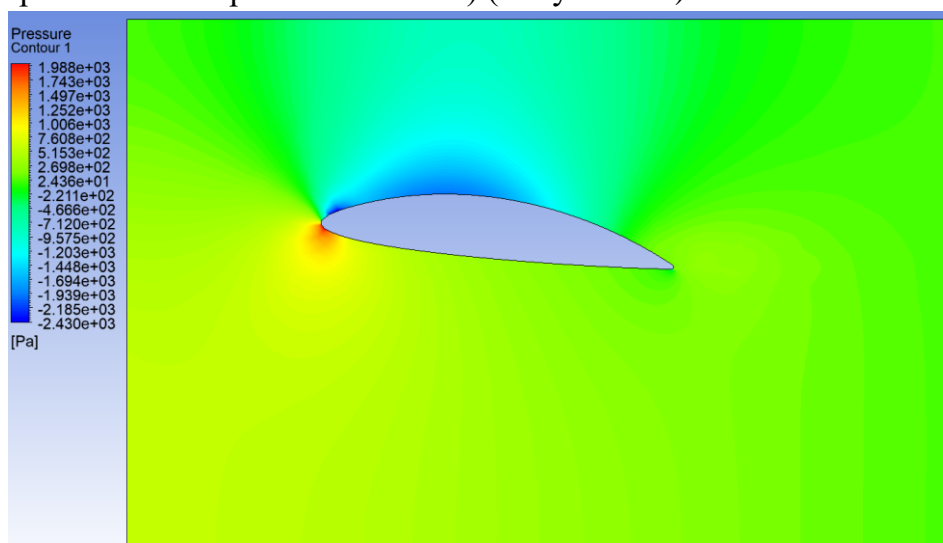


Рисунок 1.6. Давление, возникающее в области крыла бесконечного размаха

Получаемые в таком расчете результаты можно использовать в другом модуле программного комплекса, ориентированном на проведение прочностного расчета, в котором определяется деформация крыла под действием потока газа. Процесс экспорта аэродинамических показателей происходит на основе результатов, получаемых на указанных заранее поверхностях (Рисунок 1.7).

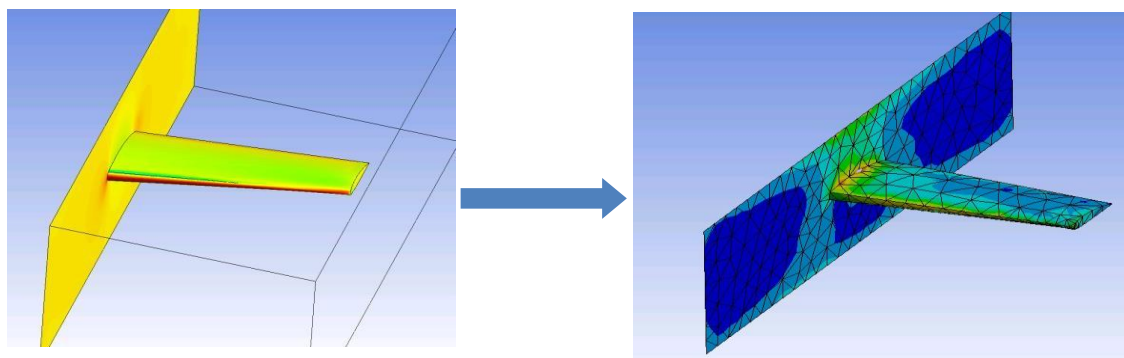


Рисунок 1.7. Давление на поверхности крыла и напряжения, возникающие в крыле самолета

1.2 Методы решения аэрогидродинамических задач в LS-Dyna

LS-Dyna в большой степени предназначена для моделирования обтекания различных объектов. Программный продукт позволяет проводить расчеты динамики взаимодействия твердых тел с окружающим пространством с применением SPH метода.

Метод SPH (Smoothed Particle Hydrodynamics) – гидродинамика сглаженных частиц) является бессеточным Лагранжевым методом, в котором локальные системы координат движутся вместе с SPH частицами. Согласно методу SPH заготовка представляется дискретными элементами, называемыми частицами. Эти частицы характеризуются пространственным расстоянием или «длиной сглаживания», обычно представляемым в уравнениях параметром h . Величина h характеризует расстояние, на котором свойства частиц «сглаживаются» [8,9,10].

Математически это описывается функцией ядра. В качестве функции ядра часто используется функция Гаусса. Принцип близкодействия позволяет экономить в расчётах вычислительные ресурсы, исключая относительно слабое влияние отдалённых частиц. Значение любой физической величины A в точке X задаётся формулой:

$$A(X) = \sum_j m_j \frac{A_j}{\rho_j} W(|X - X_j|, h), \quad (1)$$

где m_j – масса частицы j , A_j – значение величины A для частицы j , ρ_j – плотность, связанная с частицей j , W – функция ядра.

1.2.1 Подготовка исходных данных в препостпроцессоре LS-Dyna

При проведении рассматриваемого расчёта использовалась в программном продукте LS-DYNA модель RIGID (абсолютно жесткий материал) лопастей винта для уменьшения процессорного времени расчета. Исследуемая модель представляла собой упрощенную геометрию лопастей винта, расположенного в воздушном пространстве

(Рисунок 1.8). Это было сделано средствами программного продукта при помощи метода SPH.

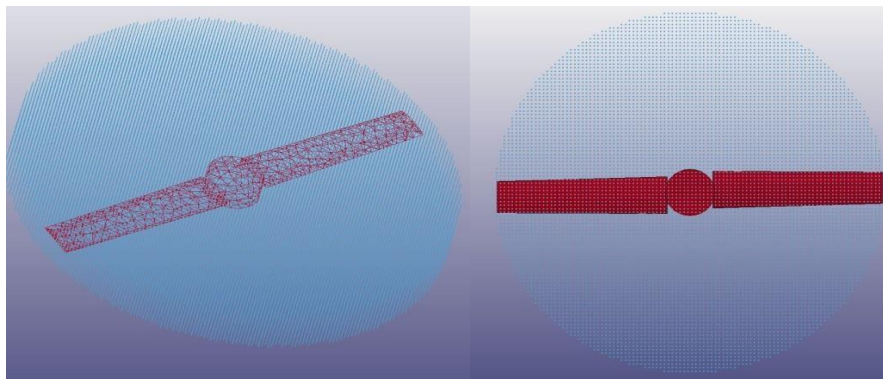


Рисунок 1.8. Цифровая модель лопастей винта, расположенного в воздушном пространстве, заданном методом SPH

Кроме этого, для проведения расчета необходимо провести настройки вручную параметров выводимых данных, выбирая их и моменты их записи в базу данных (Рисунок 1.9).

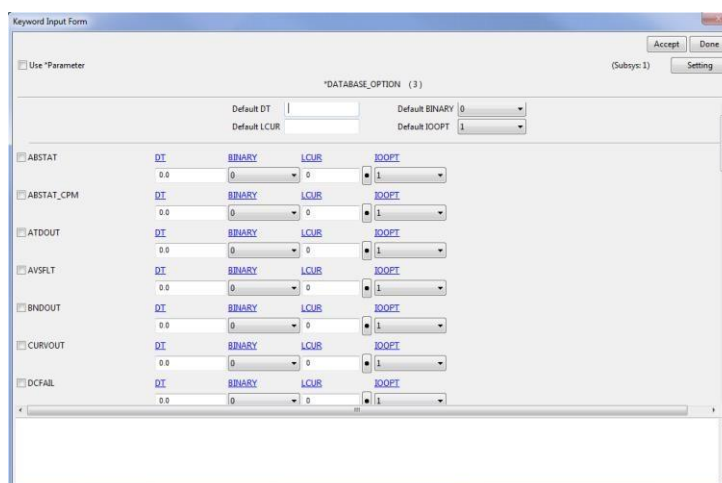


Рисунок 1.9. Окно задания параметров выводимых результатов расчета

1.2.2 Задание граничных условий для внешней воздушной среды в рамках метода SPH и условий сопряжения воздуха с лопастями

Важным моментом при задании граничных условий является указание определенных параметров контакта частиц воздушного пространства с поверхностями лопастей винта. Требуется указание степени шероховатости и условий проникновения этих частиц через поверхность модели (Рисунок 1.10).

Рисунок 1.10. Окно выбора параметров контакта лопасти винта и газа
Процесс вращения винта задается следующими двумя параметрами (Рисунок 1.11):

1. Первый параметр содержит в себе информацию о том, какое это движение и что движется. Кроме этого, задаются параметры, отвечающие за переносное перемещение отдельных частей модели, если их движение отличается от движения всей конструкции.
2. Второй параметр является базовым для первого. В первом параметре содержится общая информация о свойствах движения. Второй параметр характеризует само это движение, посредством задания некоторой функции и соответствующего графика, содержащих в себе информацию об угловой скорости вращения в различные моменты времени.

Рисунок 1.11. Окна, содержащие информацию о движении моделей проекта.

1.3 Проблематика прочности лопастей винта и методы её анализа

1.3.1 Общая характеристика лопастей винта и технологий их производства

В Таблице 1 приведены технические характеристики лопастей винта вертолета Ми-171 [10].

Таблица 1

Технические характеристики лопастей

Характеристика	Значение
Стандартный назначенный ресурс, ч	2000
Масса, кг	134
Коэффициент использования материала	0,3
Расход металла, кг	576

Анализ трудоемкости изготовления лопастей (Таблица 2) показывает, что наиболее трудоёмки в изготовлении носовая часть и лонжерон лопасти [11].

Таблица 2.

Анализ трудоемкости изготовления лопастей

Элемент лопасти	Нормочасы (на 1 лопасть)	Процент от общей трудоемкости
Носовая часть	38	25,67
Лонжерон	39	26,35
Сборка лопасти	20	13,51

Схема технологического членения современной типовой лопасти винта вертолета представлена в работах [12,13,14]. Схема технологического членения лопасти винта вертолета представлена на рисунке 1.12

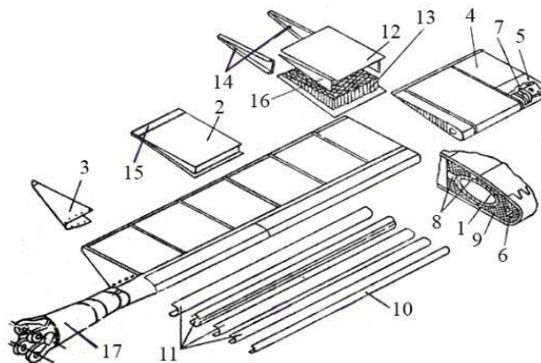


Рисунок 1.12. Схема технологического членения лопасти винта вертолета 1 – лонжерон, 2 – хвостовой отсек, 3 – комлевой обтекатель, 4,5 – концевые обтекатели, 6 – противообледенительная система, 7 – узел статической балансировки лопасти, 8 – компенсаторы, 9 – пенопластовый наполнитель, 10 – оковка-облицовка противообледенительной системы, 11 – пакет противообледенительной системы, 12 – обшивка, 13 – сотовый наполнитель, 14 – нервюры, 15 – закрылок, 16 – межотсечный вкладыш, 17 – комлевая часть лопасти

Авторы работ [15,16,17,18,19] в качестве упрочняюще-отделочной обработки используют виброударное упрочнение на установке, показанной на рисунке 1.13.

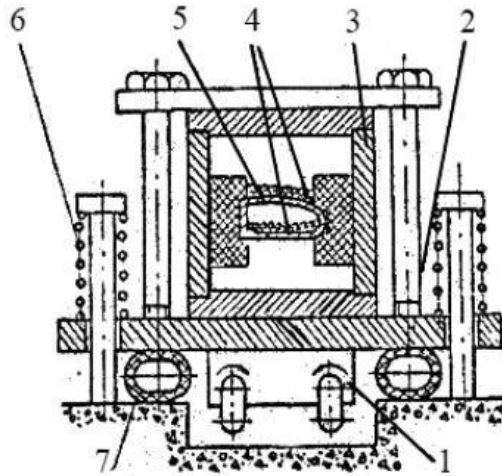


Рисунок 1.13. Установка для виброударного упрочнения 1 – дебалансные вибраторы, 2 – платформа, 3 – контейнер, 4 – упрочняющие шарики, 5 – лонжерон, 6 – подвеска, 7 – опорный элемент

Хвостовой отсек лопасти собирают в специальном приспособлении (Рисунок 1.14) [20,21,22,23].

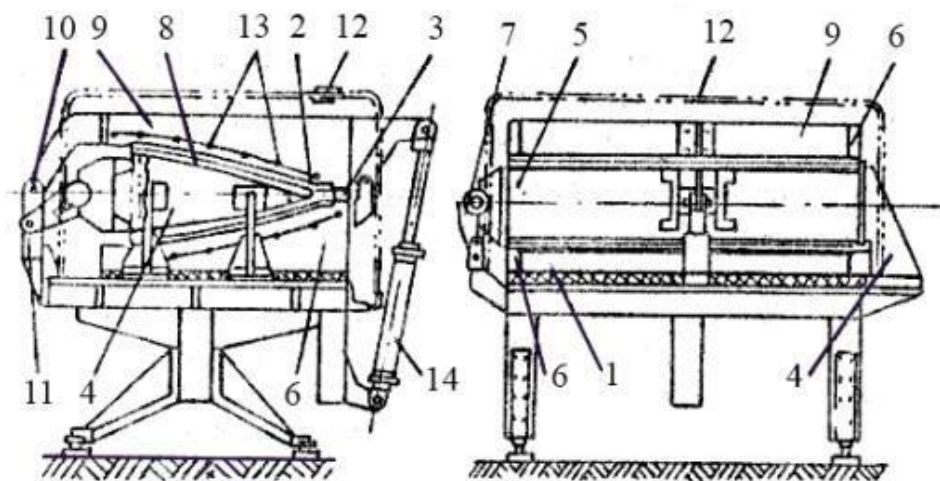


Рисунок 1.14. Приспособление для сборки хвостового отсека лопасти 1 – плита нижняя, 2 – асботкань, 3 – планка опорная, 4 – плита торцовая, 5,6 – плиты технологические, 7 – фиксатор, 8 – плита-крышка, 9 – плита верхняя, 10 – фиксатор, 11 – вилка, 12 – кожух теплоизоляционный, 13 – электронагреватели, 14 – гидроцилиндр

1.3.2 Основные сведения о работе несущего винта вертолета одноосной схемы

Лопасть несущего винта является главным элементом вертолета, существенно определяющим его летные характеристики и безопасность эксплуатации. В связи с этим

конструкция и технологии изготовления лопасти являются предметом интенсивных исследований и областью внедрения новейших разработок. Как правило, вращение несущего винта зарубежных вертолетов осуществляется против часовой стрелки, а у российских вертолетов - по часовой. (Рисунок 1.15).

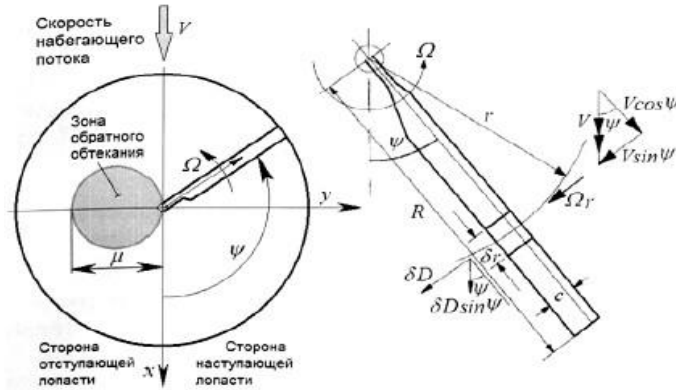


Рисунок 1.15. Схема диска несущего винта и распределения скоростей воздушного потока по лопасти при полете вперед

Скорость воздуха U можно разложить на параллельную (u_T) и перпендикулярную (u_P) составляющие относительно плоскости диска, образованной вращением винта (Рисунок 1.16). [24,25,26,27]

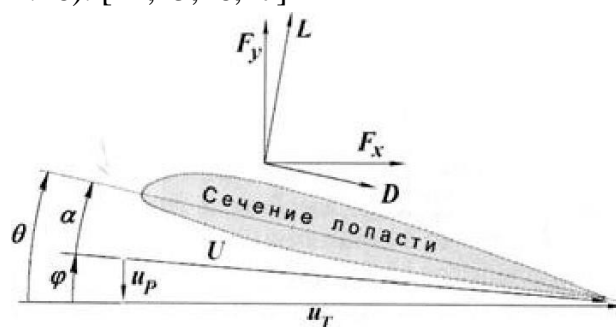


Рисунок 1.16. Скорости и силы, действующие в сечении лопасти

Силы, нормальная и параллельная плоскости диска, выражаются через подъемную силу и сопротивление:

$$F_y = L \cos \varphi - D \sin \varphi; F_x = L \sin \varphi + D \cos \varphi \quad (2)$$

В свою очередь величины L и D определяются по известным формулам:

$$L = \frac{\rho U^2}{2} c * c_L; D = \frac{\rho U^2}{2} c * c_D, \quad (3)$$

где ρ – плотность воздуха, c – длина хорды профиля в рассматриваемом сечении, c_L, c_D – коэффициенты подъемной силы и сопротивления. Для несущих винтов вертолетов на режиме висения отношение λ (называемое иногда коэффициентом протекания) имеет порядок [28]:

$$\lambda = \frac{V + v}{\Omega R} \approx 0.05..0.07 \quad (4)$$

в свою очередь отношение:

$$\frac{u_p}{u_T} = \frac{V + v}{\Omega r} = \frac{\lambda R}{r} \quad (5)$$

Основными силами, действующими на лопасть, являются центробежная сила, вычисляемая интегрированием по длине лопасти (Рисунок 1.17).

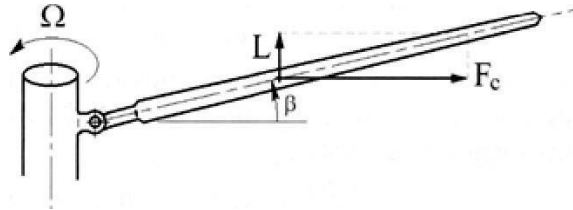


Рисунок 1.17. Образование конуса при вращении винта в режиме висения

Если ось конуса наклонить вперед в сторону желаемого перемещения, то полная тяга уже не будет совпадать с конструктивной осью втулки, а отклонится на некоторый угол (Рисунок 1.18).

Для того, чтобы обеспечить наклон конуса винта вперед, циклический шаг лопастей θ изменяется по углу азимута согласно закону [29,30,31,32]:

$$\theta = \theta_0 - K\beta - \theta_{1c} \cos(\psi - \psi_0) - \theta_{1s}(\psi - \psi_0), \quad (6)$$

где θ_0 - общий шаг, ψ_0 - угол опережения, обеспечивающий отсутствие взаимосвязи продольного и поперечного управления, θ_s, θ_c соответственно продольное и поперечное управления, β - угол взмаха.

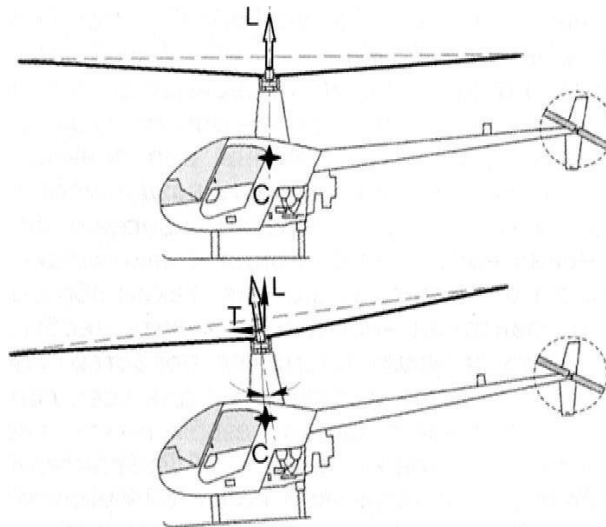


Рисунок 1.18. Возникновение горизонтальной компоненты силы тяги при наклоне конуса, описываемого лопастями

В связи с наклоном конуса меняется и угол атаки лопасти относительно встречного потока воздуха (Рисунок 1.19).



Рисунок 1.19. Изменение углов установки, атаки и взмаха лопасти при управлении циклическим шагом

У лопасти, движущейся навстречу полету, к окружной скорости добавляется зависящая от азимута проекция скорости полета, тогда как у лопасти, движущейся назад эта скорость вычитается (Рисунок 1.20).

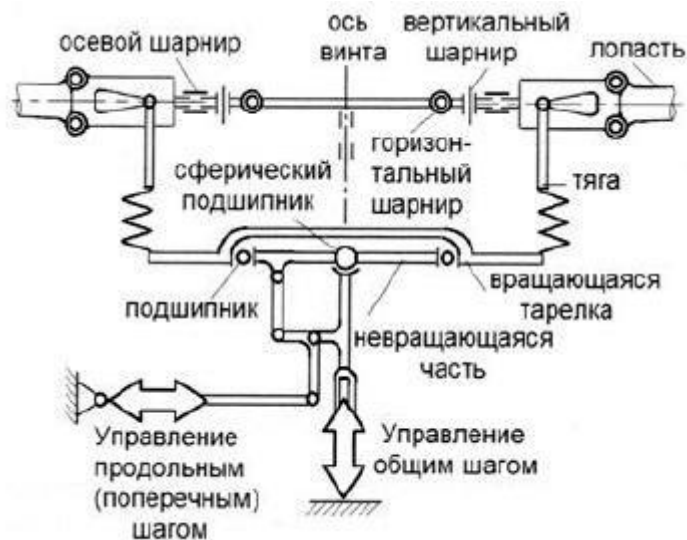


Рисунок 1.20. Схема автомата перекоса

Периодическое возбуждение лопастей приводит к возникновению в них интенсивных изгибнокрутильных колебаний, что накладывает жесткие ограничения на динамические и прочностные характеристики лопастей. Среди всех факторов, вызывающих интенсивные колебания лопастей и конструкции вертолета в целом, отмеченное явление играет очень важную роль.

Другим источником колебаний лопасти является срыв потока, обусловленный турбулентными явлениями, интенсивность которых нарастает по мере приближения скоростей к скорости звука.

1.4 Проблематика резонансных колебаний лопастей винта и методы их исследования

Для расчета изгибных колебаний лопасти воздушного винта схематизируется прямым стержнем, растянутым центробежными силами [33,34,35,36]. Приближенно частота свободных изгибных колебаний ω_n определяется по формуле:

$$\omega_j^2 = \omega_j^2_0 + k_j \Omega^2, \quad (7)$$

где ω_j - частота j-го тона в поле центробежных сил; ω_{j0} - частота свободных колебаний невращающейся лопасти; Ω – угловая скорость вращения;

$$k_1 = \frac{kR \int_{r_0}^R mr \int_{r_0}^r \left(\frac{df_1}{dr}\right)^2 dr^2 - \int_{r_0}^r mf_1^2 \sin^2 \phi_0 dr}{I_d \theta_d^2 + I_b \theta_b^2 + kR \int_{r_0}^r mf_1^2 dr}; \quad (8)$$

$$k_2 = \frac{\int_{r_0}^R mr \int_{r_0}^R \left(\frac{df_2}{dr}\right)^2 dr^2}{\int_r^R mf_2^2 dr}, \quad (9)$$

где k – число лопастей; R – радиус винта; r – текущий радиус; r_0 – радиус заземленного конца; m – погонная масса лопасти; f_1 – форма i-го тона колебаний; ϕ_0 – угол установки; I_d, I_b – соответственно массовые моменты инерции двигателя и винта относительно оси вращения; θ_d, θ_b - соответственно амплитуды углов поворота двигателя и винта при собственных колебаниях винта с формой f_1 .

Собственные формы изгибных колебаний невращающейся лопасти используют для приближенного определения спектра частот лопасти, растянутой при вращении центробежными силами [37],

$$\omega_j^2 = \omega_j^2_0 + k_j \Omega^2, \quad (10)$$

$$k_j = \frac{\int_0^R N_{\omega-1} \left(\frac{df_j}{dr}\right)^2 dr}{\int_0^R mf_j^2 dr}, \quad (11)$$

здесь f_j – собственная форма изгибных колебаний j-го тона невращающейся лопасти; N – растягивающая сила в сечении r ; m – погонная масса вдоль лопасти; R – радиус винта.

При расчете крутильных колебаний лопасти обычно принимают, что лопасти можно рассматривать как абсолютно жесткое тело, упруго прикрепленное к втулке винта на жесткости проводки управления.

Приближенный расчет вынужденных колебаний лопасти сводится к вычислению коэффициентов динамического увеличения амплитуды колебаний λ_d :

$$q_d^j = \lambda_d q_{ст}^j, \quad (12)$$

где $q_d^j, q_{ст}^j$ – собственно динамическая и статическая величины обобщённой координаты q^j ;

$$\lambda_d = \frac{1}{\left[1 - \left(\frac{n}{\omega_j}\right)^2\right]^2 + 4\xi_j^2 \left(\frac{n}{\omega_j}\right)^2}; \quad (13)$$

$$\xi_j = \frac{1}{4} C_y^a \frac{\Omega}{\omega_j} \frac{1}{m_j} \int_0^R br f_j^2 dr, \quad (14)$$

где n – частота вынужденных колебаний; $m_j = \int_0^R m f_j^2 dr$; C_y^a – производная коэффициента подъемной силы по углу атаки; b – хорда лопасти в сечении r .

1.5 Актуальность, новизна темы. Цели и задачи работы

Кроме этого, на основе проведенного анализа информации о производстве и теоретических данных о работе НВ на различных этапах полета вертолета можно сделать вывод о возможности улучшения его конструкции для обеспечения безопасной и надежной работы вертолета. Однако при развитии современных цифровых технологий стало возможно проводить моделирование разнообразных состояний в которых может находиться летательное средство во время эксплуатации [38,39,40,41,42]. Однако, в существующих комплексах программ не реализованы многие из этих возможностей. Разработка методов моделирования взаимодействия деформированных твердых тел с окружающей средой и исследования этих моделей является актуальной, поскольку создает дополнительные возможности при проектировании рациональных и эффективных винтов. Основной проблематикой при создании несущего винта вертолета, винта самолета или пропеллера другого назначения является правильная установка всех элементов конструкции, при этом сама конструкции должна обладать такими свойствами, которые обеспечивали бы гашение колебаний, возникающие при работе конструкции. Если даже вероятность разрушения несущего винта из-за возникающего резонанса в конструкции можно снизить при помощи множества компенсирующих элементов и подвижных частей, то колебания лопастей погасить таким образом не удастся. В настоящее время устранение резонанса, например, на лопастях во время полета вертолета осуществляется за счет изменения геометрических параметров лопасти и применения композиционных материалов, которые повышают прочностные характеристики изделия и понижают его вес. Однако данные способы ухудшают летные характеристики самого летательного аппарата за счет снижения максимально допустимой подъемной силы и скорости полета.

Поэтому **целью** данной работы является разработка моделей динамического взаимодействия упругих лопастей несущего винта с окружающей средой. Для достижения поставленной цели требуется провести разработку программной среды, предназначенной для реализации этой цели, для поиска резонансных частот лопастей

винта и для разработки мер по устранению резонансов. При этом для повышения качества получаемых результатов разработан новый конечный элемент и проведена соответствующая модификация МКЭ.

Тем не менее, ощущается недостаток информации по данному вопросу, так как экспериментальные и теоретические исследования имеют ограниченный характер, а методы расчёта не позволяют в полной мере оценить влияние сложной геометрии лопасти.

Необходимо отметить, что расчёты аэродинамических характеристик всей геометрии винта вертолёта методами CFD в настоящее время требуют значительных затрат вычислительных ресурсов и времени, но при существующих темпах развития вычислительной техники эта проблема исчезнет.

Так как при расчете приближенных или уточненных значений частот вынужденных колебаний рассматривается система абсолютно жесткое тело конечным числом степеней свободы. Это приводит к необходимости проводить большое число вычислений с использованием сложных алгебраических уравнений, которые содержат в себе различные коэффициенты, что приводит к увеличению погрешности между результатами расчета и показаниями во время эксплуатации. Помимо этого, определение резонансных колебаний, оказывающих высокие нагрузки на прочностные характеристики конструкции лопастей и вызывающие микротрещины на их поверхности, невозможно учитывать в течении длительного времени с помощью аналитических уравнений. Другим важным элементом расчета резонансных колебаний лопастей является воздействие вихревых потоков, возникающих в следствие работы воздушного винта. Учетом данного фактора является крайне сложным, а в некоторых ситуациях невозможным. Актуальными техническими задачами в данное время являются: определение резонансных колебаний на упругих элементах конструкций, в следствие их взаимодействия с окружающей средой, определение значений передачи энергии с одного элемента конструкции на другой за счет движения воздушного потока, определение значений перемещения упругих элементов конструкции под воздействием вихревых потоков.

Поэтому разработка метода расчета резонансных колебаний лопастей воздушных винтов при взаимодействии с окружающей средой является **актуальной задачей**. Для этого в рамках данной работы была поставлена **цель** в создании конечно-элементных математических моделей динамического взаимодействия упругих винтов с окружающей газовой или жидкой средой с использованием алгоритма конечно-элементного расчета со встроенными ортогональными финитными функциями, а также программной среды, реализующей эти метод и алгоритм.

Для достижения поставленной цели в работе решаются следующие **задачи**:

1. Анализ современных способов моделирования взаимодействия исследуемого объекта с газовой средой.
2. Разработка математических конечно-элементных моделей динамики взаимодействия лопастей винта с окружающей средой в программном комплексе ANSYS WorkBench.

3. Построение метода конечных элементов двухмерной и трехмерной теории упругости с использованием ортогональных финитных функций и соответствующих конечно-элементных моделей трехмерных упругих элементов конструкций.

4. Разработка программного комплекса конечно-элементного анализа, учитывающей его взаимодействие с внешними программными библиотеками решателей систем алгебраических уравнений, с существующими препроцессорами и с комплексом ANSYS.

5. Применение созданного программного продукта для решения актуальных технических задач взаимодействия вращающихся упругих винтов с окружающей средой.

ГЛАВА 2. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ УПРУГИХ ВРАЩАЮЩИХСЯ ЛОПАСТЕЙ ВИНТА С ВОЗДУШНОЙ СРЕДОЙ В КОМПЛЕКСЕ ПРОГРАММ ANSYS. КОНТИНУАЛЬНАЯ МОДЕЛЬ И ДИСКРЕТНАЯ МОДЕЛИ

Все функции, выполняемые программным комплексом ANSYS, объединены в группы, которые называются процессорами. Программа имеет один препроцессор, один процессор решения, два постпроцессора и несколько вспомогательных процессоров, включая оптимизатор [43,44,45].

Препроцессор используется для создания конечно-элементной модели и выбора опций для выполнения процесса решения. Процессор решения используется для приложения нагрузок и граничных условий, а затем для определения решения. С помощью постпроцессора можно обращаться к результатам решения для их оценки, а также для планирования проведения дополнительных вычислений. Анализ, который проводится, состоит из трех стадий: препроцессорная подготовка, получение решения и постпроцессорная обработка.

2.1 Математическая модель лопасти винта вертолета

В данной работе на примере винта вертолета рассматриваются резонансные колебания, которые могут привести к возникновению повреждений или полному выходу из строя лопастей винта в процессе их работы. Колебания различной физической природы имеют много общих закономерностей и тесно взаимосвязаны с волнами. Аналогичным образом строятся модели винтов другого назначения.

При расчете взаимодействия винта и газа происходит обмен результатами расчетов между двумя типами сетки (для газовой среды и для упругого тела), с этой целью используется контактное тело. Контактное тело представляет собой поверхность, полностью повторяющую поверхность профиля упругого тела. Помимо физических изменений поверхностное контактное тело позволяет добавлять авторский конечный элемент с ортогональными финитными функциями, структура и разработка этого конечного элемента рассматривается в главе 3, внедрение в контактное тело в главе 4.

Контактное тело является поверхностью или двумерным телом, следовательно, для него применяется уравнение равновесия в виде[32]:

$$Eh \left[\frac{1}{1-\sigma^2} \frac{\partial^2 u_x}{\partial x^2} + \frac{1}{2(1+\sigma)} \frac{\partial^2 u_x}{\partial y^2} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_y}{\partial x \partial y} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_z}{\partial z \partial x} \right] + P_x = 0, \quad (15)$$

$$Eh \left[\frac{1}{1-\sigma^2} \frac{\partial^2 u_y}{\partial y^2} + \frac{1}{2(1+\sigma)} \frac{\partial^2 u_y}{\partial x^2} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_x}{\partial x \partial y} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_z}{\partial z \partial y} \right] + P_y = 0, \quad (16)$$

$$Eh \left[\frac{1}{1-\sigma^2} \frac{\partial^2 u_z}{\partial y^2} + \frac{1}{2(1+\sigma)} \frac{\partial^2 u_z}{\partial x^2} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_x}{\partial z \partial y} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_y}{\partial z \partial x} \right] + P_y = 0, \quad (17)$$

где P_y, P_x, P_z – компоненты внешней объемной силы.

Уравнения колебаний упругих тел с подстановкой в уравнение равновесия сил инерции $- \rho h \frac{\partial^2 u_x}{\partial t^2}$, $- \rho h \frac{\partial^2 u_y}{\partial t^2}$ и $- \rho h \frac{\partial^2 u_z}{\partial t^2}$ вместо P_y, P_x, P_z :

$$\frac{\rho}{E} \frac{\partial^2 u_x}{\partial t^2} = \frac{1}{1-\sigma^2} \frac{\partial^2 u_x}{\partial x^2} + \frac{1}{2(1+\sigma)} \frac{\partial^2 u_x}{\partial y^2} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_y}{\partial x \partial y} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_z}{\partial z \partial x}, \quad (18)$$

$$\frac{\rho}{E} \frac{\partial^2 u_y}{\partial t^2} = \frac{1}{1-\sigma^2} \frac{\partial^2 u_y}{\partial y^2} + \frac{1}{2(1+\sigma)} \frac{\partial^2 u_y}{\partial x^2} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_x}{\partial x \partial y} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_z}{\partial z \partial y}, \quad (19)$$

$$\frac{\rho}{E} \frac{\partial^2 u_z}{\partial t^2} = \frac{1}{1-\sigma^2} \frac{\partial^2 u_z}{\partial y^2} + \frac{1}{2(1+\sigma)} \frac{\partial^2 u_z}{\partial x^2} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_x}{\partial z \partial y} + \frac{1}{2(1-\sigma)} \frac{\partial^2 u_y}{\partial z \partial x}, \quad (20)$$

Эти уравнения описывают продольные колебания в упругом теле и будут применяться для передачи внешних нагрузок, создаваемых движением воздушных масс и потоков при их взаимодействии с упругим телом. Способы применения будут даны в главе 4.

2.1.1 Форматы различных САД-систем в модуле Geometry

Программный комплекс ANSYS Workbench по сравнению со старыми своими версиями приобретает новые инструменты для создания цифровых моделей внутри себя. Однако обзор рынка систем ориентированных на создание изделий, обладающих сложным профилем, или конструкций, которые содержат в себе тысячи мелких компонентов, порождает вопрос о создании модели для проведения расчета уже во внешнем программном продукте[46,47,48,49].

Существует огромное множество различных расширений файлов, которые содержат всю информацию о модели, но программный комплекс ANSYS, в частности, его составной элемент, отвечающий за работу с геометрией, может не распознать ту или иную геометрию в этих файлах. Поэтому после загрузки можно будет наблюдать различные дефекты в профиле модели, например, исчезновение некоторых плоскостей, появление лишних линий и ребер и другие дефекты. Также следует определять какую именно модель следует загружать твердотельную или полигональную. Полигональная модель представляет собой пустотелый объект, обладающий только поверхностным профилем и при этом внутри него, имеется пустота. Для некоторых типов задач достаточно и полигональной модели, но чаще требуется определить температурные и прочностные характеристики изделия внутри нее.

В методе, который будет расписан в тексте, были проведены эксперименты с импортом наиболее часто встречающихся расширений. В основном это форматы: *stp*, *step*, *stl*, *igs*, *iges*, *sldas*, *sldprt*, *x_t*, *x_b*, *catpart* [50,51,52].

Различие этих форматов заключается в программных продуктах, в которых они были созданы. В частности, в их структуре, в которую заносится вся информация о создаваемой модели. Практически все форматы, кроме *stl*, являются представителями твердотельной геометрии. Эти форматы имеют внутри себя большое количество строк кода, в который содержит информацию о расположении всех точек модели в пространстве, их взаимосвязь между собой и со всеми элементами конструкции в целом. Однако, такие форматы как *igs* и *iges* могут содержать полигональную модель, если конструктор в процессе создания модели принял решение экспортировать только профиль изделия, в дальнейшем это приводит к большим трудностям для проведения полноценного расчета в программном комплексе.

Наиболее подходящим для работы являются форматы *stp* и *step*, данные форматы первоначально создавались как обладатели полноценной информации о том изделии или конструкции, которую конструктор хотел передать. Но полноценная информация об изделии приводит к большим емкостным ресурсам и к увеличению времени импорта. Встроенная CAD-система программного комплекса ANSYS способна загрузить и обработать файлы данного формата в автоматическом режиме, исправляя возможные дефекты геометрии.

2.1.2 Возможности редактирования цифровой модели

В данной работе будет рассматриваться версия 19 ANSYS и встроенная CAD-система Design Geometry. На момент выхода этой версии была добавлена новая CAD-система SpaceClaim Geometry, однако цель работы не заключается в использовании современных средств проектирования для создания высокоточных изделий, поэтому будет рассматриваться старая версия этой системы.

Модуль Geometry представляет собой набор основных средств проектирования: функции выдавить и вращать, булевские операции объединения, вычитания, пересечения и другие. Кроме основных операций и функций данный модуль обладает возможностью создавать новые рабочие плоскости и эскизы. А для создания самих изделий модуль обладает примитивными элементами создания: точки, линии, окружности, прямоугольники и другими элементами.

2.1.3 Построение сетки конечных элементов в лопастей винта

По завершению этапа проектирования и устранения дефектов импорта исследуемой цифровой модели в модуль Geometry, требуется установить связь этого модуля с модулем построения сетки Mechanical Mesh.

Данный модуль отвечает за построение объемной сетки на телах, которые будут подвергаться различным механическим нагрузкам в процессе их исследования. Так как

исследуемым объектом являются лопасти винта вертолета, состоящие из ряда различных материалов как снаружи, так и внутри конструкции, то поэтому при моделировании используются композиционные материалы. В данном модуле на основе применения языка программирования APDL возможно использовать один тип конечного элемента, присваивая различные свойства материалов в различных частях конструкции. Так как таких частей в лопастях большое число, было принято решение внедрить автоматизированный способ привязки свойств материалов, более подробно об этом написано в главе 4.

Методика построения сетки рассматривается здесь на примере ротора винта (с лопастями) легкого двухместного вертолета. Геометрия винта представлена на Рисунок 2.1.

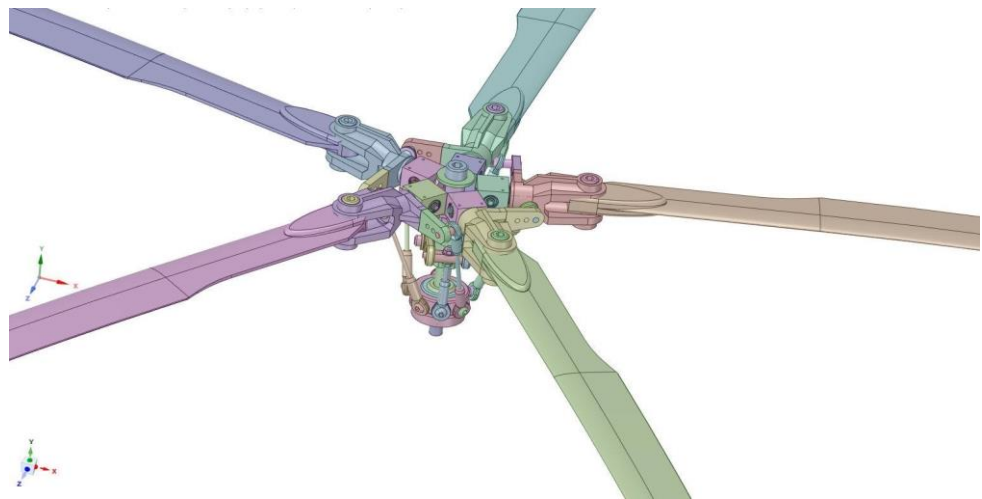


Рисунок 2.1. Общий вид конструкции винта с лопастями в модуле проектирования Ansys Workbench

Наложение сетки связано с выбором некоторых параметров, при описании методики большинство таких параметров будут иметь статус “стандартные” или “на выбор программы”. При применении разработанной программной среды каждый параметр будет определяться на основе вводимых пользователем значений в автоматизированном процессе. Для описания методики была построена сетка конечных элементов винта (Рисунок 2.2), размеры этих элементов учитываются при наложении сетки на воздушную область.

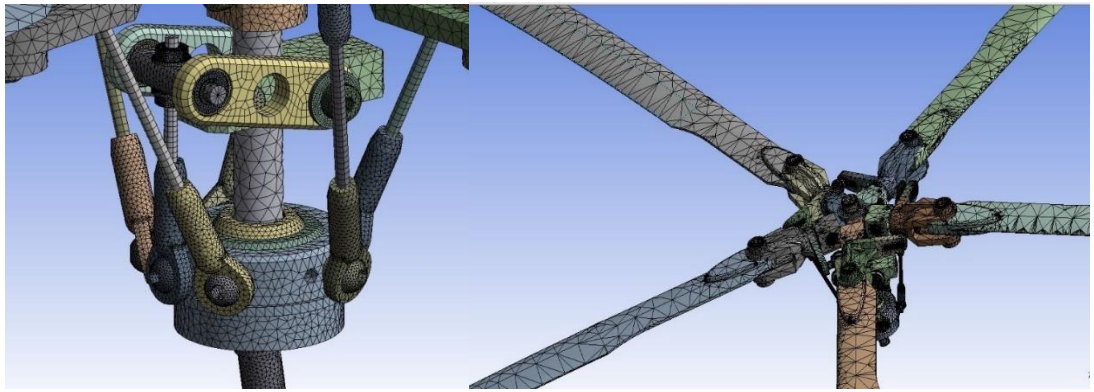


Рисунок 2.2. Результат наложения сетки конечных элементов на исследуемый винт с лопастями

Для создания такой сетки ползунок окна интерфейса, отвечающий за качество создаваемых конечных элементов, был установлен в максимальное положение для получения более густой сетки в элементах, имеющих маленькие размеры, при этом само качество элементов становится выше и снижается вероятность возникновения ошибок в последующем процессе расчета (Рисунок 2.3).

Details of "Mesh"	
Display	
Display Style	Body Color
Defaults	
Physics Preference	Mechanical
<input type="checkbox"/> Relevance	100
Element Order	Program Controlled
Sizing	
Size Function	Adaptive
Relevance Center	Fine
<input type="checkbox"/> Element Size	Default
Mesh Defeaturing	Yes
<input type="checkbox"/> Defeature Size	Default
Transition	Fast
Initial Size Seed	Assembly
Span Angle Center	Fine
Bounding Box Diagonal	961,870 mm
Average Surface Area	27,2250 mm ²
Minimum Edge Length	1,9259e-003 mm
Quality	
Check Mesh Quality	Yes, Errors
Error Limits	Standard Mechanical
<input type="checkbox"/> Target Quality	Default (0.050000)
Smoothing	Medium
Mesh Metric	Element Quality
<input type="checkbox"/> Min	3,8521e-003
<input type="checkbox"/> Max	1,
<input type="checkbox"/> Average	0,67613
<input type="checkbox"/> Standard Deviation	0,20071
Inflation	
Advanced	
Statistics	

Рисунок 2.3. Настройки для наложения сетки конечных элементов на исследуемую модель

Указанные на Рисунок 2.3 параметры - стандартные, для повышения точности получаемых результатов расчетов на определенных элементах конструкции необходимо регулировать большинство из них. Более подробно описание этих параметров и способы их изменения с помощью программной среды будут даны в главе 4.

Кроме этого, для сокращения времени расчета, снижения требований для вычислительной техники и сохранения точности получаемых результатов, применяется

методика наложения сетки конечных элементов с ортогональными финитными функциям. Данная методика основывается на наложении на поверхностях цифровых моделей сетку стандартных параметров, однако внутри моделей накладывается параллелепипедная сетка с заданными параметрами на языке APDL

Для введения собственных параметров с целью внедрения нового конечного элемента следует при выделении цифрового тела, на которое будет накладываться эта сетка, добавить возможность считывания параллельного цикла команды, записанных на языке APDL (Рисунок 2.4).

```

Commands
! Commands inserted into this file will be execu
! The material number for this body is equal to

! Active UNIT system in Workbench when this obje
! NOTE: Any data that requires units (such as m
! See Solving Units in the help sys

/Prep7
/UIS,MSGPOF,3
!ELEMENT TYPES
ET,1,USER300
USRELEM,8,3,BRICK,0,20,50,0,4,0,0
USRDOF,DEFINE,UX,UY,UZ
KEYOPT,2,1,6
!MATERIAL PROPERTIES
MP,EX,1,200e9
MP,PRXY,1,0.3
VMESH,1,1,2,3,4
EMODIF,ALL,TYPE,1,
ALLS
EMODIF,ALL,MAT,1,
ACLEAR,1
ALLS
/VIEW,1,,-1
/REP,FAST
/SOLU
ANTYPE,0
SOLCONTROL,OFF
SOLVE

```

Рисунок 2.4. Код для применения на цифровую модель нового конечного элемента с ортогональными финитными функциями

2.1.4 Определение частот и форм свободных колебаний лопастей винта вертолета

Для определения частот свободных колебаний упругих лопастей винта в программном комплексе Ansys Workbench существует модуль Modal, который способен проводить расчет более ста частот и форм свободных колебаний в заданном диапазоне с последующей демонстрацией деформаций и напряжений, соответствующих каждой моде.

Данный модуль в разработанной методике расчета резонансных колебаний не содержит необходимых функций и процедур:

- Использование нового конечного элемента с ОФФ.
- Формирование таблицы частот свободных колебаний и форм для каждой лопасти винта.
- Создание файла с заданной структурой для хранения и передачи результатов расчета в следующий модуль.

Для реализации метода был разработан на основе модуля Modal модуль ModalTable, который уже имеет все необходимые функции и процедуры работы с использованием стандартных средств вычисления (Рисунок 2.5).



Рисунок 2.5. Модуль расчета, хранения и передачи частот свободных колебаний каждой лопасти винта.

При подготовке расчета считается, что вся конструкция ротора с валом винта является полностью жесткой, колебаниям подвержены только сами лопасти. Диапазон поиска частот и форм устанавливается без ограничений и указывается количество частот и форм равным шести. По завершению расчета оказалось, что Ansys демонстрирует колебания каждой лопасти, при этом различные частоты колебаний могут отличаться на тысячные и сотые доли герца (Рисунок 2.6).

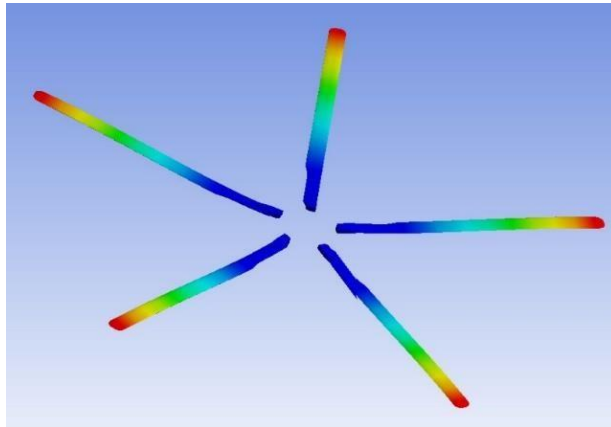


Рисунок 2.6. Пример продольных колебаний лопастей винта с жестким ротором

Помимо визуального отображения результатов расчета частот свободных колебаний элементов конструкции, есть возможность рассмотреть таблицу, которая содержит все полученные частоты (Рисунок 2.7).

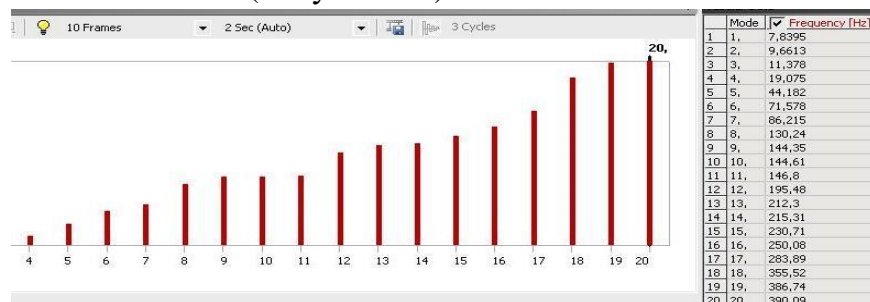


Рисунок 2.7. Таблица частот свободных колебаний лопастей винта вертолета

Для дальнейшего использования информации о частотах свободных колебаний каждой лопасти винта, а также форм этих колебаний, требуется сохранить данные о результатах расчета в отдельный файл, структура и алгоритм формирования которого определяются с помощью внедрения дополнительной функции, которая размещается на панели управления модуля ModalTable (Рисунок 2.8).

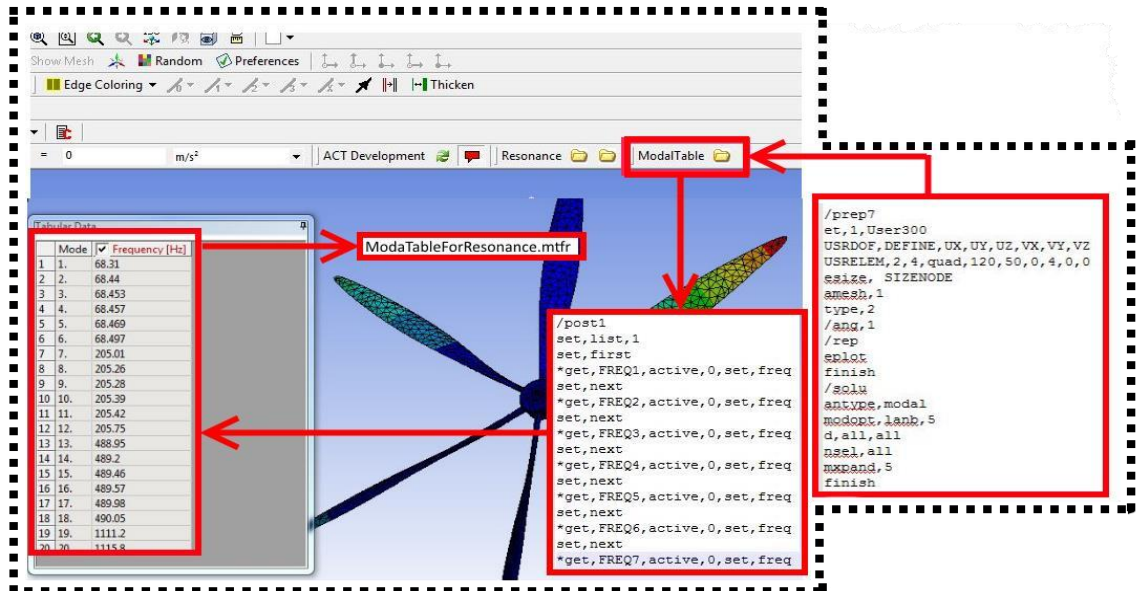


Рисунок 2.8. Алгоритм работы дополнительных функций и процедур в модуле ModalTable

Функция для сохранения в файл ModalTableForResonance.mtdr имеет название ModalTable, она добавляется в проект расчета и представляет собой набор команд на языке APDL для автоматического внедрения конечного элемента с ОФФ и сохранения результатов в необходимый формат с целью дальнейшего использования.

2.2 Математическая модель воздушной среды

При использовании стандартных средств расчета аэродинамики внутри программного комплекса ANSYS в процессе движения твердого тела расчет производится с учетом применения модели идеального газа, но только на малых расстояниях от тела.

Что является неверным при расчете движения лопастей винта в воздушном пространстве на высоких угловых скоростях поворота. Возмущения, создаваемые движением лопастей, приводят к возникновению турбулентных потоков, или вихрей, которые могут возникать в отдалении от тела и в течении времени оказывать на него физическое воздействие. Кроме этого, следует учитывать, что возмущения, создаваемые одной лопастью, приводят к внешнему воздействию на последующие движущиеся лопасти. При использовании стандартных средств возмущения рассеиваются в пространстве, описанном уравнениями идеального газа.

Для изменения стандартных настроек расчета уравнений идеального газа необходимо в ручном режиме прописать уравнение движения вязкого газа, при этом пренебрегая сжимаемостью за счет движения, а также считать вязкость постоянной, то есть записать уравнение Навье-Стокса в виде [53,54,55]:

$$\frac{dw}{dt} = \frac{\partial w}{\partial t} + w_x \frac{\partial w}{\partial x} + w_y \frac{\partial w}{\partial y} + w_z \frac{\partial w}{\partial z} = -\frac{1}{\rho} \text{grad } P + \nu \nabla^2 w \quad (23)$$

где P – общее давление, ρ – плотность, $\nu = \mu/\rho$ – коэффициент кинематической вязкости, μ – коэффициент динамической вязкости, w_x, w_y, w_z – проекция скорости воздушного потока на соответствующие оси системы координат.

К уравнению движения среды необходимо добавить уравнение неразрывности:

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho w) = 0 \quad (24)$$

Данные формулы будут использоваться непосредственно при постановке задачи в вычислительном модуле аэродинамики CFX Ansys.

2.2.1 Определение в модуле Geometry размеров и формы области, занимаемой воздушной средой

Форма и размеры окружающего пространства определяются типом задачи, которую требуется решить. Если исследуемый объект располагается внутри аэродинамической трубы, то окружающая среда представляет собой цилиндр, обладающий тремя плоскостями: входом (inlet), выходом (outlet) и стенкой или открытой плоскостью (wall, opening) в зависимости от вида аэродинамической трубы (Рисунок 2.9).

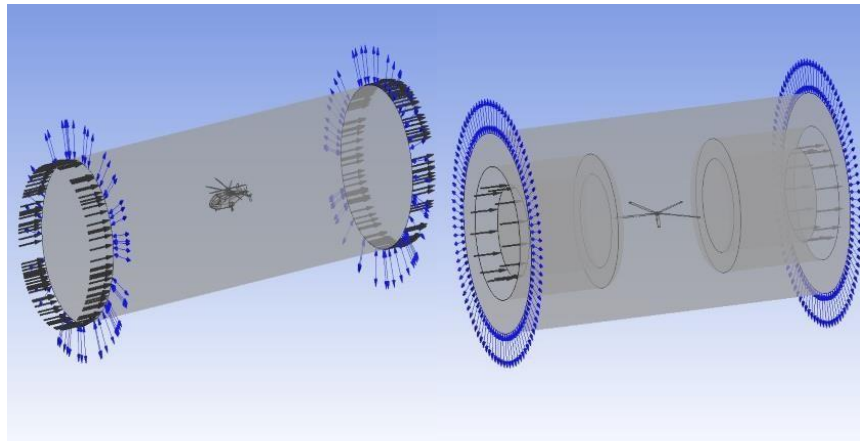


Рисунок 2.9. Воздушные области, в виде аэродинамической трубы, с исследуемым телом внутри

Однако часто бывают случаи, когда требуется определить, как поведет себя объект в воздушном пространстве на определенной высоте при определенных температуре и влажности. В этом случае, если требуется использовать параллелепипедную форму области окружающего пространства, размеры которой основываются на наблюдениях за поведением воздушного потока на различных расстояниях от движущегося объекта, создающего возмущения воздуха.

Если рассматривается задача о движении объекта в открытом пространстве, то в этом случае рекомендуется использовать параллелепипед, у которого две стороны будут являться входом и выходом, а остальные могут быть как непроницаемыми, так и открытыми, стенками (Рисунок 2.10).

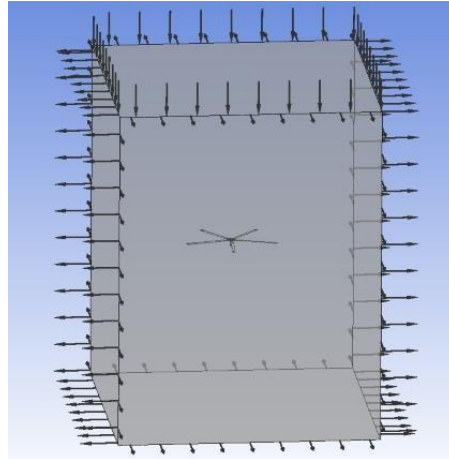


Рисунок 2.10. Воздушное пространство с исследуемым телом, в виде параллелепипеда

Под открытыми стенками понимаются плоскости, на которые будет накладываться сетка, описывающая прохождение через нее возмущений от объекта. Данный тип сетки используется также в том случае, если требуется провести исследование без влияния отраженного возмущения.

2.2.2 Построение и перестройка сетки конечных элементов

Построение сетки конечных элементов связано с выполнением нескольких требований, учитывающих характеристики решаемой задачи.

Предположим, исследуемый объект — это параллелепипед, у которого длина 1 метр, ширина и высота по 50 сантиметров, а окружающая область представляет собой цилиндр диаметром 5 метров и высотой 10 метров. При этом исследуемый объект располагается в центре окружающей среды.

Создается такая сетка конечных элементов, которая позволяет за небольшой промежуток времени (в зависимости от задачи) получить решение.

Обычно при решении задач, как описывалось в первой главе, используются два метода решения, во-первых, метод двумерного решения в определенном сечении объекта и метод вычитания объекта из исследуемой области с дальнейшим наблюдением линий тока обтекания его воздушным потоком. При этом отсутствуют какие-либо возможности задать движение объекту внутри исследуемой области, не прибегая к крайне сложным способам, которые заключаются в проектировании дополнительных окружающих областей внутри основной и задании вращения для них.

Разрабатываемая методика построения сетки заключается в создании двух различных видов сетки, каждая из которых накладывается на исследуемый объект и на окружающую его среду. Основным изменением по сравнению со стандартным алгоритмом наложения сетки, является её перестройка в результате деформации лопасти и вследствие этого изменение размеров конечных элементов для сохранения точности результатов в течении всего времени вращения лопасти.

При этом перестройка сетки осуществляется на уровне кода, то есть требуется сформировать алгоритм перезаписи файла с содержанием сетки воздушного пространства и расположения исследуемого тела.

Оба объекта соединяются в вычислительном модуле, сетки согласуются друг с другом. Новизна предлагаемой методики заключается в объединение двух видов сетки механической (деформированной) и аэрогидродинамической. Использование и применение данной возможности более подробно будет рассмотрено в главе 4.

Зная размеры и формы исследуемого объекта и окружающей среды, необходимо подобрать определенные параметры сетки. В методике используются два способа: применение тетраэдров с созданием слоев по граням окружающей среды для повышения точности результатов вблизи границы и применение параллелепипедных элементов сетки, но с важным условием, чтобы размеры элементов были существенно меньше размеров исследуемого объекта.

В том случае, если размеры элемента сетки будут больше или равны размерам исследуемого объекта, при расчете элемент не будет воспринимать ту часть объекта, которая оказалась внутри него и по результатам расчета можно будет наблюдать как линии тока проходят сквозь объект и полностью его игнорируют.

Кроме этого, при воздействии внешних сил на лопасти они подвергаются деформации в каждый момент времени, определяемый пользователем. Это означает то, что требуется изменять сеточную модель исследуемого тела на основании получаемых перемещений, подгружать в воздушное пространство новую модель и проводить следующий шаг по времени. Данный метод позволяет на каждом шаге сохранять значения напряжений и деформаций исследуемого тела в отдельный файл, который затем используется для объединения с файлом частот свободных колебаний и обнаружения резонансных колебаний тела. Это возможно осуществить благодаря применению языков программирования Python и XML, однако само применение языка не способствует адекватной работе метода. Для правильности работы метода требуется разработка дополнительных вычислительных модулей в программном комплексе ANSYS WorkBench, в которых уже осуществлено внедрение необходимых программных кодов для работы метода.

2.3 Континуальная модель взаимодействия вращающихся упругих лопастей винта с воздушной средой (связанная задача)

В работе строится континуальная модель динамического взаимодействия лопастей винта с воздушной средой, данном случае рассматривается винт легкого

двухместного вертолета, однако разработанная программная среда для автоматизированного процесса дискретизации моделей и установки граничных условий с выводом результатов решения может быть применена и для других видов винтов, в том числе и для гребных винтов в судостроении.

В рассматриваемой задаче используются три цифровые модели: винта с лопастями, воздушной среды и контактного тела. Добавление и дискретизация контактного тела осуществляется в автоматизированном процессе внутри созданной программной среды, более подробно это изложено в 4 главе. В дальнейшем будут показаны математические модели двух остальных объектов и общая континуальная модель их взаимодействия.

Математическая модель, характеризующая напряженно-деформированное состояние винта, содержит:

- тензорное уравнение движения (три скалярных уравнения)

$$\rho \left(\frac{d^2 \bar{u}}{dt^2} - \bar{F} \right) = \text{div} \bar{\sigma} \quad (21)$$

- шесть кинематических уравнений, связывающих компоненты ε_{ij} тензора деформации с компонентами u_i вектора перемещений

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i} \frac{\partial u_k}{\partial x_j} \right) \quad (22)$$

- закон Гука в тензорной форме (шесть скалярных уравнений)

$$\bar{\sigma} = \lambda I_1(\bar{\varepsilon}) \bar{g} + 2\mu \quad (23).$$

Для винта задаются скорость его вращения, сила его веса, возникающие при вращении центробежные силы, а также задаются силовые и кинематические граничные условия сопряжения поверхности винта с воздухом (Рисунок 2.11).

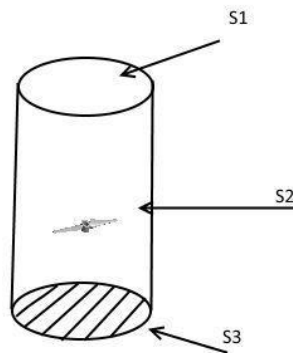


Рисунок 2.11. Постановка задачи с вращением винта в воздушном пространстве на заданной высоте над землей

На части поверхности S_2 трехмерной области, занимаемой воздухом и находящимся в нем винтом, задаются в рамках данной модели граничные условия непроницаемости, наложенные на вектор скорости частиц воздуха: $V \cdot \vec{n}|_{S_2} = 0$ (\vec{n} – внешняя нормаль к S_2). Такие же условия ставятся и на части поверхности S_3 , моделирующей участок поверхности Земли. На части поверхности S_1 задается постоянное атмосферное давление: $P|_{S_1} = P$. При этом считается, что возмущения воздуха, создаваемые вращением винта, не успевают достигнуть S_2 и, следовательно, граничные условия там со временем не изменяются.

2.4 Дискретизация континуальной модели средствами метода конечных элементов в комплексе программ ANSYS

Метод конечных элементов (МКЭ) – основной метод решения начальнокраевых задач, лежащий в основе большинства современных программных комплексов, предназначенных для выполнения расчетов состояния различных конструкций. Последние десятилетия особенно характерны для развития и применения МКЭ в таких областях механики сплошных сред, как оптимальное проектирование, учет нелинейного поведения, динамика конструкций и тому подобное.

Метод конечных элементов основан на представлении реальной континуальной конструкции ее дискретной моделью и замене дифференциальных уравнений, описывающих НДС сплошных тел, системой алгебраических уравнений. Вместе с тем МКЭ допускает ясную геометрическую, конструктивную и физическую интерпретацию.

Суть метода заключается в том, что область, занимаемая конструкцией, разбивается на некоторое число малых, но конечных по размерам подобластей.

Последние носят название конечных элементов, а сам процесс разбиения – дискретизацией. Процесс дискретизации в программном комплексе ANSYS проходит в отдельных модулях, ориентированных на выполнение этой операции.

Рассмотрим подготовленный проект, предназначенный для расчета динамики взаимодействия лопастей винта вертолета с окружающей средой (Рисунок 2.12).

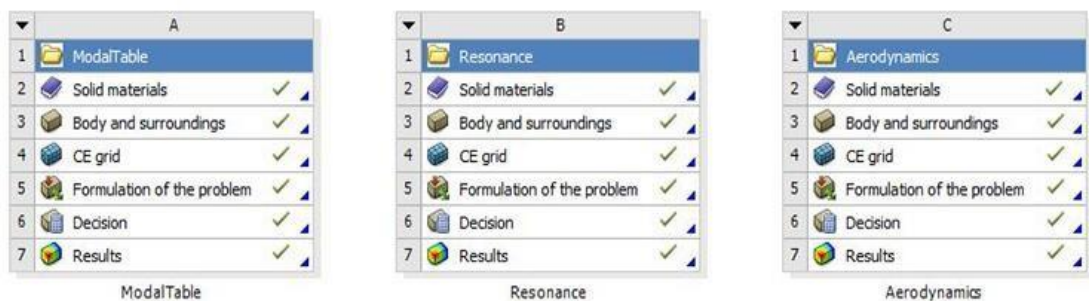


Рисунок 2.12. Проект расчета динамики взаимодействия лопастей винта с окружающей средой в программном комплексе ANSYS WorkBench

Первая дискретизация модели происходит в модуле ModalTable, предназначенном для расчета собственных колебаний конструкции. В данном модуле

дискретизации подвергаются твердотельные компоненты общей модели, окружающая среда проходит дискретизацию в соседнем модуле Aerodynamics. Дискретизация позволяет настраивать параметры наложения сетки индивидуально для каждой модели, в том числе проводить настройки самого конечного элемента.

2.5 Постановка граничных условий в задаче о взаимодействии лопастей винта с воздушной средой

2.5.1 Загрузка цифровых моделей лопастей и воздуха в модуле CFX и проверка их совместимости

Загрузка осуществляется через установку связи между двумя модулями наложения сетки конечных элементов и между модулем Aerodynamics. При установке связи требуется провести обновления созданных сеток для импорта их в следующий модуль.

Под проверкой подразумевается визуальная проверка правильно импортированной сетки, однако саму сетку в модуле Aerodyna без активации определенных параметров увидеть нельзя, но если присутствуют дефекты импорта, то видны пробелы в профиле геометрии. Исправление обнаруженных дефектов возможно только в модулях, в которых была наложена сетка конечных элементов. Существуют скрытые возможности на уровне кода на языке программирования Python, которые позволяют, не устанавливая связи и перед передачей сетки с модулем CFX, провести подобную проверку и выдать рекомендации или провести автоматическую оптимизацию сетки. Данный способ описан в главе 4.

Перед тем как приступить к заданию граничных условий для решения поставленной задачи, требуется определить типы областей, на которых задаются различные граничные условия. Для этого существует пункт Domain Type (Рисунок 2.13).

Пункт «Domain Type» отвечает за свойства материала области в проводимом расчете. Это может быть «Fluid Domain» (жидкая или газовая область), «Solid Domain» (абсолютно твердая область или объект, во многих CAE-системах также используется обозначение «Rigid»), «Porous Domain» (объект, обладающий свойствами пористого материала, или объекта с большим числом отверстий малого диаметра) и «Immersed Solid» (помещенный внутри какой-либо другой области объект). В зависимости от выбранного свойства материала становятся доступны новые параметры для дальнейшего определения модели. Для воздушной области устанавливается тип «Fluid Domain», а для винта с лопастями, так как эта модель расположена внутри воздушного пространства, то «Immersed Solid». Однако для получения полноценного влияния воздушных масс на твердое тело требуется указывать тип Solid, который позволяет устанавливать двустороннюю связь между модулем Aerodynamics и модулем Resonance, в котором происходит наложение частот и форм свободных колебаний на перемещение лопастей под действием воздушных потоков в течении всего времени вращения винта.

Объединение двух этих типов сделало возможным после внедрения авторского кода в модуль расчета

Aerodynamics вычислительными средствами программного комплекса Ansys проводить решение поставленной задачи с переносом информации от одного типа и модуля в другой.

2.5.2 Задание граничных условий на внешней границе воздушной среды

Для определения воздушной области требуется указать основные граничные условия, такие как Inlet, Outlet и Wall, или Opening. Рассмотрим подробнее каждый из условий:

1. Inlet – это поверхность, которая относится к входным, то есть оно отвечает за граничные условия для области, в которой проводится расчет. Входными условиями могут быть скорость водного или воздушного потоков, давление, создаваемое на указанной поверхности, объем потока, проходящего через заданное сечение, или перемещение по координатам.

2. Outlet – поверхность, характеризующая выход потока из замкнутой области. Имеется возможность установить выходные параметры на выбранной поверхности.

3. Wall – поверхность, являющаяся непроницаемой, полностью исключаяющей любое проникновение вещества через нее. Это условие имеет ряд преимуществ перед остальными, может быть задано перемещение такой поверхности, ее деформация во времени, или какие-либо другие ее изменения под действием внешних или внутренних сил, например, под действием достаточно горячего потока газа или воды изменение объекта за счет создаваемого внутри него давления в результате нагрева.

4. Opening – по своим свойствам эта поверхность имеет некоторое сходство с Outlet и Wall, но имеет и отличия. По отношению к Outlet, она позволяет указывать все параметры выхода системы из расчета, но при достижении значений этих параметров расчет задачи будет продолжен и не остановится. Это условие выполняет функцию «двери», оно пропускает потоки воздуха или жидкости по заданным свойствам на поверхности, если значение потока не достигает этого свойства, то Opening срабатывает в качестве Wall и поток не сможет пройти через поверхность, в ином случае срабатывает функция открытия «дверцы» и потоки выходят «наружу» и в дальнейшем уже не участвуют в расчете, это происходит и в том случае, если поток превосходит те значения, которые были установлены на поверхности.

В соответствии с описанной континуальной моделью динамического взаимодействия лопастей винта с воздушной средой определяются модели внутри программного комплекса Ansys Workbench. Определение осуществляется в созданном модуле аэродинамики Aerodynamics на базе модуля CFX, при переносе сетки конечных элементов исследуемого объекта и окружающей области обе сетки программной считываются как единое целое (Рисунок 2.13).

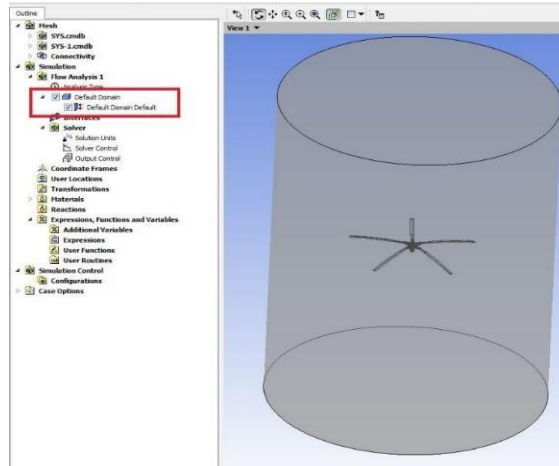


Рисунок 2.13. Загруженные цифровые модели в модуле Aerodynamics

Перед началом задания граничных условий необходимо определить типы применяемых в задаче моделей. Обычно исследуемая модель представляет собой сложную твердую деформируемую конструкцию, содержащую в себе сотни элементов. При задании ее типа могут возникнуть трудности на этапе выбора компонентов сетки, так как при импорте сетки конечных элементов в Aerodynamics им задаются программные имена формата $B_{\text{номер_элемента}}.\text{номер_подэлемента}$, что усложняет процесс разбиения. Однако, по причине того, что сетки накладывались в различных модулях и импортировались также независимо друг от друга, то вместе с импортом происходит загрузка цельных сборок формата $Assembly_{\text{номер_сборки}}$ (Рисунок 2.14).

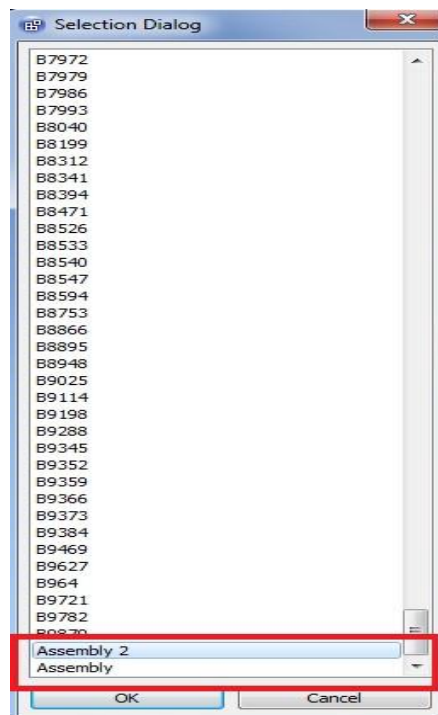


Рисунок 2.14. Указание типа объекта через сборку.

По окончании процесса разбиения в дереве компонентов решаемой задачи будут расположены два Domain'a, которые обозначают, что в задаче участвуют два объекта с заданным типом (Рисунок 2.15).

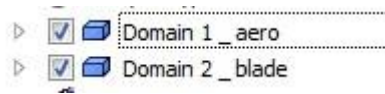


Рисунок 2.15. Созданные объекты в дереве компонентов.

После этого в каждом типе создаются граничные условия, которые будут определять, как будут вести и взаимодействовать между собой эти объекты (Рисунок 2.16).



Рисунок 2.16. Заданные граничные условия для каждого типа объекта

Важным отличием является определение исследуемого тела как тип Solid, но с сохранением некоторых свойств типа Immersed Domain. Это необходимо сделать для того, чтобы решатель при расчете перемещения элементов тела определял его как помещенное в газовое пространство с дополнительными свойствами, в данном случае с указанием уравнения (23), на всё пространство вязкого газа, в котором будет происходить перемещение тела.

2.5.3 Установка параметров проведения расчета, форматов вывода результатов, а также точек мониторинга параметров процессов

Так как требуется определить влияние воздушных масс на лопасти винта в процессе его вращения в некотором интервале времени, то необходимо определить тип решаемой задачи. По стандартным настройкам модуля Aerodynamics используется Steady State (установившийся поток) – тип анализа решаемой задачи. С целью получения адекватных результатов необходимо изменить данный тип на Transient (переходный динамический процесс) и указать в течение какого промежутка времени будет вращаться винт и на каких шагах по времени требуется сохранять результаты расчета в процессе решения задачи.

Перед началом расчета требуется сформировать перечень переменных, которые будут сохраняться в процессе расчета и считываться написанным кодом в отдельный файл для дальнейшего использования в модуле для определения резонанса. Нам необходимо сохранять информацию по перемещению лопастей винта под воздействием воздушного потока, затем загружать деформированную геометрию и снова проводить

расчет на следующем моменте времени, этот цикл повторяется до конца расчета. Запишем уравнения перемещения лопастей [56,57,58]:

$$u_{def} = \left(\frac{\partial u}{\partial x}\right)(x - x_0) + \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)(y - y_0) + \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)(z - z_0) \quad (24)$$

$$v_{def} = \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)(x - x_0) + \left(\frac{\partial v}{\partial y}\right)(y - y_0) + \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)(z - z_0) \quad (25)$$

$$w_{def} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)(x - x_0) + \frac{1}{2}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right)(y - y_0) + \left(\frac{\partial w}{\partial z}\right)(z - z_0) \quad (26)$$

Затем необходимо првоести в модуле расчета Aerodynamics сохранение необходимых параметров: перемещение узлов лопастей по трем осям, давление, создаваемое на поверхности лопасти, напряжение в каждом узле лопасти (Рисунок 2.17).

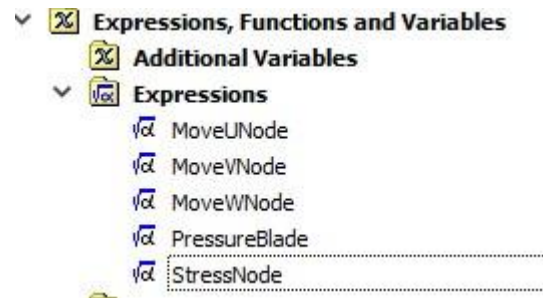


Рисунок 2.17. Функции для записи в выходной файл необходимых параметров в модуле Resonance

Получаемые данные в процессе расчета записываются с помощью языка программирования Python в подготовленную структуру файла DataNodeAerodynamics.dna.

Завершающим этапом настроек расчета является добавление экспертного параметра. Необходимый экспертный параметр заменяет привязку промежуточного расчета параметров от переменной f , которая берется со всех объектов и переносится на следующий промежуток и повторяет расчет с новыми внешними условиями без привязки по времени. В данном случае завершение процесса решения будет заключаться в прекращении скачков значений параметра f на протяжении некоторого числа итераций. Данный алгоритм применяется для анализа задач в режиме Steady State, где необходимо найти установившиеся параметры в среде решаемой задачи. Для решения динамического взаимодействия в заданном промежутке времени, необходимо заменить переменную f на t (Рисунок 2.18).

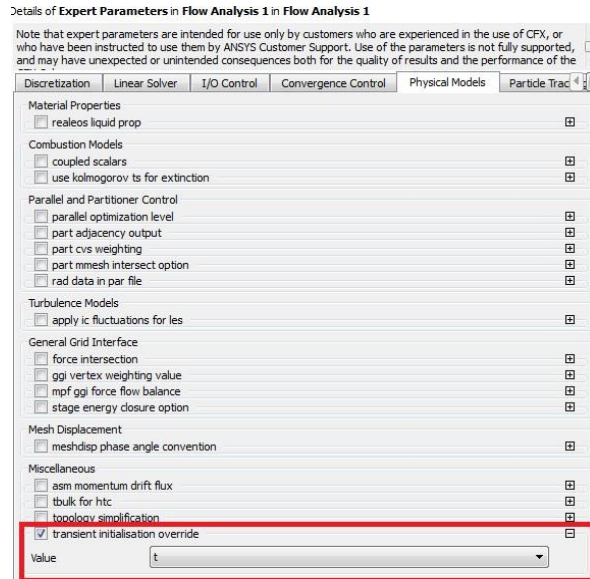


Рисунок 2.18. Включение экспертного параметра для расчета задачи по времени

2.5.4 Запуск модуля CFX и проверка исходных данных задачи

После задания всех необходимых параметров расчета и установки контрольных точек, требуется осуществить запуск решения. Процессом решения поставленной задачи занимается решатель, который принимает исходные данные данной задачи и переводит их на понятный ему язык. Решатель расположен в модуле Aerodynamics и обозначается как Decision. При его запуске настроить параметры первоначальных значений на каждом шаге по времени, это делается для того, чтобы при изменении сетки лопастей винта, вследствие деформации, в решатель подгружалась также и новая сетка воздушной области, для сохранения густоты сетки в области лопастей и уравнения (23) (Рисунок 2.19).

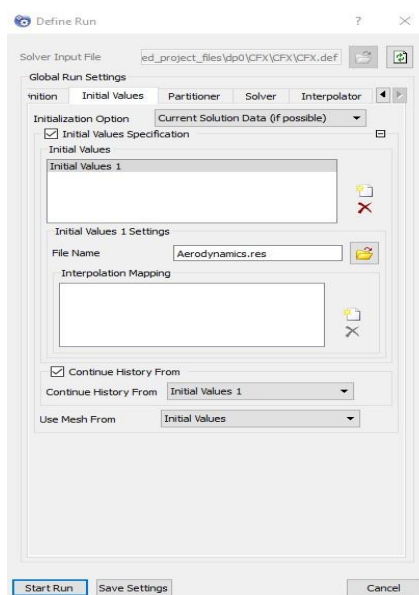


Рисунок 2.19. Настройки решателя для решения задачи динамики.

При включении данного параметра необходимо инициализировать файл, который будет содержать результаты предыдущих по времени шагов. В нем будут содержаться заданные ранее параметры расчета перемещения лопастей винта, давления и напряжения.

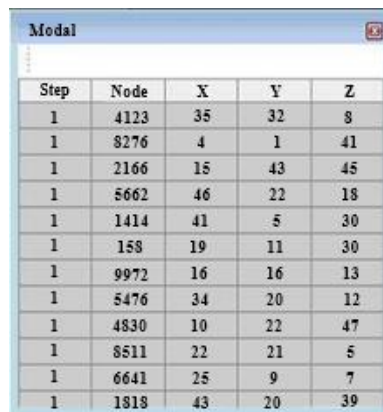
Следует ответить, что при установке данного параметра, необходимо заменить считывание сетки не из результата решателя в формате *.def, а из исходной сетки, которая редактируется в процессе расчета используя сеточные генераторы ANSYS.

Также модуль Aerodynamics содержит в себе код на языке программирования Python, позволяющий перезаписывать общий файл с содержанием сеток воздушной области и исследуемого тела отдельно друг от друга (так как сетка накладывалась на две цифровые модели отдельно), что позволяет ускорить процесс перестройки благодаря задействованию процесса распараллеливания.

2.5.5 Учет частот свободных колебаний лопастей винта в расчетах динамического взаимодействия лопастей винта с воздушной средой

По завершении процесса формирования файла перемещения лопастей под действием воздушного потока его необходимо подключить в модуль Resonance с помощью добавленной функции в процессе создания модуля.

Данная функция подключает файл и формирует таблицу перемещения лопастей винта в каждый момент времени по трем координатным осям (Рисунок 2.20).



Step	Node	X	Y	Z
1	4123	35	32	8
1	8276	4	1	41
1	2166	15	43	45
1	5662	46	22	18
1	1414	41	5	30
1	158	19	11	30
1	9972	16	16	13
1	5476	34	20	12
1	4830	10	22	47
1	8511	22	21	5
1	6641	25	9	7
1	1818	43	20	39

Рисунок 2.20. Таблица перемещений каждого узла по результатам модуля Aerodynamics

После этого происходит перерасчет с помощью вычислительных средств ANSYS на основе алгоритма перебора частот свободных колебаний с каждым результатом подключенной таблицы. Перед запуском расчета устанавливается критерий напряжений или деформаций, который является фильтром для вывода конечных результатов (Рисунок 2.21).

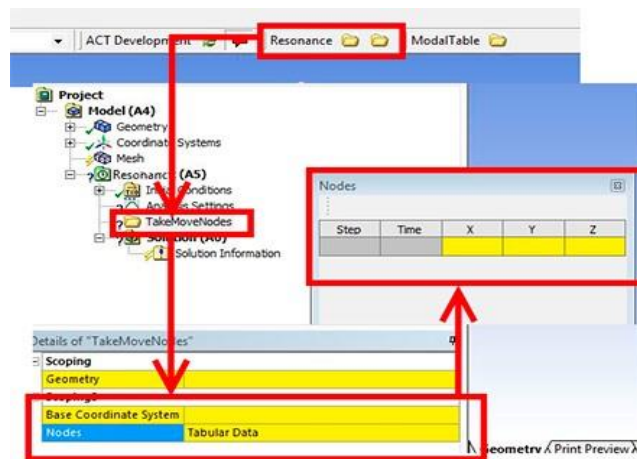


Рисунок 2.21. Подключение таблицы в модуле Resonance.

Алгоритм перебора оформляется на языке Python и внедряется в модуль Resonance и запускается в момент нажатия кнопки Solve. Подробная информация по алгоритму перебора описана в главе 4.

По окончании этапа расчета формируются внутри модуля готовые результаты по деформации и напряжениям согласно критерию. Кроме этого, формируется таблица, которая демонстрирует какая частота свободных колебаний влияет на лопасть и в какой промежуток времени.

2.5.6 Расчеты. Анализ результатов расчета

Анализируя результаты расчета, получаемые в виде таблицы, можно определить какие частоты свободных колебаний являются опасными и используя встроенные средства оптимизации геометрии провести процесс изменения её параметров в автоматическом режиме с заданными данными по снижению или увеличению той или иной частоты свободных колебаний. Однако в данной случае необходимо будет провести дополнительные расчеты по влиянию воздушного потока на исследуемое тело (Рисунок 2.22).

Step	Node	X	Y	Z	Time	Modal	Stress	Crit
1	354	544	534	43	4	1204	745	660
2	743	618	641	21	18	746	671	660
3	18445	235	653	12	32	846	801	660
4	84536	1733	1754	84	42	642	726	660
5	6346	843	793	70	47	982	841	660
6	96647	784	842	78	53	729	866	660

Рисунок 2.22. Таблица результатов колебаний лопастей винта, превышающих установленный критерий

Кроме таблиц со значениями совпадающих данных, возможно построить графики перемещения лопастей винта, получаемые из модуля Aerodynamics, по оси

X отчается время вращения винта, по оси Y – расстояние, на сколько переместилась точка из состояния покоя (Рисунок 2.23).

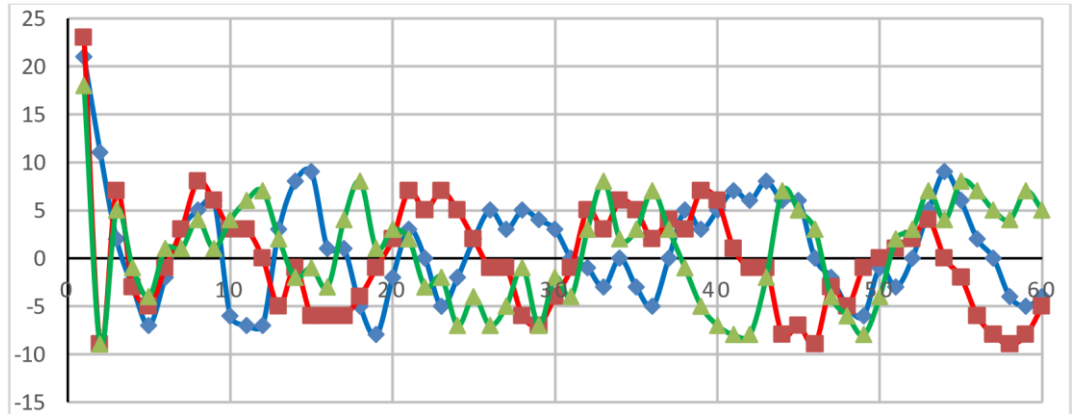


Рисунок 2.23. График перемещения трех контрольных точек, расположенных вдоль на одной лопасти (красный – точка №1, синий – точка №2, зеленый – №3)

ГЛАВА 3. РАЗРАБОТКА НОВОГО КОНЕЧНОГО ЭЛЕМЕНТА НА ОСНОВЕ ОРТОГОНАЛЬНЫХ ФИНИТНЫХ ФУНКЦИЙ И ЕГО РЕАЛИЗАЦИЯ В ПРОГРАММНОЙ СРЕДЕ ANSYS. ПРИМЕНЕНИЕ МОДИФИЦИРОВАННОГО МКЭ, СВЯЗАННОГО С НОВЫМ КОНЕЧНЫМ ЭЛЕМЕНТОМ В РЕШЕНИИ СВЯЗАННОЙ ЗАДАЧИ АЭРОУПРУГОСТИ

3.1 Общая структура пользовательского конечного элемента

Общая структура файла для конечного элемента практически ничем не отличается от основной массы любых файлов, содержащих в себе исходную информацию о проекте. Структура включает в себя: раздел функций, раздел переменных, раздел ввода и вывода данных и раздел расчета.

Перед тем как приступить к внесению изменений в структуру файла необходимо разобраться в том, как работает этот файл в программном комплексе и какие функции он выполняет в нем, но сначала необходимо определить саму структуру программного комплекса, чтобы определить в каком месте требуется вносить изменения.

Программный комплекс ANSYS имеет два рабочих пространства. Это стандартная оболочка Mechanical ANSYS APDL, созданная на языке программирования Tc\Tkl, ее вычислительная часть - на Fortran 77 (90), и оболочка ANSYS Workbench, у которой интерфейсные части основана на языке программирования Python, а вычислительная использует Python в качестве конвейера передачи данных ввода и вывода, а ядро решателя и всех входящих в него компонентов базируется на Fortran 77 (90) [59,60,61]. Первоначальная разработка пользовательского конечного элемента будет осуществлена на языке программирования высокого уровня Fortran 77 с апробацией результатов внутри оболочки Mechanical ANSYS APDL со стандартным элементом Plane 142. Это позволит сформировать и внедрить в структуру вычислительной системы динамическую библиотеку, которая будет содержать в себе разработанный конечный элемент с ортогональными финитными функциями, для дальнейшего её использования в разработанной программной среде, направленной на расчет взаимодействия твердых тел с окружающей средой.

Составлена схема работы в Mechanical ANSYS APDL для решения упругой задачи (Рисунок 3.1).



Рисунок 3.1 – Блок-схема решения задачи в программном комплексе Mechanical ANSYS APDL

Существует два способа внедрения в вычислительную структуру ANSYS нового конечного элемента [62,63].

1 способ. Так как интерфейсная оболочка Mechanical ANSYS APDL сформирована на языке Тс\Тк, то на этапе запуска оболочки происходит загрузка списка всех имеющихся в библиотеке конечных элементов. Библиотека конечных элементов представляет собой динамическую библиотеку формата DLL, которая компилируется в специализированной программной среде из отдельных статических библиотек формата LIB, которые в свою очередь содержат полный функционал каждого конечного элемента.

Способ внедрения в основной пакет программ заключается в формировании полной структуры нового конечного элемента с ортогональными финитными функциями в формате статической библиотеки с описанием функций ввода и вывода

данных для работы с программным комплексом ANSYS и его компиляция совместно с другими файлами в единую динамическую библиотеку.

2 способ. Первоначально программный комплекс ANSYS содержал в себе до 10 конечных элементов. Разработчики на стадии развития программного комплекса добавили возможность использовать исходные файлы одного из конечных элементов Plane142 (Solid182) на языке программирования Fortran 77. Это позволило многим людям увидеть принципы работы конечного элемента внутри вычислительной системы, при этом возможности были ограничены внедрением в структуру программного комплекса своего конечного элемента и вычислительными способностями.

Конечный элемент Plane142 (Solid182) был направлен на решение задач, связанных с определением упругопластических деформаций и напряжений в 2D и 3D пространствах. Внедрение в данном случае заключается в поиске среди исходных файлов вычислительного модуля, отвечающего за формирование функций формы, производных функции формы и матрицы жесткости, и внедрение в него своего кода, кроме этого, доступ к вычислительному полиному будет скрыт внутри программного комплекса.

Рассмотрим теоретическую сторону применения четырехугольного конечного элемента Plane142 в теории упругости внутри программного комплекса ANSYS для понимания, какую структуру файла необходимо искать. Для этого необходимо произвести постановку плоской задачи теории упругости.

Система разрешающих уравнений теории упругости относительно трех неизвестных:

- полей перемещений $u = (u_x, u_y)^T$,
- деформаций $\varepsilon = (\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy})^T$,
- напряжений $\sigma = (\sigma_{xx}, \sigma_{yy}, \sigma_{xy})^T$

При отсутствии начальных напряжений в теле эта система уравнений записывается в развернутом матричном виде:

$$\begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{pmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad (27)$$

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{pmatrix} \quad (28)$$

$$\begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial y} \end{bmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad (29)$$

где $\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}$ – компоненты тензора деформаций; $\sigma_{xx}, \sigma_{yy}, \sigma_{xy}$ – компоненты тензора напряжений; d_{ij} ($i, j = 1, 2, 3$) – модули упругости; u_x, u_y – компоненты вектора перемещений; f_x, f_y – компоненты вектора внешних объемных сил.

Записанная система трех матричных уравнений содержит вектор объемных сил с компонентами f_x, f_y , входящий в уравнение равновесия, матрицу упругих модулей с компонентами d_{ij} , связывающую напряжения и деформации.

В компактном матричном виде эта система уравнений может быть записана:

$$\varepsilon = B u, \sigma = D \varepsilon, B^T \sigma = f, \quad (30)$$

где $D = (d_{ij})$ – симметричная матрица упругих модулей, B – матрица, состоящая из операторов – частных производных, f – вектор объемных сил.

В задачах теории упругости различают плоско-напряженное и плоскодеформированное состояния. С вычислительной точки зрения эти модели будут отличаться только соответствующими матрицами упругих модулей для изотропного материала [64]:

$$D = \frac{E}{1 - \nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - 2\nu}{2(1 - \nu)} \end{pmatrix}, \quad (31)$$

$$D = \frac{E(1 - \nu)}{(1 + \nu)(1 - 2\nu)} \begin{pmatrix} 1 & \frac{\nu}{1 - \nu} & 0 \\ \frac{\nu}{1 - \nu} & 1 & 0 \\ 0 & 0 & \frac{1 - 2\nu}{2(1 - \nu)} \end{pmatrix}, \quad (32)$$

где E – модуль Юнга материала, ν – коэффициент Пуассона.

Рассмотрим несколько граничных условий.

1. Кинематические граничные условия $u|_{\Gamma} = \bar{u}$ (условия Дирихле).
2. Силовые граничные условия, когда на часть границы области действует сила (условия Неймана).

Действующие силы могут быть объемными $f = (f_x, f_y)$, поверхностными $p = (p_x, p_y)$ [65].

Система эквивалентна задаче минимизации полной потенциальной энергии тела и сводится к минимизации функционала энергии.

Элементная матрица жесткости и элементный вектор правых частей для плоской задачи с единичной толщиной элемента могут быть записаны в виде:

$$k^e = \int_{S^e} (N^e)^T \begin{pmatrix} f_x \\ f_y \end{pmatrix} dS + \int_{l^e} (N^e)^T \begin{pmatrix} p_x \\ p_y \end{pmatrix} dl, \quad (34)$$

где k^e – элементная матрица жесткости; $B_N = BN$ – матрица производных функций формы N ; p_x, p_y – компоненты вектора сил на границе области; S^e – площадь конечного элемента; l^e – граница элемента, на которой задана нагрузка.

Для четырехугольного билинейного элемента матрица B_N зависит от координат и её нельзя вынести за знак интеграла (Рисунок 3.2).

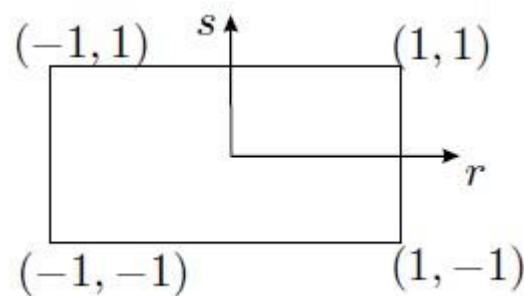


Рисунок 3.2 – Четырехугольный базовый билинейный элемент в локальной системе координат

Система уравнений и глобальная матрица системы будет иметь вид:

$$Ku = f, \quad K = \sum_e k^e, \quad f = \sum_e f^e + G, \quad (35)$$

где G – вектор узловых сил.

Теперь необходимо переписать формулу для билинейного прямоугольного элемента. В программном комплексе Mechanical ANSYS APDL для вычисления интегралов в этой формуле используются квадратичные формулы Гаусса [66]

$$\int_{-1}^1 f(x) dx \cong \sum_{i=1}^s W_i f(t_i). \quad (37)$$

При этом узлы t_1, t_2, \dots, t_s и коэффициенты W_1, W_2, \dots, W_s подобраны так, чтобы формула интегрирования была точной для всех многочленов степени $(2s - 1)$. Тогда элементная матрица для билинейного элемента с использованием двух точек интегрирования для каждой и координат имеет вид

$$k^e = \int_{s^e} B_N^T D B_N dS = \sum_{i=1}^2 \sum_{j=1}^2 W_i W_j B_N^T(r_i, s_j) D B_N(r_i, s_j) \det |J(r_i, s_j)|, \quad (38)$$

$$B_N = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{pmatrix}. \quad (39)$$

Производные функций форм получаются с помощью матрицы Якоби, которая вычисляется исходя из состояний [67] $x = \sum_k N_k x_k$, $y = \sum_k N_k y_k$ и в итоге получается:

$$J(r_i, s_j) = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{pmatrix} = \begin{pmatrix} \sum_k x_k \frac{\partial N_k}{\partial r}(r_i, s_j) & \sum_k y_k \frac{\partial N_k}{\partial r}(r_i, s_j) \\ \sum_k x_k \frac{\partial N_k}{\partial s}(r_i, s_j) & \sum_k y_k \frac{\partial N_k}{\partial s}(r_i, s_j) \end{pmatrix} \quad (40)$$

Необходимо рассмотреть вычисление интеграла $\int_l N^T \cdot p dl = \int_{l^e} (N^e)^T \begin{pmatrix} p_x \\ p_y \end{pmatrix} dl$, который присутствует в правой части уравнения, для понимания всего процесса работы кода. Пусть в качестве стороны интегрирования выбрана сторона 2-3 прямоугольника (Рисунок 3.3).

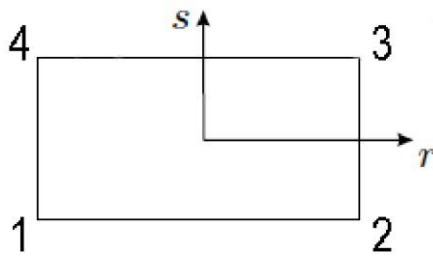


Рисунок 3.3. Нумерация узлов четырехузлового конечного элемента

Эта сторона соответствует значению локальной координаты $r = 1$. Значит функции формы будут иметь вид

$$N_1 = \frac{(1-r)(1-s)}{4},$$

$$N_2 = \frac{(1+r)(1-s)}{4},$$

$$N_3 = \frac{(1+r)(1+s)}{4},$$

$$N_4 = \frac{(1-r)(1+s)}{4}, \quad (41)$$

$$\int_l N^T \cdot pdl = \int_{-1}^1 N^T | \det(J) | ds = \frac{l}{2} \begin{pmatrix} 0 \\ 0 \\ 2p_x \\ 2p_y \\ 2p_x \\ 2p_y \\ 0 \\ 0 \end{pmatrix} \quad (42)$$

Переходим к описанию созданных исходных подпрограмм, содержащих в себе все описанные выше формулы.

Такой подпрограммой является UserElem.f, которая содержит в себе всю необходимую для разработки информацию. Ее структура отражает определенные фрагменты кода: переменные, удаленные функции, вычисление свойств материала, формирование функций формы на основе указанного 2D или 3D пространства, вычисление производных функции формы, построение матрицы Якоби, локальной и глобальной матриц жесткости, верификацию полученных значений, вывод и предоставление результатов работы файла. Для формирования алгоритма внедрения ортогональных финитных функций необходимо использовать и модифицировать существующие в ANSYS (полный код файла с подпрограммой представлен в приложении А).

Далее рассматривается блок подпрограммы, содержащий в себе название подпрограммы и имена входных переменных. Затем идет описание всех содержащихся в подпрограмме переменных с присвоением им различного типа.

В данный блок будут введены дополнительные переменные для осуществления возможности внедрения ортогональных финитных функций.

```

subroutine UserElem (elId, matId, keyMtx, lumpm, nDim, nNodes,
&                Nodes, nIntPnts, nUsrDof, kEStress,
&                keyAnsMat, keySym, nKeyOpt, KeyOpt,
&                temper, temperB, tRef, kTherm,
&                nPress, Press, kPress, nReal, RealConst,
&                nSaveVars, saveVars, xRef, xCur,
&                TotValDofs, IncValDofs, ItrValDofs,
&                VelValDofs, AccValDofs,
&                kfstps, nlgeom, nrkey, outkey, elPrint, iott,
&                keyHisUpd, ldstep, isubst, ieqitr, timval,
&                keyEleErr, keyEleCnv,
&                eStiff, eMass, eDamp, eSStiff,
&                fExt, fInt, elVol, elMass, elCG,
&                nRsltBsc, RsltBsc, nRsltVar, RsltVar,
&                nElEng, elEnergy)

```

```

&

#include "impcom.inc"
c
EXTERNAL      ElemGetMat

INTEGER      elId, matId, keyMtx(10), lumpm, nDim, nNodes,
&            Nodes(nNodes), nIntPnts, nUsrDof, kESTress,
&            keyAnsMat, keySym, nKeyOpt, KeyOpt(nKeyOpt),
*            kTherm, nPress, kPress, nReal, nSaveVars,
&            kfstps, nlgeom, nrkey, outkey,
&            elPrint, iott, keyHisUpd,
&            ldstep, isubst, ieqitr, keyEleErr, keyEleCnv,
&            nRsltBsc, nRsltVar, nElEng

DOUBLE PRECISION temper(nNodes), temperB(nNodes), tRef,
&            Press(nPress), RealConst(nReal),
&            saveVars(nSaveVars),
&            xRef(nDim,nNodes), xCur(nDim,nNodes),
&            TotValDofs(nUsrDof), IncValDofs(nUsrDof),
&            ItrValDofs(nUsrDof), VelValDofs(nUsrDof),
&            AccValDofs(nUsrDof),      timval,
&            eStiff(nUsrDof,nUsrDof), eMass(nUsrDof,nUsrDof),
&            eDamp(nUsrDof,nUsrDof), eSStiff(nUsrDof,nUsrDof),
&            fExt(nUsrDof), fInt(nUsrDof),
&            elVol, elMass, elCG(3),
&            RsltBsc(nRsltBsc), RsltVar(nRsltVar),
&            elEnergy(nElEng)

#include "locknm.inc"

EXTERNAL      vzero, vmove, vmult, vdot, vidot,
&            maxv, matxb, matba, maat, matsym, getMatProp,
&            erhandler, equivStrain, ElemJac, ElemMass,
&            ElemRsltNode, ElemShpFn, plock, ppunlock

DOUBLE PRECISION vdot, vidot,TBMat(8,8),Stif(8,8)

INTEGER      nUsrDof2, intPnt, iNode, nTens, flgSingular,
&            k1, k2, k3, nComp, iDim, iDim1, iComp,
&            nNodesCorner, nDirect, kThermIP,n,ic,ir
DOUBLE PRECISION BMat(nDim*2,nUsrDof), Ex, nu, density, G, workDb,
&            con1, con2, cMat(nDim*2,nDim*2), shIsoC(nNodes),
&            shIso(nNodes), shDerIso(nDim,nNodes), wtIP(1),
&            workArr(360), elJac(nDim*nDim), detJac, dperr(2),
&            shDerEl(nDim,nNodes), dVol, Strain(nDim*2),
&            Stress(nDim*2), wStrain(48), wStress(48),
&            nStrain(48), nStress(48), sigm, tem, prop(3),
&            IncStrain(nDim*2), defG(3,3), Pnt(2,4),Nn(4),
&            defG0(3,3), xCurIP(nDim), TemperIP, dN(2,4),
&            TemperIPB, StressTh(nDim*2), MatProp(5),
&            StrainPl(nDim*2), StrainCr(nDim*2),
&            StrainTh(nDim*2), StrainSw, StressBk(nDim*2),
&            MatRotGlb(3,3), wStrainTh(48), wStrainPl(48),

```

& CHARACTER*4	wStrainCr(48), eMassb(nNodes,nNodes), EnergyD(3) label(3)
------------------	--

Следующий блок кода содержит системную информацию о настройках конечного элемента в программном комплексе, о задании для него правил поведения в вычислительной области. Так как подпрограмма использует базовый конечный элемент Plane 142 и Solid 186, в первом случае используется для решения 2D задач, второй – для решения 3D задач. В данной блоке формируются данные для переменных решение задач какой размерности будет проводиться далее. Кроме этого, в этом блоке указывается какие расчетные матрицы использовать при формировании конечного элемента. Модификация данного блока заключается во ведении дополнительного цикла формирования массива точек квадратуры Гаусса для расчета координат узлов конечного элемента согласно информации из описания решателя программного комплекса ANSYS. О параметрах вводимых для работы конечного элемента в этом блоке будет рассмотрена в подглаве 3.6.

```

nTens = nDim*2
  nComp = nDim*nDim
  nDirect = 3
  nUsrDof2 = nUsrDof*nUsrDof
  CALL vzero (BMat(1,1),nUsrDof*nTens)
  IF (keyMtx(1).EQ.1) CALL vzero (eStiff(1,1),nUsrDof2)
  IF (keyMtx(2).EQ.1) CALL vzero (eMass(1,1) ,nUsrDof2)
  IF (keyMtx(5).EQ.1) CALL vzero (fExt(1) ,nUsrDof)
  IF (keyMtx(6).EQ.1) CALL vzero (fInt(1) ,nUsrDof)
  IF (nlgeom.EQ.0) THEN
    DO iDim = 1, 3
      DO iDim1 = 1, 3
        defG0(iDim, iDim1) = 0.0D0
      END DO
      defG0(iDim, iDim) = 1.0D0
    END DO
    CALL vmove (defG0(1,1),defG(1,1),9)
  ELSE

  END IF
  elVol = 0.d0
  elMass= 0.d0
  IF (keyMtx(2).EQ.1) CALL vzero (eMassb(1,1),nNodes*nNodes)
  CALL vzero (elEnergy(1), nElEng)

  IF (debug.EQ.1) THEN
    write (*,*)
    write (*,*) 'elId =',elId,
    & ' keyAnsMat=',keyAnsMat, ' nDim=',nDim, ' kEStress=',
    & kEStress, ' keySym=',keySym, ' nIntPnts=',nIntPnts
  END IF

  IF (KeyOpt(1).EQ.0) THEN
    CALL ElemShpFn (1, 0, 1, shIsoC(1), nNodes)
  ELSE IF (KeyOpt(1).EQ.1) THEN
    CALL ElemShpFn (2, 0, 1, shIsoC(1), nNodes)
  
```



```

ELSE
    dperr(1) = KeyOpt(1)
    dperr(2) = elId
    CALL erhandler ('UserElem', 1000, 3, 'Invalid input
&                                KEYOPT(1) = %I at element %I.', dperr(1),
&                                ' ')
    END IF
    elCG(3) = 0.0d0
    DO iDim = 1, nDim
        elCG(iDim) = vdot (shIsoC(1), 1, xCur(iDim,1), nDim,
&                                nNodes)
    END DO

```

Следующий блок кода отвечает за формирование матрицы упругих элементов. Как следует из теории, в этой части кода программы формируются матрицы из формул (35) или (36).

Данный блок вступает в действие только в том случае, если в программном комплексе отсутствуют уже заданные свойства материала. При этом эта часть кода может работать только параллельно с внедряемым файлом `usermat.f`, который содержит в себе полную информацию о математическом моделировании материала, что позволяет пользователю задать свои значения и применить их для решения своей задачи с модифицированным материалом. При этом в данной работе эта часть кода не является важной и её редактирование не требуется, однако его исключение из общей структуры файла влечёт за собой возникновение ряда ошибок, связанных с условием, указанном в его начале, то есть в переменных из базовой функции.

```

IF (keyAnsMat.EQ.1) THEN

    ELSE

        label(1) = 'ex  '
        label(2) = 'prxy'
        label(3) = 'dens'
        tem = vdot (shIsoC(1), temper(1), nNodes)

        CALL getMatProp (elId, matId, tem, 3, label(1), prop(1))
        Ex = prop(1)
        nu = prop(2)
        density = prop(3)

        IF (nReal.GE.1 .AND. Ex .EQ.0.0d0) Ex = RealConst(1)
        IF (nReal.GE.2 .AND. nu .EQ.0.0d0) nu = RealConst(2)
        IF (nReal.GE.3 .AND. density.EQ.0.0d0) density = RealConst(3)

        G = 0.5d0*Ex/(1.0d0+nu)
        workDb = (1.0d0+nu)*(1.0d0-2.0d0 * nu)
        con1 = (1.0d0-nu)*Ex/workDb
        con2 = nu*Ex/workDb
        CALL vzero (cMat(1,1), nTens*nTens)
        cMat(1,1) = con1
        cMat(2,2) = con1
        cMat(3,3) = con1

```

```

cMat (4,4) = G
cMat (1,2) = con2
cMat (1,3) = con2
cMat (2,3) = con2
cMat (2,1) = con2
cMat (3,1) = con2
cMat (3,2) = con2
IF (KeyOpt (1) .EQ. 1) THEN
    cMat (5,5) = G
    cMat (6,6) = G
END IF
END IF
kTherm = 0

```

Представленный фрагмент блока ориентирован на вычисление значений функции формы, её производной и весового параметра для различного пространства 2D или 3D. За эти вычисления отвечает функция ElemSphFn, которая вызывается из программного комплекса в процессе его работы и отправляет внутрь себя требуемые параметры, на основании которых производится расчет с выводом результатов. Данная функция вычисляет функции формы по формулам (35). Также формируется матрица производных функций форм B_N по формуле (39). Этот блок будет подвергаться модификации. Функция ElemSphFn является поступает из динамической библиотеки ANSYS в подпрограмму UserElem, значит доступа к этой функции отсутствует и её необходимо будет раскрыть через код, предварительно используя новые переменные и координаты узлов.

```

IF (KeyOpt (1) .EQ. 0) THEN
    CALL ElemShpFn (1, intPnt, 1, shIso (1), nNodes)
    CALL ElemShpFn (1, intPnt, 2, shDerIso (1,1), nUsrDof)
    CALL ElemShpFn (1, intPnt, 3, wtIP (1), 1)
ELSE
    CALL ElemShpFn (2, intPnt, 1, shIso (1), nNodes)
    CALL ElemShpFn (2, intPnt, 2, shDerIso (1,1), nUsrDof)
    CALL ElemShpFn (2, intPnt, 3, wtIP (1), 1)
END DO

```

Далее идет фрагмент кода, в котором происходит формирование матрицы производных функций формы и проверка на положительное значение определителя матрицы Якоби (Якобиан), после чего выполняется математическая операция умножение матрицы производных функции формы и матрицы Якоби.

$$J^{-1} = \begin{pmatrix} \frac{\partial N_s}{\partial r} \\ \frac{\partial N_s}{\partial s} \end{pmatrix} \quad (43)$$

Остальные блоки кода в структуре файла отвечают за вывод значений упругой деформации и напряжений в интерфейс программного комплекса, за вывод текстовой информации о работе конечного элемента в консольное окно программы и за создание

файла с результатами расчета в рабочем каталоге проекта. Модификации данной блок подпрограммы не подвергается.

```

      DO iDim = 1, nDim
        xCurIP(iDim) = vdot(shIso(1), 1, xCur(iDim,1), nDim,
&                               nNodes)
      END DO

      TemperIP = vdot(shIso(1), temper(1), nNodes)
      TemperIPB = vdot(shIso(1), temperB(1), nNodes)

      CALL vzero (workArr(1), nComp)
      iComp = 1
      DO iDim = 1, nDim
        DO iDim1 = 1, nDim
          DO iNode = 1, nNodes
            workArr(iComp) = workArr(iComp)
&                               + shDerIso(iDim1,iNode)*xCur(iDim,iNode)
          END DO
          iComp = iComp + 1
        END DO
      END DO
      CALL ElemJac (workArr(1), elJac(1), nDim, detJac,
&                               flgSingular)
      IF (flgSingular.LE.0) THEN
        dperr(1) = detJac
        dperr(2) = elId
        CALL erhandler ('UserElem', 1100, 3, 'Negative element
&                               Jacobian value %I at element %I. This
&                               is due to wrong element order or bad
&                               mesh.', dperr(1), ' ')
      GOTO 990
    END IF
    DO iNode = 1, nNodes
      IF (KeyOpt(1).EQ.0) THEN
        shDerEl(1,iNode) = elJac(1)*shDerIso(1,iNode)
&                               + elJac(3)*shDerIso(2,iNode)
        shDerEl(2,iNode) = elJac(2)*shDerIso(1,iNode)
&                               + elJac(4)*shDerIso(2,iNode)
      ELSE
        shDerEl(1,iNode) = elJac(1)*shDerIso(1,iNode)
&                               + elJac(4)*shDerIso(2,iNode)
&                               + elJac(7)*shDerIso(3,iNode)
        shDerEl(2,iNode) = elJac(2)*shDerIso(1,iNode)
&                               + elJac(5)*shDerIso(2,iNode)
&                               + elJac(8)*shDerIso(3,iNode)
        shDerEl(3,iNode) = elJac(3)*shDerIso(1,iNode)
&                               + elJac(6)*shDerIso(2,iNode)
&                               + elJac(9)*shDerIso(3,iNode)
      END IF
    END DO
    dVol = detJac*wtIP(1)

```

Дальнейшие блоки представляют собой цикл DO...END DO, который занимает большую часть структуры файла. В цикле производятся все необходимые математические вычисления для каждого узла элемента, а также сохранение и вывод результатов на экран.

3.2 Ортогональные финитные функции

Традиционными аппроксимирующими функциями для метода конечных элементов являются функции Куранта.

В книге [68] для аппроксимации функции были предложены ортогональные финитные функции (ОФФ):

$$\varphi_i(x) = \begin{cases} \frac{(\sqrt{2}-1)(x_{i-1}-x)}{h}, & x \in \left[x_{i-1}, x_{i-1} + \frac{h}{2}\right] \\ \frac{(\sqrt{2}+1)(x-x_i)}{h} + 1, & x \in \left[x_{i-1} + \frac{h}{2}, x_i\right] \\ \frac{(\sqrt{2}-1)(x-x_i)}{h} + 1, & x \in \left[x_i, x_i + \frac{h}{2}\right] \\ \frac{(\sqrt{2}+1)(x_{i+1}-x)}{h}, & x \in [x_i + h/2, x_{i+1}] \\ 0, & x \notin [x_{i-1}, x_{i+1}] \end{cases} \quad (44)$$

Графики этих финитных базисных функций представлены на рисунке 3.4.

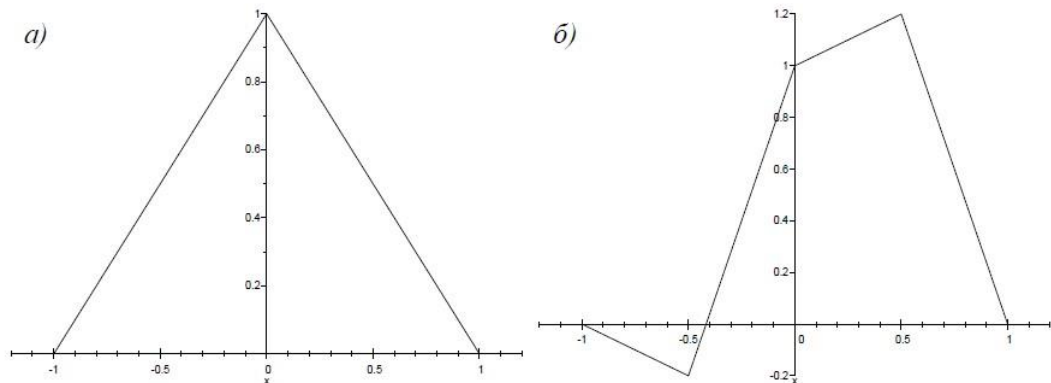


Рисунок 3.4 – Графики базисных функций: а) Куранта; б) ортогональных финитных функций первого типа

Приведем формулировку теоремы об аппроксимирующих свойствах этих ОФФ [68]. В механике деформируемого твердого тела независимая аппроксимация перемещений и напряжений приводит к повышению точности и гладкости приближенного решения для напряжений [69,70,71].

Рассмотрим функцию:

$$\varphi^{(1)}(x) = \begin{cases} 1 + x, x \in [-1, -1 + H_1] \cup [-1 + H_2, 0], \\ -a + \frac{2(a+H_1)(K_N-1-x)}{H_2-H_1}, x \in [-1 + H_1, -1 + K_N], \\ -a + \frac{2(a+H_2)(1-K_N+x)}{H_2-H_1}, x \in [-1 + K_N, -1 + H_2], \\ 1 - x, x \in [0, H_1] \cup [H_2, 1], \\ \beta + \frac{2(\beta+H_1-1)(x-K_N)}{H_2-H_1}, x \in [H_1, K_N], \\ \beta + \frac{2(\beta+H_2-1)(K_N-x)}{H_2-H_1}, x \in [K_N, H_2], \\ 0, x \notin [-1, 1] \end{cases} \quad (45)$$

где $K_N = (H_1 + H_2)/2$, $a > 0, \beta > 0$. Используется равномерная сетка $a = x_1 < x_2 < \dots < x_N = b$ с шагом h , и каждому узлу сетки ставится в соответствие сеточная функция вида:

$$\varphi_i(x) = \varphi^{(1)}(x/h - i) = \begin{cases} \frac{x - x_{i-1}}{h}, x \in [x_{i-1}, x_{i-1} + h_1] \cup [x_{i-1} + h_2, x_i], \\ -a + \frac{2(ah + h_1)(x_{i-1} + k_N - x)}{h(h_2 - h_1)}, x \in [x_{i-1} + h_1, x_{i-1} + k_N], \\ -a + \frac{2(ah + h_2)(x - x_{i-1} - k_N)}{h(h_2 - h_1)}, x \in [x_{i-1} + k_N, x_{i-1} + h_2], \\ (x_{i+1} - x)/h, x \in [x_i, x_i + h_1] \cup [x_i + h_2, x_{i+1}], \\ \beta + \frac{2(\beta h + h_1 - h)(x - K_N)}{H_2 - H_1}, x \in [H_1, K_N], \\ \beta + \frac{2(\beta + H_2 - 1)(K_N - x)}{H_2 - H_1}, x \in [K_N, H_2], \\ 0, x \notin [-1, 1] \end{cases} \quad (46)$$

где $k_N = (h_1 + h_2)/2$, $h_1 = H_1 h$, $h_2 = H_2 h$ ($0 \leq h_1 \leq h_2 \leq h$).

Очевидно, что функция $\varphi_i(x)$ ($i = 1, 2, \dots, N$) линейно независимы. Аппроксимирующие свойства функций (45) описываются следующей теоремой [68].

Теорема 1.1. Если $u(x) \in W_2^1(\mathbb{R})$ и $H_1 + H_2 = 1$ ($h_1 + h_2 = h$), $a = \beta - 1$, существует функция $u^h = \sum_{i=1}^N a_i \varphi_i \in M_N$ (a_i – некоторые постоянные):

$$\|u - u^h\|_{L_2(R)} \leq ch \|u\|_{W_2^1(R)}, \quad \sum_{i=1}^N |a_i|^2 \leq c_i \|u\|_{L_2(R)}^2, \quad (47)$$

M_2 – линейная оболочка функций $\varphi_i(x)$. Постоянные c, c_1 – не зависят от u, h . W_2^1 – функциональные пространства Соболева $\mathbb{L}_2 = W_2^0$.

3.3 Модификация структуры пользовательского конечного алгоритма. Формирование функций формы конечного элемента на основе ортогональных финитных функций

Модификация структуры подпрограммы в основном будет касаться замены функции ElemSphFn и добавления новых переменных. Для удобства новые функции и циклы будут выделены в отдельные файлы и ссылки на них включены в общую структуру.

Для начала на основании теории расчета конечного элемента в программном комплексе создадим массив точек квадратуры Гаусса, присвоив значение 0.57735027, которое соответствует двумерному, используя цикл. Это значение соответствует приближенным значениям переменной из функции ElemSphFn, в которой записаны вычислительные координаты узлов конечного элемента согласно квадратурной формуле Гаусса для двумерного расчета.

```
DO iPnt = 1, 2
  DO jPnt = 1, 4
    Pnt(i, j) = 0.57735027
  END DO
END DO
```

Функция формы будет представлять собой произведение двух базовых ортогональных финитных функций локальных координат системы:

$$N_i = \varphi_i(t)\varphi_i(s), \quad i = \overline{1,4} \quad (48)$$

Запишем функции формы для каждого узла четырехугольного билинейного элемента в локальных координатах (Таблица 3).

Таблица 3.

Ортогональные финитные функции для каждого узла

	$\varphi_i(t)$	$\varphi_i(s)$
N_1	$\begin{cases} \frac{(\sqrt{2}-1)(x+1)}{2} + 1, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(1-x)}{2}, & x \in [0,1] \end{cases}$	$\begin{cases} \frac{(\sqrt{2}-1)(x+1)}{2} + 1, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(1-x)}{2}, & x \in [0,1] \end{cases}$
N_2	$\begin{cases} \frac{(\sqrt{2}-1)(-1-x)}{2}, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(x-1)}{2} + 1, & x \in [0,1] \end{cases}$	$\begin{cases} \frac{(\sqrt{2}-1)(x+1)}{2} + 1, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(1-x)}{2}, & x \in [0,1] \end{cases}$
N_3	$\begin{cases} \frac{(\sqrt{2}-1)(-1-x)}{2}, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(x-1)}{2} + 1, & x \in [0,1] \end{cases}$	$\begin{cases} \frac{(\sqrt{2}-1)(-1-x)}{2}, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(x-1)}{2} + 1, & x \in [0,1] \end{cases}$
N_4	$\begin{cases} \frac{(\sqrt{2}-1)(x+1)}{2} + 1, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(1-x)}{2}, & x \in [0,1] \end{cases}$	$\begin{cases} \frac{(\sqrt{2}-1)(-1-x)}{2}, & x \in [-1,0] \\ \frac{(\sqrt{2}+1)(x-1)}{2} + 1, & x \in [0,1] \end{cases}$

Теперь необходимо записать данные функции, используя на язык программирования Fortran 77. В предыдущих подглавах был определен фрагмент кода в качестве цикла, который перебирает все узлы элемента и проводит расчет для каждого из них. Для нормального функционирования записанных функций требуется сформировать условие, при котором в каждом узле будет проводиться расчет соответствующих базовых функций. Пусть переменные fs и ft соответствуют базовым функциям $\varphi_i(t)$ и $\varphi_i(s)$, а $intPnt$ – номер узла при расчете, тогда данная часть программы имеет вид:

```

if (intPnt.EQ.1) then

    ft(1)=(((sqrt(2.)-1.)*(-Pnt(1,1)+1.))/2.)+1.)
    ft(2)=(((sqrt(2.)-1.)*(-1.+Pnt(1,2)))/2.)
    ft(3)=(((sqrt(2.)-1.)*(-1.+Pnt(1,3)))/2.)
    ft(4)=(((sqrt(2.)-1.)*(-Pnt(1,4)+1.))/2.)+1.)

    fs(1)=(((sqrt(2.)-1.)*(-Pnt(2,1)+1.))/2.)+1.)
    fs(2)=(((sqrt(2.)-1.)*(-Pnt(2,2)+1.))/2.)+1.)
    fs(3)=(((sqrt(2.)-1.)*(-1.+Pnt(2,3)))/2.)
    fs(4)=(((sqrt(2.)-1.)*(-1.+Pnt(2,4)))/2.)

end if

if (intPnt.EQ.2) then

    ft(1)=(((sqrt(2.)+1.)*(1.-Pnt(1,1)))/2.)
    ft(2)=(((sqrt(2.)+1.)*(Pnt(1,2)-1.))/2.)+1.)
    ft(3)=(((sqrt(2.)+1.)*(Pnt(1,3)+1.))/2.)+1.)
    ft(4)=(((sqrt(2.)+1.)*(1.-Pnt(1,4)))/2.)

    fs(1)=(((sqrt(2.)-1.)*(-Pnt(2,1)+1.))/2.)+1.)
    fs(2)=(((sqrt(2.)-1.)*(-Pnt(2,2)+1.))/2.)+1.)
    fs(3)=(((sqrt(2.)-1.)*(-1.+Pnt(2,3)))/2.)
    fs(4)=(((sqrt(2.)-1.)*(-1.+Pnt(2,4)))/2.)

end if

if (intPnt.EQ.3) then

    ft(1)=(((sqrt(2.)+1.)*(1.-Pnt(1,1)))/2.)
    ft(2)=(((sqrt(2.)+1.)*(Pnt(1,2)-1.))/2.)+1.)
    ft(3)=(((sqrt(2.)+1.)*(Pnt(1,3)+1.))/2.)+1.)
    ft(4)=(((sqrt(2.)+1.)*(1.-Pnt(1,4)))/2.)

    fs(1)=(((sqrt(2.)+1.)*(1.-Pnt(2,1)))/2.)
    fs(2)=(((sqrt(2.)+1.)*(1.-Pnt(2,2)))/2.)
    fs(3)=(((sqrt(2.)+1.)*(Pnt(2,3)-1.))/2.)+1.)
    fs(4)=(((sqrt(2.)+1.)*(Pnt(2,4)-1.))/2.)+1.)

end if

if (intPnt.EQ.4) then

    ft(1)=(((sqrt(2.)-1.)*(-Pnt(1,1)+1.))/2.)+1.)
    ft(2)=(((sqrt(2.)-1.)*(-1.+Pnt(1,2)))/2.)

```

```

ft(3)=(((sqrt(2.)-1.)*(-1.+Pnt(1,3)))/2.)
ft(4)=(((sqrt(2.)-1.)*(-Pnt(1,4)+1.))/2.)+1.)

fs(1)=(((sqrt(2.)+1.)*(1.-Pnt(2,1)))/2.)
fs(2)=(((sqrt(2.)+1.)*(1.-Pnt(2,2)))/2.)
fs(3)=(((sqrt(2.)+1.)*(Pnt(2,3)-1.))/2.)+1.)
fs(4)=(((sqrt(2.)+1.)*(Pnt(2,4)-1.))/2.)+1.)

end if

```

Для формирования функций формы требуется также составить цикл с определением узла расчета, чтобы избежать ошибок в записи результатов.

Обозначим $Nn(i)$, $i = 1..4$ массив функций формы в каждом узле на каждом вычисляемом шаге. Соответствующая часть программы имеет вид

```

if (Pnt.EQ.1) then
  N(1)=ft(1)*fs(1)
  N(2)=ft(2)*fs(2)
  Nn3)=ft(3)*fs(3)
  N(4)=(fs(4)*ft(4))
  end if

if (iPnt.EQ.2) then
  N(1)=ft(1)*fs(1)
  N(2)=ft(2)*fs(2)
  N(3)=ft(3)*fs(3)
  N(4)=ft(4)*fs(4)
  end if

if (iPnt.EQ.3) then
  N(1)=ft(1)*fs(1)
  N(2)=ft(2)*fs(2)
  N(3)=ft(3)*fs(3)
  N(4)=ft(4)*fs(4)
  end if

if (iPnt.EQ.4) then
  N(1)=ft(1)*fs(1)
  N(2)=ft(2)*fs(2)
  N(3)=ft(3)*fs(3)
  N(4)=ft(4)*fs(4)
  end if

```

После формирования функций формы их значения при помощи простого цикла перебора передаются в системную переменную базовой функции файла. Если это не сделать, то произойдет crash-ситуация при проведении вычислений внутри конечного элемента во время решения задачи. Такую процедуру требуется проводить каждый раз после вычислений с использованием новых функций формы.

Остается модифицировать алгоритм вычисления производных функций формы. Для этого применяется тот же алгоритм привязки к каждому узлу, что и для вычисления

самих функций формы. Требуется не забыть вставить функцию перебора для предотвращения crash-ситуации.

```

if (intPnt.EQ.1) then
  dN(1,1)=- (1.+s) /4.
  dN(1,2)= (1.+s) /4.
  dN(1,3)= (1.-s) /4.
  dN(1,4)=- (1.-s) /4.
  dN(2,1)=- (1.+t) /4.
  dN(2,2)=- (1.-t) /4.
  dN(2,3)= (1.-t) /4.
  dN(2,4)= (1.+t) /4.
end if

if (intPnt.EQ.2) then
  dN(1,1)=- (1.+s) /4.
  dN(1,2)= (1.+s) /4.
  dN(1,3)= (1.-s) /4.
  dN(1,4)=- (1.-s) /4.
  dN(2,1)=- (1.-t) /4.
  dN(2,2)=- (1.+t) /4.
  dN(2,3)= (1.+t) /4.
  dN(2,4)= (1.-t) /4.
end if

if (intPnt.EQ.3) then
  dN(1,1)=- (1.-s) /4.
  dN(1,2)= (1.-s) /4.
  dN(1,3)= (1.+s) /4.
  dN(1,4)=- (1.+s) /4.
  dN(2,1)=- (1.-t) /4.
  dN(2,2)=- (1.+t) /4.
  dN(2,3)= (1.+t) /4.
  dN(2,4)= (1.-t) /4.
end if

if (intPnt.EQ.4) then
  dN(1,1)=- (1.-s) /4.
  dN(1,2)= (1.-s) /4.
  dN(1,3)= (1.+s) /4.
  dN(1,4)=- (1.+s) /4.
  dN(2,1)=- (1.+t) /4.
  dN(2,2)=- (1.-t) /4.
  dN(2,3)= (1.-t) /4.
  dN(2,4)= (1.+t) /4.
end if

```

Разработанные фрагменты кода будут располагаться в отдельных файлах OFF.for, N42LE2.for и dN42LE2.for соответственно. Это сделано для удобства внесения изменений в каждый фрагмент кода и снижения вероятности поломки основного кода.

3.4 Компиляция динамической библиотеки модифицированных конечных элементов

В старых версиях ANSYS присутствовал собственный компилятор исходного текста файла в конечную библиотеку. Однако, после 15 версии компилятор был выведен из составляющей части и перенесен в раздел с привязкой к каждому виду процессора. На данный момент для компиляции требуется установиться рабочую среду, направленную на компилирование файлов на языке программирования Fortran с ресурсными библиотеками ANSYS для получения исходного файл конечного элемента. В данный момент права на предоставление компилятора для

Fortran принадлежат компании Microsoft, которая предоставляет его сроком на полгода в составе паке Intel Composer XE 2015 [72,73]. Для упрощения процесса компилирования применяется bat-файл ANSUSERSHARED, расположенный в рабочем каталоге программного комплекса ANSYS. Для правильной работы данного файла необходимо редактировать его внутренние параметры, которые обращаются к переменным операционной системы [74,75]. Требуется заменить в его структуре переменные:

```

set PLATFORM_DIR=winx64 set MACHINE_TARGET=X64 set
IFORT_PLATFORM=intel64

%IFORT_COMPILER15% = «Путь к файлу compilervars.bat»
%ROOT_DIR% = «Путь к каталогу с установленным программным
комплексом ANSYS»

```

Для создания динамической библиотеки необходимо расположить bat-файл и разработанные файлы в одной рабочей папке. После запуска требуется указать имя целевого файла, который требуется скомпилировать. При появлении ошибок их требуется устранить различными способами. После компиляции в рабочей папке располагается динамическая библиотека, которую требуется поместить в каталог программного комплекса.

3.5 Использование новых КЭ при решении тестовых задач теории упругости. Сравнительный анализ результатов расчетов с применением нового и стандартного конечных элементов

В качестве способа внедрения нового КЭ была выбрано замена пользовательской библиотеки в структуре программного комплекса ANSYS. Для добавления нового КЭ используется команда:

ET,1,USER300

USER300 – стандартное имя пользовательского конечного элемента и при данном способе внедрения это имя изменить нельзя.

При рассмотрении структуры файла был отмечен фрагмент кода с системными настройками поведения конечного элемента. Данный код говорит о том, что при добавлении элемента в задачу он не имеет никаких настроек и все параметры его работы требуется задавать вручную. Для этого используются две команды:

```
USRDOF, DEFINE, UX,UY
USRELEM, 4, 2, QUAD, 0, 20, 50, 0, 4, 0, 0
```

Системная команда USRDOF отвечает за задание степеней свободы для конечного элемента, в данном случае указывается две степени свободы: перемещения по осям абсцисс и ординат[76,77,78].

Вторая команда отвечает за настройки поведения конечного элемента, то есть за количество узлов в элементе, типа задачи, в которой он применяется (2D или 3D), а также вид конечного элемента (QUAD или BRICK). Эти параметры могут быть прописаны вручную в выделенном фрагменте кода, либо для сохранения структуры файла - вручную внутри программного комплекса. За описанные параметры отвечают первые четыре значения после команды. Число «4» означает количество узлов в конечном элементе, оно также может быть увеличено при условии, что в фрагменте кода прописаны циклы для обработки дополнительных узлов. Остальные числа по большей степени носят обычно незначительное влияние на поведение или расчет задачи, так как их обработка не задана в коде.

Для проведения сравнительного анализа результатов была решена тестовая плоская задача теории упругости. Исследуемый объект представляет собой двухмерную пластину в форме квадрата со стороной 2 м, центр которой совпадал с началом системы координат. Пластина состоит из упругого изотропного материала со следующими свойствами. Модуль Юнга равен 200 ГПа, коэффициент Пуассона равен 0.33. На одной из сторон было поставлено условие жесткого закрепления, на противоположенной стороне по всей длине задавалась сила 10 Н, направленная по нормали к ребру квадрата в сторону от пластины, две остальные стороны пластины свободны от связей и нагрузок.

Расчет проводился с увеличением числа конечных элементов от 10 до 100 на сторонах квадрата. Результаты расчетов представлены в виде графиков, отражающих зависимость максимальной разницы приближенных решений для напряжений и перемещений, полученных в расчетах с использованием разработанного конечного элемента и стандартного конечного элемента Plane142 ANSYS. На рисунке представлено изображение исследуемого тела, сторона 4-1 имеет жесткую заделку, на сторону 2-3 применяется сила направленная в перпендикулярном в сторону от тела – на растяжение. Для проведения сравнительного анализа результатов взяты 5 точек на теле с применением разработанного конечного элемента и стандартного (Рисунок 3.5).

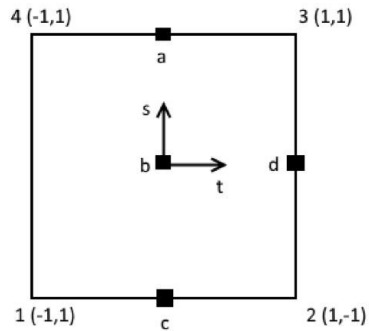


Рисунок 3.5. Места определения номеров узлов для снятия значений результатов перемещений (a,b,c,d - точки, в которых снимаются показания)

Кроме этого, были решены дополнительные задачи с размещением внутри плоского тела вставки прямоугольной формы из другого материала, отличного от основного, с размерами 0.56 x 0.08 (Рисунок 3.6).

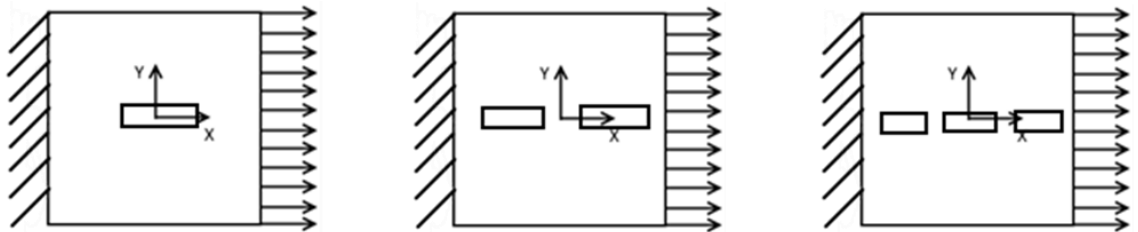


Рисунок 3.6. Варианты размещения вставки из другого материала внутри исследуемого тела.

Было определено время, затрачиваемое на выполнение вычислительных операций. Результаты по времени указаны в таблицах 4 – 7 и соответствующих им рисунках 3.7-3.10 для всех четырех вариантов исследуемого тела.

Таблица 4.

Время, затрачиваемое на выполнение вычислительных операций для конечных элементов User300 и Plane142 с однородной пластиной

Количество элементов на ребре	CPU User300	CPU Plane142	Во сколько раз быстрее
10	0	0	0
20	0,01	0,006	0,6
30	0,04	0,062	1,55
40	0,12	0,125	1,041
50	0,062	0,125	2,016
60	0,094	0,219	2,329
66	0,128	0,241	1,882
70	0,172	0,266	1,546
76	0,181	0,285	1,574

80	0,188	0,312	1,659
86	0,215	0,427	1,986
90	0,25	0,512	2,048
96	0,257	0,593	2,307
100	0,266	0,622	2,335

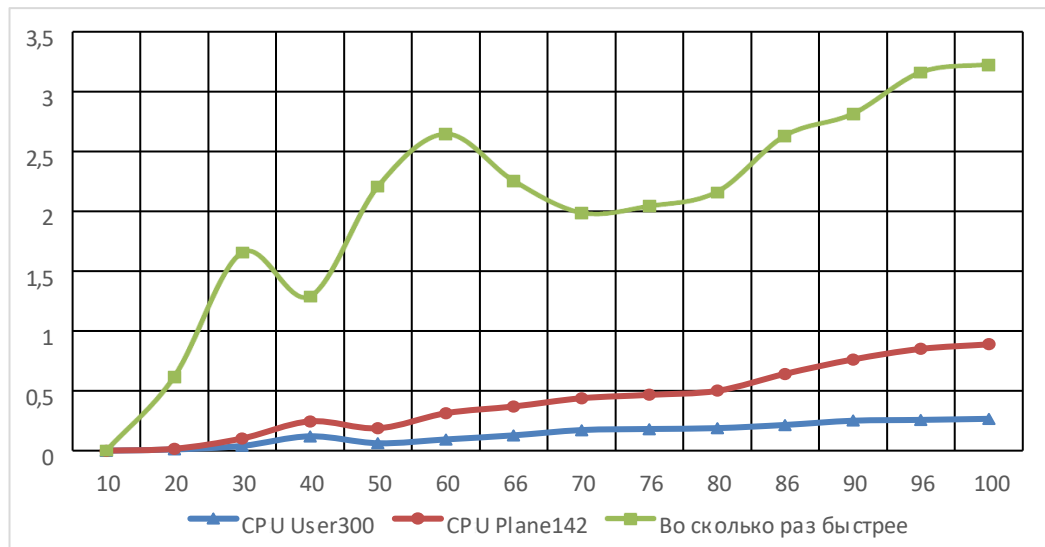


Рисунок 3.7. График затрачиваемого времени на расчет однородной пластины

Таблица 5.

Время, затрачиваемое на выполнение вычислительных операций для конечных элементов User300 и Plane142 с одной вставкой

Количество элементов на ребре	CPU User300	CPU Plane142	Во сколько раз быстрее
10	14,658	16,641	1,135284486
20	34,248	46,781	1,365948377
30	51,671	80,625	1,560353003
40	67,684	130,562	1,928993558
50	82,107	185,734	2,26209702
60	125,347	308,422	2,460545526
70	287,647	765,047	2,659673141
80	541,374	1586,438	2,930391929
90	951,764	3032,172	3,18584439
100	1880,467	6176,734	3,28468088

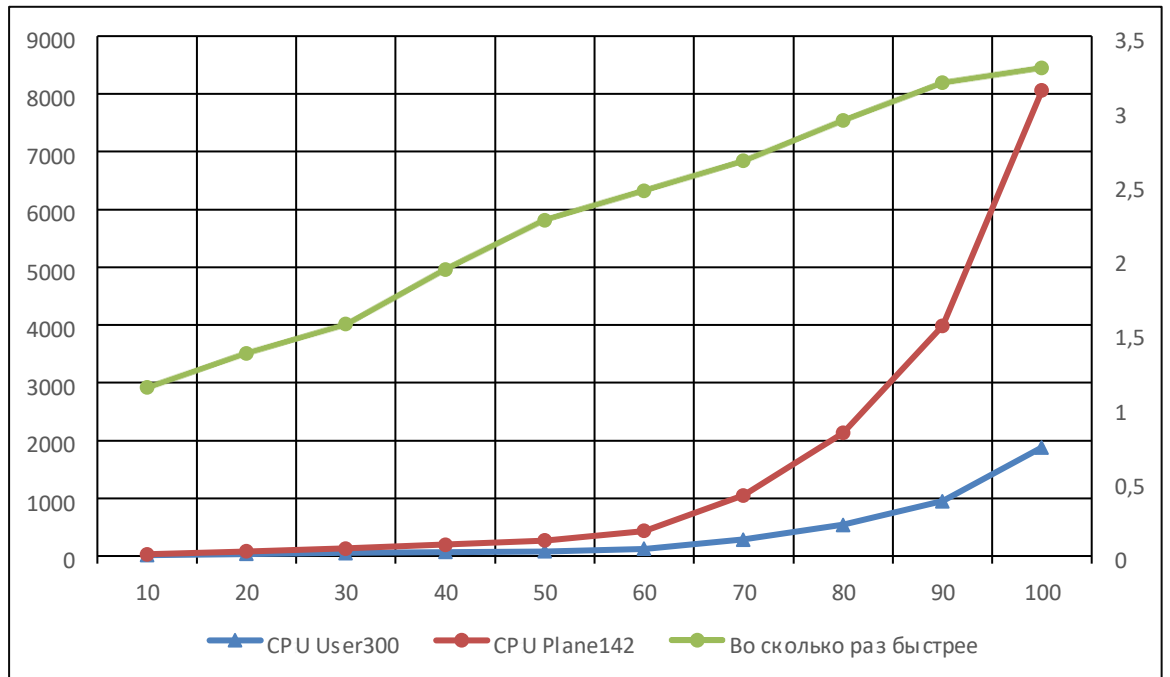


Рисунок 3.8. График затрачиваемого времени на расчет пластины с одной вставкой

Таблица 6.

Время, затрачиваемое на выполнение вычислительных операций для конечных элементов User300 и Plane142 с двумя вставками

Количество элементов на ребре	CPU User300	CPU Plane142	Во сколько раз быстрее
50	0.103	0.125	1.213592233
60	0.279	0.312	1.11827957
70	0.331	0.391	1.181268882
80	0.373	0.577	1.54691689
90	0.414	0.63	1.52173913
100	0.447	0.718	1.606263982

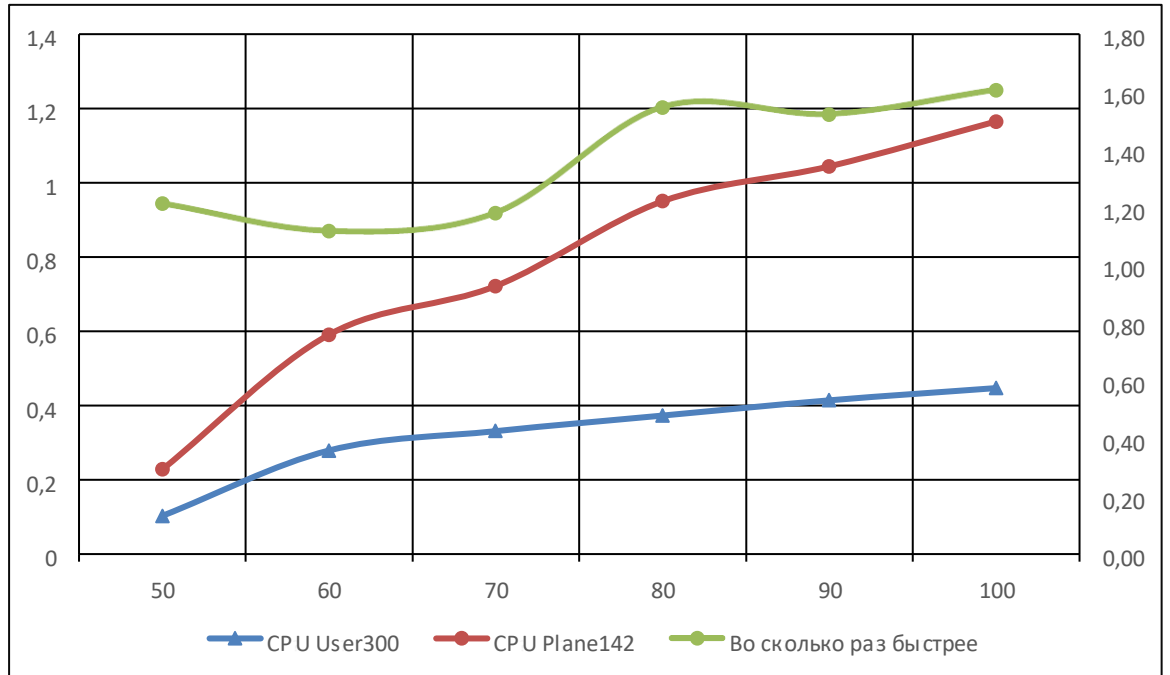


Рисунок 3.9. График затрачиваемого времени на расчет пластины с двумя вставками

Таблица 7.

Время, затрачиваемое на выполнение вычислительных операций для конечных элементов User300 и Plane142 с двумя вставками

Количество элементов на ребре	CPU User300	CPU Plane142	Во сколько раз быстрее
50	0,103	0,125	1,21359223301
60	0,187	0,250	1,33689839572
70	0,301	0,406	1,34883720930
80	0,353	0,469	1,32861189802
90	0,394	0,531	1,34771573604
100	0,438	0,625	1,42694063927

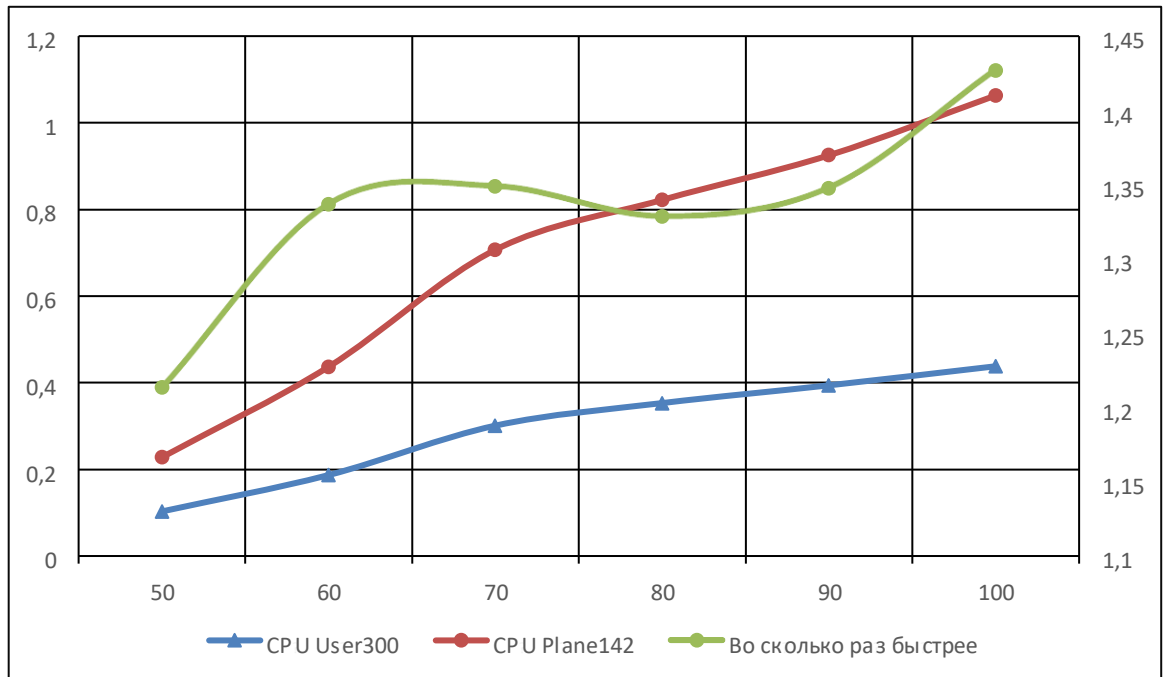


Рисунок 3.10. График затрачиваемого времени на расчет пластины с тремя вставками

Кроме результатов по времени были получены сравнительные результаты по модулю вектора перемещения выбранных узлов для четырех вариантов пластины.

Результаты представлены в таблицах 8 - 19. Изображение результатов в виде графиков по перемещению не предоставляют информативных данных.

Таблица 8

Значения модуля вектора перемещения точек А в плоскости XY и погрешность значений между User300 и Plane142

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «А»	Модуль вектора перемещения для Plane142 в точке «А»	Погрешность между результатами решения на одинаковой сетке в точке «А», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «А», %
10	2.620063556E-09	2.914537501E-10	798.963748142213	1.242032506472
20	2.623274810E-09	5.539481345E-10	373.559643335165	1.120990817457
30	2.629161510E-09	8.163550033E-10	222.061051793887	0.899103632349
40	2.635150100E-09	1.078745142E-09	144.279208999673	0.673375910899
50	2.644124400E-09	1.341128972E-09	97.156608738149	0.335107922839
60	2.650115000E-09	1.603509427E-09	65.269686312858	0.109304438525
66	2.650664780E-09	1.760936616E-09	50.525848342062	0.088581599477
70	2.651109410E-09	1.865887724E-09	42.083008312884	0.071822175841
76	2.651712470E-09	2.023313942E-09	31.057885529066	0.049091055545

Продолжение таблицы 8

80	2.652105320E-09	2.128264497E-09	24.613520722561	0.034283373708
86	2.652384210E-09	2.285689993E-09	16.043042500208	0.023771182318
90	2.652625110E-09	2.390640126E-09	10.958779665359	0.014690946720
96	2.652790241E-09	2.548065063E-09	4.109988379838	0.008466669491
100	2.652914453E-09	2.653014863E-09	0.003784750753	0.003784750753

Таблица 9

Значения модуля вектора перемещения точек В в плоскости XY и погрешность значений между User300 и Plane142

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «В»	Модуль вектора перемещения для Plane142 в точке «В»	Погрешность между результатами решения на одинаковой сетке в точке «В», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «В», %
10	2.354386940E-09	2.530536263E-10	830.3905162018220	1.6634500738085
20	2.364379203E-09	4.911274176E-10	381.4186946748050	1.2460995301569
30	2.371358460E-09	7.290775710E-10	225.2546168369240	0.9545943222542
40	2.377346930E-09	9.669864625E-10	145.8511077656370	0.7044716813528
50	2.384330250E-09	1.204875647E-09	97.8901520614766	0.4127967726267
60	2.390327060E-09	1.442753634E-09	65.6781174324874	0.1623257918613
66	2.391776400E-09	1.585476860E-09	50.8553332024032	0.1017906721456
70	2.392956900E-09	1.680624550E-09	42.3849782510912	0.0524842921213
76	2.393279400E-09	1.823344694E-09	31.2576501785679	0.0390143153675
80	2.393484700E-09	1.918490667E-09	24.7587356649883	0.0304394743519
86	2.393581570E-09	2.061208626E-09	16.1251481197712	0.0263934692414
90	2.393666810E-09	2.156353353E-09	11.0053139792623	0.0228332186414
96	2.393899556E-09	2.299069696E-09	4.1247057522870	0.0131120303948
100	2.394299556E-09	2.394213486E-09	0.0035949175169	0.0035949175169

Таблица 10

Значения модуля вектора перемещения точек D в плоскости XY и погрешность значений между User300 и Plane142

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «D»	Модуль вектора перемещения для Plane142 в точке «D»	Погрешность между результатами решения на одинаковой сетке в точке «D», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «D», %
10	4.840145745E-09	5.095828728E-10	849.825043845155	1.748237213357
20	4.853458480E-09	1.000479579E-09	385.113197897665	1.477997482949
30	4.869125470E-09	1.491258513E-09	226.511160040566	1.159967930502
40	4.884046840E-09	1.982001553E-09	146.419930025151	0.857073971739
50	4.900145748E-09	2.472727909E-09	98.167607934739	0.530276769078
60	4.913012570E-09	2.963444851E-09	65.787211067624	0.269088778144
66	4.920997840E-09	3.257872015E-09	51.049452444497	0.106993069634
70	4.923254780E-09	3.454155841E-09	42.531345041302	0.061178677840
76	4.924048724E-09	3.748580402E-09	31.357692671414	0.045062138863
80	4.924617846E-09	3.944862772E-09	24.836227028077	0.033509338041
86	4.925134567E-09	4.239285476E-09	16.178412491511	0.023020242920
90	4.925547622E-09	4.435566783E-09	11.046634240249	0.014635499151
96	4.925885447E-09	4.729988103E-09	4.141603313458	0.007777874709
100	4.926427484E-09	4.926268606E-09	0.003225118497	0.003225118497

Таблица 11

Значения модуля вектора перемещения точек A в плоскости XY и погрешность значений между User300 и Plane142 (с одной вставкой)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «A»	Модуль вектора перемещения для Plane142 в точке «A»	Погрешность между результатами решения на одинаковой сетке в точке «A», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «A», %
10	2,62E-09	2,91E-10	798,9637481	1,242032506
20	2,62E-09	5,54E-10	373,5596433	1,120990817
30	2,63E-09	8,16E-10	222,0610518	0,899103632

40	2,64E-09	1,08E-09	144,279209	0,673375911
50	2,64E-09	1,34E-09	97,15660874	0,335107923
60	2,65E-09	1,60E-09	65,26968631	0,109304439
66	2,65E-09	1,76E-09	50,52584834	0,088581599
70	2,65E-09	1,87E-09	42,08300831	0,071822176
76	2,65E-09	2,02E-09	31,05788553	0,049091056
80	2,65E-09	2,13E-09	24,61352072	0,034283374
86	2,65E-09	2,29E-09	16,0430425	0,023771182
90	2,65E-09	2,39E-09	10,95877967	0,014690947
96	2,65E-09	2,55E-09	4,10998838	0,008466669
100	2,65E-09	2,65E-09	0,003784751	0,003784751

Таблица 12

Значения модуля вектора перемещения точек В в плоскости XY и погрешность значений между User300 и Plane142 (с одной вставкой)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «В»	Модуль вектора перемещения для Plane142 в точке «В»	Погрешность между результатами решения на одинаковой сетке в точке «В», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «В», %
10	2,35E-09	2,53E-10	830,3905162	1,663450074
20	2,36E-09	4,91E-10	381,4186947	1,24609953
30	2,37E-09	7,29E-10	225,2546168	0,954594322
40	2,38E-09	9,67E-10	145,8511078	0,704471681
50	2,38E-09	1,20E-09	97,89015206	0,412796773
60	2,39E-09	1,44E-09	65,67811743	0,162325792
66	2,39E-09	1,59E-09	50,8553332	0,101790672
70	2,39E-09	1,68E-09	42,38497825	0,052484292
76	2,39E-09	1,82E-09	31,25765018	0,039014315
80	2,39E-09	1,92E-09	24,75873566	0,030439474
86	2,39E-09	2,06E-09	16,12514812	0,026393469
90	2,39E-09	2,16E-09	11,00531398	0,022833219
96	2,39E-09	2,30E-09	4,124705752	0,01311203

100	2,39E-09	2,39E-09	0,003594918	0,003594918
-----	----------	----------	-------------	-------------

Таблица 13

Значения модуля вектора перемещения точек D в плоскости XY и погрешность значений между User300 и Plane142 (с одной вставкой)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «D»	Модуль вектора перемещения для Plane142 в точке «D»	Погрешность между результатами решения на одинаковой сетке в точке «D», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «D», %
10	4,84E-09	5,10E-10	849,8250438	1,748237213
20	4,85E-09	1,00E-09	385,1131979	1,477997483
30	4,87E-09	1,49E-09	226,51116	1,159967931
40	4,88E-09	1,98E-09	146,41993	0,857073972
50	4,90E-09	2,47E-09	98,16760793	0,530276769
60	4,91E-09	2,96E-09	65,78721107	0,269088778
66	4,92E-09	3,26E-09	51,04945244	0,10699307
70	4,92E-09	3,45E-09	42,53134504	0,061178678
76	4,92E-09	3,75E-09	31,35769267	0,045062139
80	4,92E-09	3,94E-09	24,83622703	0,033509338
86	4,93E-09	4,24E-09	16,17841249	0,023020243
90	4,93E-09	4,44E-09	11,04663424	0,014635499
96	4,93E-09	4,73E-09	4,141603313	0,007777875
100	4,93E-09	4,93E-09	0,003225118	0,003225118

Таблица 14

Значения модуля вектора перемещения точек A в плоскости XY и погрешность значений между User300 и Plane142 (с двумя вставками)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «A»	Модуль вектора перемещения для Plane142 в точке «A»	Погрешность между результатами решения на одинаковой сетке в точке «A», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «A», %
50	2.398142454E-09	1.259761750E-09	90.36476175	0.380996376
60	2.403412457E-09	1.5235065460E-09	57.75530885	0.162080087
70	2.405687456E-09	1.716678075E-09	40.13620207	0.067576471

80	2.406419786E-09	1.980305891E-09	21.51757955	0.037155415
90	2.406842153E-09	2.214154015E-09	8.702562545	0.019610277
100	2.407014575E-09	2.407314234E-09	0.012447856	0.012447856

Таблица 15

Значения модуля вектора перемещения точек В в плоскости ХУ и погрешность значений между User300 и Plane142 (с двумя вставками)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «В»	Модуль вектора перемещения для Plane142 в точке «В»	Погрешность между результатами решения на одинаковой сетке в точке «В», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «В», %
50	2.650167894E-09	1.398425434E-09	89.51084767	0.471585066
60	2.655245786E-09	1.687773516E-09	57.3223991	0.280882226
70	2.660126789E-09	1.897481615E-09	40.1924935	0.097573654
80	2.660717845E-09	2.186758763E-09	21.67404517	0.075376243
90	2.662104574E-09	2.453014139E-09	8.523816935	0.023296939
100	2.662431254E-09	2.662724907E-09	0.011028289	0.011028289

Таблица 16

Значения модуля вектора перемещения точек D в плоскости ХУ и погрешность значений между User300 и Plane142 (с двумя вставками)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «D»	Модуль вектора перемещения для Plane142 в точке «D»	Погрешность между результатами решения на одинаковой сетке в точке «D», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «D», %
50	4.944754564E-09	2.590712884E-09	90.86463014	0.548140653
60	4.954124578E-09	3.179419951E-09	55.8185032	0.359685331
70	4.966941140E-09	3.561063362E-09	39.47915651	0.101910975
80	4.969045758E-09	4.151019187E-09	19.70664394	0.05958164
90	4.970413246E-09	4.591244434E-09	8.258519385	0.032077912
100	4.971407541E-09	4.972008162E-09	0.012080045	0.012080045

Таблица 17

Значения модуля вектора перемещения точек А в плоскости ХУ и погрешность значений между User300 и Plane142 (с тремя вставками)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «А»	Модуль вектора перемещения для Plane142 в точке «А»	Погрешность между результатами решения на одинаковой сетке в точке «А», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «А», %
50	2.395145215E-09	1.211086198E-09	97.768351952	0.474398579
60	2.400123648E-09	1.528753121E-09	56.998773381	0.267529457
70	2.402536589E-09	1.760925507E-09	36.436015007	0.167264386
80	2.405134787E-09	2.009708665E-09	19.675793257	0.059301321
90	2.405746847E-09	2.240891850E-09	7.356669042	0.033868358
100	2.406123648E-09	2.406561910E-09	0.018211125	0.018211125

Таблица 18

Значения модуля вектора перемещения точек В в плоскости ХУ и погрешность значений между User300 и Plane142 (с тремя вставками)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «В»	Модуль вектора перемещения для Plane142 в точке «В»	Погрешность между результатами решения на одинаковой сетке в точке «В», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «В», %
50	2.647845123E-09	1.345819276E-09	96.7459650949	0.545631554946
60	2.652494856E-09	1.684431857E-09	57.4711879841	0.370985290731
70	2.658534615E-09	1.951511497E-09	36.2295133330	0.144128964548
80	2.660294512E-09	2.208458407E-09	20.4593441094	0.078026365441
90	2.661564875E-09	2.475036399E-09	7.5363932456	0.030310904758
100	2.662046576E-09	2.662371864E-09	0.0122179777	0.012217977676

Таблица 19

Значения модуля вектора перемещения точек D в плоскости XY и погрешность значений между User300 и Plane142 (с тремя вставками)

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «D»	Модуль вектора перемещения для Plane142 в точке «D»	Погрешность между результатами решения на одинаковой сетке в точке «D», %	Погрешность между точным решением Plane142 и решением на определенной сетке User300 в точке «D», %
50	4.940334575E-09	2.493865200E-09	98.09950333	0.5713548424621
60	4.950622457E-09	3.204016831E-09	54.51299784	0.3643020298901
70	4.963713457E-09	3.637560774E-09	36.45719659	0.1008339723677
80	4.965934457E-09	4.161572020E-09	19.32833153	0.0561343559118
90	4.967012468E-09	4.594420065E-09	8.10967211	0.0344384258726
100	4.967954785E-09	4.968723618E-09	0.01547345	0.0154734507111

Подобные расчеты проделаны для кубического конечного элемента, в том числе разработаны функции и процедуры для расчета вспомогательных функций форм на основе ортогональных финитных функций в трехмерной системе координат. Для получения сравнительных результатов рассматривались 2 варианта куба: однородный и с одной вставкой. Свойства материалов были аналогичные как при расчете квадратной пластины, нагрузки и фиксация повторяли указанные в первой задаче. Рассмотрим результаты по затрачиваемому вычислительному времени на получение решения для однородного куба и с одной вставкой (Таблица 20 – 21, Рисунок 3.11 – 3.12)

Таблица 20

Время, затрачиваемое на выполнение вычислительных операций для конечных элементов User300 и Solid185 в однородном кубе

Количество элементов на ребре	CPU User300	CPU Solid185	Во сколько раз быстрее
10	4,745	8,969	1,890200211
20	9,474	20,797	2,195165717
30	21,974	64,484	2,934559024
40	34,113	84,609	2,480256794
50	57,642	128,516	2,229554838
60	79,025	212,406	2,687832964
70	172,784	507,328	2,936197796
80	350,315	1067,438	3,047080485

90	611,674	1986,062	3,24692892
100	1107,354	3817,609	3,447505495

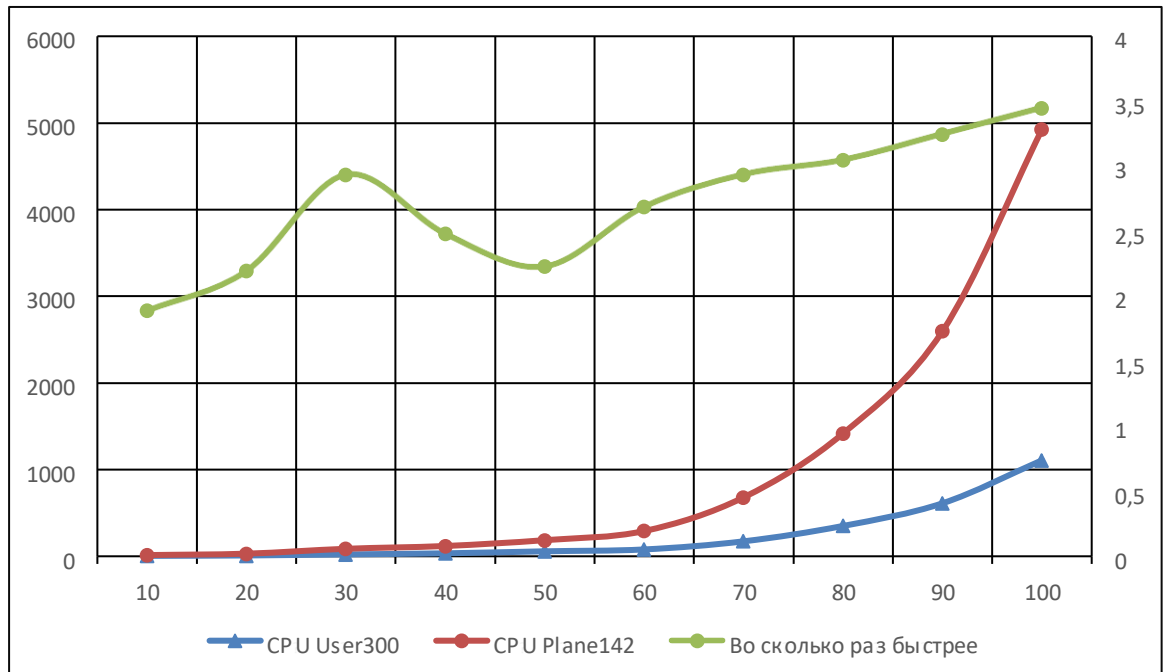


Рисунок 3.11. График затрачиваемого времени на расчет однородного куба

Таблица 21

Время, затрачиваемое на выполнение вычислительных операций для конечных элементов User300 и Solid185 в кубе с одной вставкой

Количество элементов на ребре	CPU User300	CPU Solid185	Во сколько раз быстрее
10	14,658	16,641	1,135284486
20	34,248	46,781	1,365948377
30	51,671	80,625	1,560353003
40	67,684	130,562	1,928993558
50	82,107	185,734	2,26209702
60	125,347	308,422	2,460545526
70	287,647	765,047	2,659673141
80	541,374	1586,438	2,930391929
90	951,764	3032,172	3,18584439
100	1880,467	6176,734	3,28468088

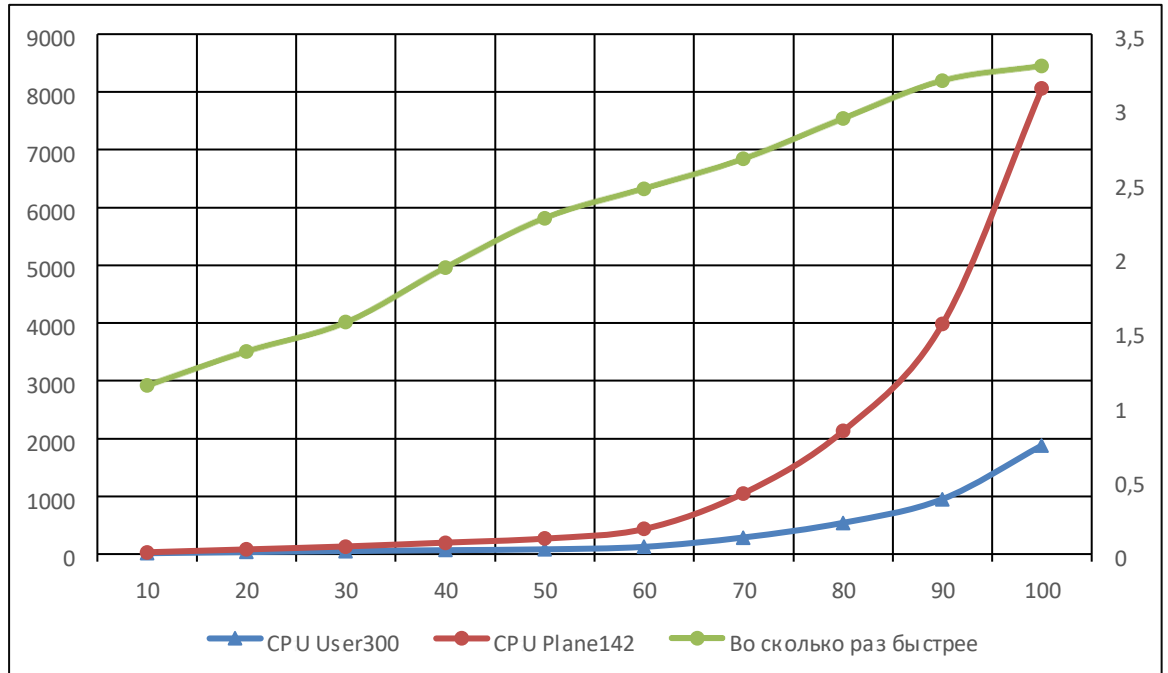


Рисунок 3.12. График затрачиваемого времени на расчет куба с одной вставкой

Проведен сравнительный анализ модуля вектора перемещения выбранных точек на исследуемом теле с применением кубического конечного элемента (Рисунок 3.13).

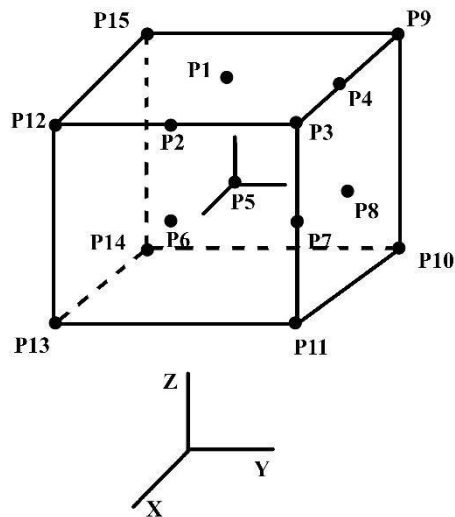


Рисунок 3.13. Расположение контрольных точек на кубе.

Результаты по модулю вектора перемещения представлены в таблицах 22 - 34. Изображение результатов в виде графиков по перемещению не предоставляют информативных данных.

Значения модуля вектора перемещения точек P1 и погрешность значений между User300 и Solid185

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P1»	Модуль вектора перемещения для Solid185 в точке «P1»	Погрешность между результатами решения на одинаковой сетке в точке «P1», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P1», %
10	1,143636555E-07	1,460654697E-09	7729,616115	9,169794537
20	1,150877412E-07	5,392669203E-09	2034,151696	8,594709279
30	1,159555441E-07	1,179830395E-08	882,8153658	7,90548057
40	1,176987455E-07	2,067792746E-08	469,1999151	6,520990523
50	1,197631446E-07	3,203155741E-08	273,8911069	4,881398057
60	1,215822546E-07	4,585917880E-08	165,1208717	3,436619662
70	1,223665452E-07	6,216077468E-08	96,85492311	2,813718312
80	1,236784452E-07	8,093633057E-08	52,80955335	1,771777587
90	1,248334858E-07	1,021858347E-07	22,16320022	0,854418181
100	1,256521146E-07	1,259092776E-07	0,204244703	0,204244703

Значения модуля вектора перемещения точек P3 и погрешность значений между User300 и Solid185

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P3»	Модуль вектора перемещения для Solid185 в точке «P3»	Погрешность между результатами решения на одинаковой сетке в точке «P3», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P3», %
10	2,5846136456E-07	4,9663429760E-09	5104,25926701040	10,25153148150
20	2,6064946754E-07	1,5737896890E-08	1556,18995575971	9,49173168802
30	2,6484675643E-07	3,1802622870E-08	732,78274723635	8,03425950278
40	2,6697564356E-07	5,3061228040E-08	403,14637150641	7,29502189993
50	2,7064894657E-07	7,9466697870E-08	240,58159433371	6,01949926964
60	2,7469485764E-07	1,1099135440E-07	147,49212146113	4,61459171921
70	2,7986497864E-07	1,4761690580E-07	89,58870403514	2,81931201621
80	2,8279764316E-07	1,8933034590E-07	49,36730919288	1,80096968406
90	2,8543197256E-07	2,3612194590E-07	20,88328828227	0,88622163284
100	2,8728565412E-07	2,8798415040E-07	0,24254677871	0,24254677871

Таблица 24

Значения модуля вектора перемещения точек P4 и погрешность значений между User300 и Solid185

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P4»	Модуль вектора перемещения для Solid185 в точке «P4»	Погрешность между результатами решения на одинаковой сетке в точке «P4», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P4», %
10	2,341981646E-07	3,429450208E-09	6729,02944774290	10,05533484100
20	2,360946424E-07	1,210478277E-08	1850,42444668340	9,32698559131
30	2,384643125E-07	2,580776449E-08	824,00220309047	8,41690509035
40	2,403164564E-07	4,449183732E-08	440,13605837755	7,70558240705
50	2,446132347E-07	6,813432365E-08	259,01616328733	6,05538894110
60	2,493164895E-07	9,672191304E-08	157,76629269305	4,24908665767
70	2,526498765E-07	1,302458155E-07	93,97926565710	2,96888714260
80	2,559495764E-07	1,686997836E-07	51,71897140003	1,70162524935
90	2,581346575E-07	2,120791452E-07	21,71619100811	0,86243685629
100	2,597713646E-07	2,603802729E-07	0,23385348407	0,23385348407

Таблица 25

Значения модуля вектора перемещения точек P5 и погрешность значений между User300 и Solid185

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P5»	Модуль вектора перемещения для Solid185 в точке «P5»	Погрешность между результатами решения на одинаковой сетке в точке «P5», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P5», %
10	1,012584563E-07	1,260309442E-09	7934,41225825554	10,12439367635
20	1,020669844E-07	4,745870372E-09	2050,64838269038	9,40675530941
30	1,026325987E-07	1,045441407E-08	881,71545543155	8,90472387406
40	1,046325987E-07	1,838625848E-08	469,08042935335	7,12955151596
50	1,057456354E-07	2,854137552E-08	270,49943625142	6,14163552427
60	1,075546585E-07	4,091973142E-08	162,84302161238	4,53597162313
70	1,094954414E-07	5,552130039E-08	97,21339484282	2,81335958006
80	1,106656946E-07	7,234606326E-08	52,96712712934	1,77465901375
90	1,116746654E-07	9,139400557E-08	22,19036106746	0,87911044111
100	1,123649866E-07	1,126651162E-07	0,26639084494	0,26639084494

Значения модуля вектора перемещения точек P8 и погрешность значений между User300 и Solid185

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P8»	Модуль вектора перемещения для Solid185 в точке «P8»	Погрешность между результатами решения на одинаковой сетке в точке «P8», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P8», %
10	2,170584750E-07	2,616403948E-09	8196,06128541127	10,17234194177
20	2,186536521E-07	1,002188354E-08	2081,76205368277	9,51219252041
30	2,206985452E-07	2,221435023E-08	893,49538885883	8,66593229233
40	2,225658365E-07	3,919529272E-08	467,83817916592	7,89317092288
50	2,273657845E-07	6,096472687E-08	272,94644986244	5,90675647702
60	2,312659867E-07	8,752264095E-08	164,23561279660	4,29269361957
70	2,350655470E-07	1,188690188E-07	97,75173495417	2,72028043510
80	2,374478540E-07	1,550038457E-07	53,18836311943	1,73438454420
90	2,395685413E-07	1,959271094E-07	22,27432027331	0,85675758612
100	2,410548000E-07	2,416388000E-07	0,24168304097	0,24168304097

Таблица 27

Значения модуля вектора перемещения точек P1 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P1»	Модуль вектора перемещения для Solid185 в точке «P1»	Погрешность между результатами решения на одинаковой сетке в точке «P1», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P1», %
10	1,143543125E-07	2,391170454E-09	4682,357203	9,160733035
20	1,150643122E-07	5,417978169E-09	2023,750015	8,59673286
30	1,159124376E-07	1,185373738E-08	877,8556242	7,923010145
40	1,176643246E-07	2,077546896E-08	466,3618222	6,531369372
50	1,192456492E-07	3,206497543E-08	271,8875426	5,275217651
60	1,212346579E-07	4,586497842E-08	164,3294778	3,695215143
70	1,223345649E-07	6,215842135E-08	96,81092634	2,821485636
80	1,236243658E-07	8,093546135E-08	52,74437649	1,796910617
90	1,248645243E-07	1,021761346E-07	22,20517529	0,811769858
100	1,256822464E-07	1,258864325E-07	0,162198623	0,162198623

Таблица 28

Значения модуля вектора перемещения точек P2 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P2»	Модуль вектора перемещения для Solid185 в точке «P2»	Погрешность между результатами решения на одинаковой сетке в точке «P2», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P2», %
10	1,233213465E-07	1,674604456E-09	7264,20746848913	10,58788235772
20	1,244754613E-07	6,051304464E-09	1957,00212376932	9,75111030053
30	1,257366795E-07	1,304656756E-08	863,75294820635	8,83668475402
40	1,277546135E-07	2,290344568E-08	457,79647857807	7,37361489722
50	1,284616495E-07	3,521345677E-08	264,80840351983	6,86099005738
60	1,324546456E-07	5,040446547E-08	162,78355384400	3,96593372034
70	1,337996735E-07	6,822464677E-08	96,11633009530	2,99074338767
80	1,350137567E-07	8,880137676E-08	52,04016157359	2,11049226356
90	1,367046466E-07	1,120134677E-07	22,04304485703	0,88454032313
100	1,376534456E-07	1,379246457E-07	0,19662912978	0,19662912978

Таблица 29

Значения модуля вектора перемещения точек P3 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P3»	Модуль вектора перемещения для Solid185 в точке «P3»	Погрешность между результатами решения на одинаковой сетке в точке «P3», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P3», %
10	2,5841326545E-07	4,9672314654E-09	5102,36005199308	10,15817594475
20	2,6063548747E-07	1,5715668795E-08	1558,44349911395	9,38558217191
30	2,6480365746E-07	3,1821335975E-08	732,15757398359	7,93644605963
40	2,6693265649E-07	5,3035646547E-08	403,30800861114	7,19626286463
50	2,7060334132E-07	7,9423498465E-08	240,70942044872	5,92008603810
60	2,7461357465E-07	1,1091649435E-07	147,58587642839	4,52585932728
70	2,7980021655E-07	1,4762678947E-07	89,53214220730	2,72263390918
80	2,8279134675E-07	1,8903498462E-07	49,59736014364	1,68271595933
90	2,8540341267E-07	2,3603498765E-07	20,91572335124	0,77458623485

Таблица 30

Значения модуля вектора перемещения точек P4 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P4»	Модуль вектора перемещения для Solid185 в точке «P4»	Погрешность между результатами решения на одинаковой сетке в точке «P4», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P4», %
10	2,340764568E-07	3,436479464E-09	6711,51915040223	10,06168928989
20	2,360736675E-07	1,211761316E-08	1848,18620027702	9,29430859465
30	2,384606495E-07	2,582649746E-08	823,31780489319	8,37716753807
40	2,401594364E-07	4,453625740E-08	439,24476464877	7,72444906629
50	2,446426795E-07	6,813158746E-08	259,07379905579	6,00186955869
60	2,493116499E-07	9,672048965E-08	157,76508244395	4,20792873590
70	2,526402177E-07	1,302355442E-07	93,98714791362	2,92900572430
80	2,559135975E-07	1,686761649E-07	51,71888548196	1,67128741210
90	2,580356949E-07	2,123521456E-07	21,51310934327	0,85592194761
100	2,597426795E-07	2,602633460E-07	0,20005373327	0,20005373327

Таблица 31

Значения модуля вектора перемещения точек P5 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P5»	Модуль вектора перемещения для Solid185 в точке «P5»	Погрешность между результатами решения на одинаковой сетке в точке «P5», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P5», %
10	1,012236564E-07	1,271039875E-09	7863,84586043533	10,08227968182
20	1,020351453E-07	4,775035845E-09	2036,84564021419	9,36142817741
30	1,026026541E-07	1,053911334E-08	873,54161398804	8,85730587649
40	1,041365846E-07	1,851974315E-08	462,30037178724	7,49470415373
50	1,057823242E-07	2,854723641E-08	270,55189050225	6,03278148715
60	1,075036875E-07	4,098136413E-08	162,32335066474	4,50368174985
70	1,094136975E-07	5,552813647E-08	97,04190417993	2,80700585757
80	1,106603985E-07	7,239364134E-08	52,85927940049	1,69955214974
90	1,116197745E-07	9,134697667E-08	22,19317872102	0,84733139351
100	1,123641999E-07	1,125736463E-07	0,18605281274	0,18605281274

Таблица 32

Значения модуля вектора перемещения точек P6 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P6»	Модуль вектора перемещения для Solid185 в точке «P6»	Погрешность между результатами решения на одинаковой сетке в точке «P6», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P6», %
10	1,143631347E-07	1,473136846E-09	7663,239038	9,165402568
20	1,150871347E-07	5,421791460E-09	2022,677265	8,590354949
30	1,159503690E-07	1,183645916E-08	879,6035061	7,904718451
40	1,176916548E-07	2,080946107E-08	465,5680097	6,521676642
50	1,197600649E-07	3,202368492E-08	273,9734052	4,878811541
60	1,215898642E-07	4,581365975E-08	165,4008976	3,425466629
70	1,223031649E-07	6,216649844E-08	96,73484588	2,858917095
80	1,236136947E-07	8,093264946E-08	52,73649815	1,818009609
90	1,248019847E-07	1,021821265E-07	22,13680513	0,874192827
100	1,256036189E-07	1,259026163E-07	0,237483047	0,237483047

Таблица 33

Значения модуля вектора перемещения точек P7 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P7»	Модуль вектора перемещения для Solid185 в точке «P7»	Погрешность между результатами решения на одинаковой сетке в точке «P7», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P7», %
10	2,341212346E-07	3,452705276E-09	6680,80565484095	10,08488009001
20	2,360965431E-07	1,217214556E-08	1839,64607107442	9,32625559899
30	2,384665431E-07	2,596464992E-08	818,42772330127	8,41604839435
40	2,403645643E-07	4,477819250E-08	436,78934070418	7,68710639138
50	2,446645310E-07	6,813432365E-08	259,09145037796	6,03568838951
60	2,493145621E-07	9,672191304E-08	157,76430004119	4,24982685622
70	2,526496453E-07	1,302458155E-07	93,97908819036	2,96897591392
80	2,559645312E-07	1,686997836E-07	51,72783614288	1,69588180273
90	2,581365432E-07	2,120791452E-07	21,71708019031	0,86171261940
100	2,597924971E-07	2,603802729E-07	0,22573744295	0,22573744295

Значения модуля вектора перемещения точек P8 и погрешность значений между User300 и Solid185 для куба со вставкой

Количество элементов на ребре	Модуль вектора перемещения для User300 в точке «P8»	Модуль вектора перемещения для Solid185 в точке «P8»	Погрешность между результатами решения на одинаковой сетке в точке «P8», %	Погрешность между точным решением Solid185 и решением на определенной сетке User300 в точке «P8», %
10	2,170584750E-07	2,634912811E-09	8137,78586121877	10,17234194177
20	2,186536521E-07	1,009464950E-08	2066,03510741012	9,51219252041
30	2,206985452E-07	2,237637321E-08	886,30168137064	8,66593229233
40	2,225658365E-07	3,948214129E-08	463,71267927247	7,89317092288
50	2,273657845E-07	6,096472687E-08	272,94644986244	5,90675647702
60	2,312659867E-07	8,752264095E-08	164,23561279660	4,29269361957
70	2,350655470E-07	1,188690188E-07	97,75173495417	2,72028043510
80	2,374478540E-07	1,550038457E-07	53,18836311943	1,73438454420
90	2,395685413E-07	1,959271094E-07	22,27432027331	0,85675758612
100	2,410548000E-07	2,416388000E-07	0,24168304097	0,24168304097

3.6 Выводы

В результате проведенного сравнительного анализа результатов применения 2D и 3D конечного элемента (КЭ) с ОФФ и стандартного конечного элемента можно сделать вывод о том, что применение нового КЭ снижает потребность в высоких вычислительных мощностях системы за счет получения результатов по модулю вектора перемещения точек на исследуемом теле на густой сетке меньше 1%. При этом скорость расчета в три раза выше в сравнении со стандартными вычислениями. Точность получаемых результатов также выше, чем при использовании КЭ ANSYS.

Можно сделать вывод о том, что применение квадратных или кубических КЭ с ОФФ позволяет снизить нагрузку на вычислительную машину в плане хранения и расчета большого количества линейных уравнений, при этом размеры КЭ влияют на точность получаемых результатов по сравнению со стандартными от 10% до 0.3%.

Применение нового КЭ сокращает время перемещения лопасти в воздушном пространстве под действием окружающих потоков, а также сокращает время на сравнение частот свободных колебаний с колебаниями лопасти. Однако недостатком является сложность внедрения КЭ в программный комплекс ANSYS и его использование с тонкими настройками сетки конечных элементов на исследуемом теле, для устранения данной проблемы в следующей главе будет рассматриваться разработка автоматизированной программной среды, ориентированной на применение нового метода моделирования динамики лопастей винта с газами с использованием нового КЭ [79].

ГЛАВА 4. ПРОГРАММНАЯ СРЕДА РЕШЕНИЯ ЗАДАЧ ДИНАМИКИ ВИНТОВ И ЕЕ ИСПОЛЬЗОВАНИЕ ПРИ ПРОЕКТИРОВАНИИ

4.1 Обзор способов внедрения программной среды расчета динамики винтов в программный комплекс ANSYS

Программный комплекс ANSYS представляет собой набор большого количества отдельных вычислительных модулей, которые включают в себя препроцессоры и решатели.

Первоначально программный комплекс работал в консольном режиме с интерфейсом написанным на распространенном в момент появления комплекса на рынке на языке Tcl/Tk, в современном комплексе была создана новая интерфейсная оболочка с более удобным представлением вычислительных модулей с необходимыми компонентами. Однако первоначальные файлы и библиотеки остаются в исполнительных директивах современного комплекса и становится возможно внедрять свои коды внутрь его с помощью оболочечного языка программирования Tcl/Tk, а также с помощью языка высокого уровня Fortran, который используется по сей день для формирования сложных вычислительных алгоритмов благодаря своей простоте общения напрямую с вычислительным процессором машины [80].

В данной подглаве рассмотрены два способа внедрения в программный комплекс своих систем:

1. С помощью устанавливаемых средств Application Customization Toolkit.
2. С помощью создания собственной оболочки, которая будет формировать пакеты данных и отправлять их на исполнительный файл ANSYS.

При использовании первого способа требуется наличие лицензионного соглашения на внедрение собственных разработок в программный комплекс. Однако, в том случае если не используются исходные коды программного комплекса, имеется возможность создания двух основных файлов: *.xml и *.py.

Первый формат файла представляет собой иерархическое дерево расположения элементов интерфейса с заданием их имен и процедур, которые они должны выполнять при нажатии, вводе текста или выделении. Второй форма является файлов, написанным на языке Python, в котором содержатся информация о процедурах, описанных ранее в предыдущем файле.

Принцип работы данной связки заключается в том, что, формируя файл xml, создается расположение кнопок, полей, таблиц и новых модулей внутри программного комплекса. Но разместив в директиве комплекса данный файл, он не будет задействован до тех пор, пока не будет подключен в специальном окне дополнений (Рисунок 4.1).

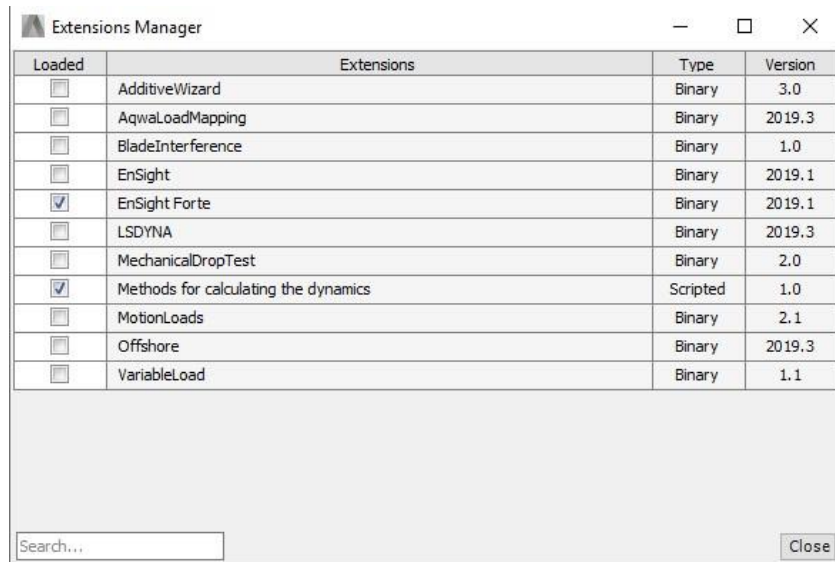


Рисунок 4.1. Окно подключения дополнительных функций, элементов и процедур.

После того как будет подключен файл xml, описанные в нем элементы интерфейса появятся только в тех модулях, в которых были указаны. Например, кнопка демонстрации версии файла, размещенная в Static Structure, не будет показана в модуле Transient Structure. Затем требуется расположить в дочерней папке по отношению к файлу xml, процедурный файл с расширением «ру». При формировании процедур следует правильно прописывать имена обращения к переменным оболочки программного комплекса и стараться не создавать конфликтующие функции и переменные, иначе может возникнуть поломка интерфейса и возникнуть непредвиденные ошибки при работе комплекса.

Второй способ внедрения представляет собой создание собственной оболочки для формирования пакета данных о всех шагах, которые необходимо проделать программном комплексе ANSYS, от подготовки материала до получения результатов расчета.

Оболочка может быть описана на любом доступном языке программирования. Основной особенностью является отправка пакета данных на исполнительный файл программного комплекса RunW.exe. При этом стоит учитывать, что при отправке пакета на исполняемый файл происходит запуск ядра программного комплекса и запускается интерфейсная оболочка, с указанием верных параметров при отправке пакета позволяет запускать программный комплекс в фоновом режиме, не отвлекая всплывающими окнами расчета.

В этом случае пакет данных содержит в себе блоки кода оформленные на разных языках программирования: JavaScript – формирование геометрических параметров геометрии в модуле Geometry; Python – формирование процедур и функций; XML – для формирования интерфейса в программном комплексе; SCL – формирование входного файла для модуля CFX.

В результате можно сделать вывод, что второй вариант является простым и доступным. При решении задач по новым методикам конечному пользователю не

требуются знания о том, как ему внедрить и запустить необходимые для работы методики файлы. Конечный пользователь запускает программную среду, в которой формируется пакет данных с выбранным типом решаемой задачи, и в автоматическом режиме этот пакет отправляется на исполняемый файл программного комплекса, и пользователю остается только ожидать окончания расчета и анализ результатов. В данной главе будет рассмотрено создание программной среды, которая обладает пошаговым интерфейсом выбора типа решаемой задачи, импорт геометрии, настройка материалов, наложение сетки конечных элементов, постановка задачи, настройка решателя и демонстрация результатов в виде таблиц и графиков.

4.2 Общая структура программной среды решения технических задач динамики упругих винтов

Общая структура программной среды представляет собой набор интерфейсов, которые выполняют каждый свою функцию и формируют блок для пакета данных на необходимом языке программирования.

Файловая структура оформлена в основном на языке программирования Visual Basic благодаря своей доступности и простоте, а также с внедрением части кода на языке Python, для формирования отдельных элементов программной среды, направленных на повышение её функционала (Рисунок 4.2).

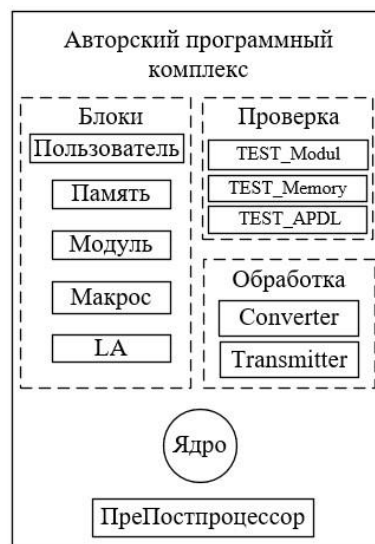


Рисунок 4.2 - Структура авторского программного комплекса

Основой программной среды является ядро и автономные модули, функционирование которых обеспечивается программными библиотеками. За обработку и хранение данных отвечает модуль блока памяти, предназначенный для формирования параметров постановки задачи на основании больших данных по результатам прошлых расчетов. Данный блок выдает ядру рекомендации для формирования вычислительных модулей внутри программного комплекса ANSYS или внесения изменения в собственный

алгоритм при помощи JIT-компиляции для увеличения числа различных технических задач.

Выполнена программная реализация ряда основных компонентов программного комплекса: Ядро; блоки управления; блок проверки; блок обработки; препостпроцессор.

Блоки управления представляют собой автономные структуры, содержащие в себе алгоритмы управления отдельными элементами программной среды. Их автономность заключается в обработке информации независимо от других структур программной среды.

Блок «Пользователь» формирует формы окон интерфейса получения и передачи информации между пользователем и программной средой. Принцип формирования таких форм основывается на изменении структуры препостпроцессора в зависимости от выбранного типа технической задачи и вводимых пользователем данных.

Блок «Память» создает, дополняет и обрабатывает базу данных входных параметров по различным типам технических задач, тем самым определяет связь постановки задачи и конечных результатов. Кроме этого, блок «памяти» в процессе работы авторского программного комплекса формирует вспомогательные алгоритмы и внедряет их в структуру препостпроцессора для реализации выдачи рекомендаций о влиянии того или иного параметра на конечный результат.

Блок «Модуль» отвечает за реализацию работы нового метода расчета динамического взаимодействия твердого тела с окружающей средой на основе применения КЭ (ОФФ). Блок реализует пакеты данных, которые хранят в себе алгоритмы создания модулей внутри программного комплекса ANSYS WorkBench, необходимые для решения выбранного типа технической задачи в зависимости от версии конечного программного комплекса.

Блок «Макрос» реализует алгоритмы формирования и обработки информации, которая образуется в процессе работы блока «модуль» внутри программного комплекса ANSYS WorkBench. Блок «макрос» наполняет пакеты данных алгоритмами передачи данных как внутри ANSYS, так и между ANSYS и авторским программным комплексом.

Блок «LA» хранит в себе алгоритмы внедрения КЭ (ОФФ) в процессе наложения сетки конечных элементов на твердое тело для определения влияния резонансных колебаний в процессе динамического взаимодействия с окружающей средой.

Следующим важным компонентом комплекса является «Ядро», которое отвечает за формирование стека данных, которые приходят от управляющих блоков, в зависимости от приоритета, которые выставляет препостпроцессор, и оно обрабатывает ту или иную задачу. Кроме этого, «Ядро» реализует принцип JIT-компиляции (изменение кода), то есть позволяет вносить изменения в авторскую структуру динамических библиотек, в которых хранятся все исполняющие процедуры и функции программного комплекса в процессе его работы без прерывания на обновления кода.

Компонент «Проверка» состоит из двух модулей «Converter» и «Transmitter». Первый модуль отвечает за проверку правильности преобразования параметров, вводимых пользователем, и параметров из базы данных, которые либо заносятся в неё,

либо выгружаются для формирования рекомендаций. Работа данного модуля также заключается в проверке формирования авторских динамических библиотек. Так как пользователь может выбрать вариант решения своей собственной технической задачи, то модуль обязан правильно обработать полученные данные и дополнить ими алгоритм для внесения в динамическую библиотеку. Второй модуль осуществляет передачу данных между блоками-препостпроцессор-ядром, а также между авторским программным комплексом и ANSYS WorkBench.

Работа компонента «Проверка» состоит в формировании отчета по работе блоков «Память», «Модуль», «LA». Компонент проводит сравнение получаемого отчета с эталонной формой, которая характеризует правильность сформированного алгоритма работы с данными при решении технической задачи. При возникновении ошибок формируется запрос автору программного комплекса с выявленными отклонениями в структуре работы среды, это может быть связано с тем, что «Ядро» имеет наивысший приоритет и может вносить изменения в алгоритмы работы среды без ведома пользователя или автора.

Последним компонентом комплекса является «ПреПостпроцессор», который работает с выводом информации пользователю, формированием интерфейса, выставлением приоритета задач обработки информации от пользователя для «Ядра», а также отвечает за анализ и вывод результатов расчета из программного комплекса ANSYS WorkBench.

База данных представляет собой многомерную матрицу в двоичном оформлении, что позволяет уменьшить объем информации, хранимый на информационном носителе. Поиск параметров по множеству результатов осуществляется на основе псевдо-искусственного алгоритма с построением нейронной сети, которая формирует графы поиска информации с записью расположения того или иного параметра в базе данных. Каждому параметру выдается весовой коэффициент, обозначающий его влияние на постановку задачи и получаемые результаты решения типовой технической задачи.

4.3 Обзор функций и возможностей программной среды и её детализация

Применение языков программирования для работы программной среды будет рассмотрено с формирования блока «Пользователь» на языке программирования Visual Basic .NET, который начинает действовать с запуска среды.

Первым шагом работы блока «Пользователь» является формирование системных переменных, которые хранят в себе данные о расположении исполнительных файлов комплекса, название проекта с решаемой задачей и расположение рабочей папки. Для упрощения нахождения исполнительного файла может быть задействован автоматический поиск, который отыскивает необходимый файл в папках версии ANSYS на языке программирования VB.net (Рисунок 4.3).

```
Try
    textBox.Text = IO.Directory.GetFiles("C:\Program Files\",
"RunWB2.exe",
```

```

IO.SearchOption.AllDirectories) (0)
Catch ex As Exception
End Try

```

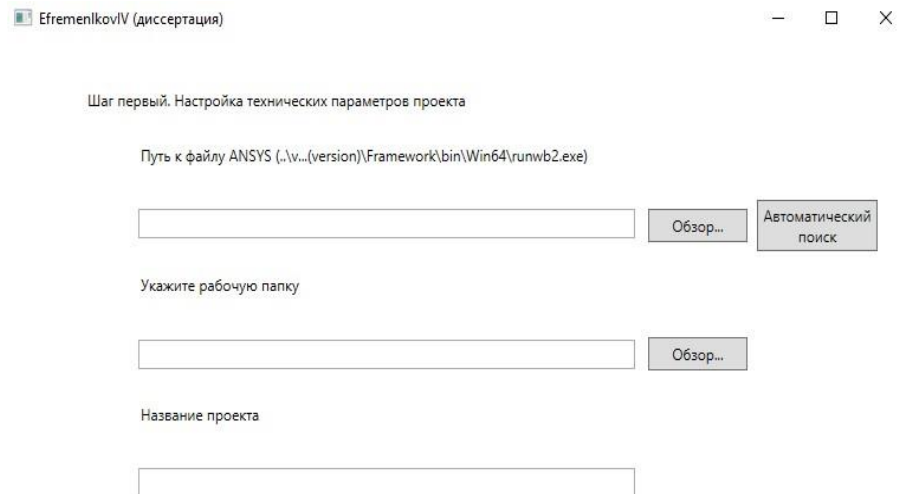


Рисунок 4.3. Интерфейс для ввода системных файлов для дальнейшей работы программной среды

Модуль «Convert», входящий в состав «Проверка», представляется собой процедуру, отлавливающую ошибки при передаче данных в программный комплекс ANSYS. При возникновении ошибки, этот модуль формирует информацию в чем заключается ошибка и обращаясь к блоку «Пользователь» указывает на интерфейс, в котором она возникла. Первая процедура заключается в проверке доступа к рабочей папке, исполняемого файла и названии проекта, проверка осуществляется созданием пустого проекта в этой папке с помощью языка Python.

Как только проект создан блок «Пользователь» переходит на следующий этап работы.

```

WorkingDir = PathJob
WorkingDir = WorkingDir + "\"
PythonFileName = WorkingDir + "Core1.py"
PythonWorkingDir = Replace(WorkingDir, "\", "/")
WBCommand = PathExe + " -B -R " + PythonFileName
NameJob = textBox2.Text

label12.Content = PathJob + " / " + PathExe + " / " + NameJob

Dim FilePyt As New IO.StreamWriter(PythonFileName, False)

FilePyt.WriteLine("# encoding: utf-8")
FilePyt.WriteLine("SetScriptVersion(Version=""15.0"")")
FilePyt.WriteLine("")
FilePyt.WriteLine("workingDir      =      AbsUserPathName(""" +
PythonWorkingDir + """)")
FilePyt.WriteLine("logfile = workingDir + ""WB.log"" ")
FilePyt.WriteLine("paramfilename = workingDir + ""Param.out""")
FilePyt.WriteLine("wbproj = workingDir + "" + NameJob + ".wbpj""")
FilePyt.WriteLine("logfile = open(logfilename, ""w"")")
FilePyt.WriteLine("logfile.write("""Proj Dir: %s\n"" % workingDir)")

```

```

FilePyt.WriteLine("Save ("
FilePyt.WriteLine("FilePath= wbproj,")
FilePyt.WriteLine("Overwrite=True)")
FilePyt.WriteLine("logFile.write ("Save: %s\n" % wbproj)")
FilePyt.WriteLine("logFile.close()")
FilePyt.Close()

Interaction.Shell(WBCommand)

```

По окончании процесса заполнения необходимых полей наступает этап выбора типа решаемой задачи. На данном шаге предлагается выбрать один из возможных вариантов решения технической задачи с подробным описанием того, на что направлена та или иная задача. Для выполнения этих действий задействованы программные элементы **RadioButton**, которые исключают возможность выбора двух и более вариантов решаемых технических задач, что может вызвать нарушение в работе программной среды (Рисунок 4.4).

```

Private Sub radioButton1_Checked(sender As Object, e As RoutedEventArgs)
Handles radioButton1.Checked
Fla = 2
image.Source= New BitmapImage(New Uri("pack://application:,,,/2.JPG")
TBL1.Text = "Описание: Вращение винта задается за счет нагоняемого сверху
потока"
End Sub

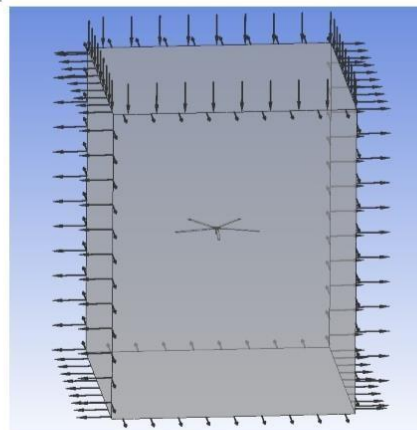
```

EfremenkovIV (диссертация)

Шаг второй. Выбор типа решаемой задачи

- Первый тип
- Второй тип
- Третий тип
- Четвертый тип

Описание: Расчет эмулирует вращение лопастей винта в открытом пространстве, окружающая среда представляет собой куб, нижняя грань которого - земля, боковые - открытые грани, через верхнюю - подается



Назад

Вперед

Выход

Рисунок 4.4. Выбор типа решаемой технической задачи из возможных в программной среде

После выбора типа решаемой задачи фиксируется данный выбор и вносятся необходимые данные в переменную. Эта переменная передается в модуль «ПреПостпроцессор», который в свою очередь комбинирует интерфейсные окна для реализации последовательных действий с целью решения выбранной технической задачи.

Каждое интерфейсное окно имеет двойное обозначение: первое – номер типа решаемой задачи, второе – номер в последовательности действий. По этой причине все интерфейсные окна занесены в двумерный массив данных для быстрого поиска и формирования окон. При дальнейшем развитии программного кода возможно изменить алгоритм формирования интерфейсов не в качестве окон, а в виде отдельных элементов, что во много раз ускорит и оптимизирует процесс работы блока «ПреПостпроцессор». Так как на данный момент ему необходимо расположить на каждом окне интерфейса необходимые элементы, а затем отправить собранный одномерный массив в блок «Проверка», которая запросит эталонный набор через «Ядро» у модуля «Память». Имеется возможность избежать данной проверки, однако в этом случае не гарантируется, что «ПреПостпроцессор» мог ошибиться в формировании окон в следствии неверно введенных данных.

Следующим этапом является загрузка или создание простой исследуемой модели, а также окружающего пространства по вводимым пользователем данных (Рисунок 4.5).

Этап создания и импорта исследуемой модели и окружающей среды.

Объект исследования

Загрузить существующую геометрию винта

Обзор...

Создать геометрию винта по параметрам

Загрузить модель

Окружающая область

Загрузить готовую окружающую область расчета

Обзор...

Создать окружающую область

Загрузить модель

Label

Вперед

Выход

Рисунок 4.5. Интерфейс загрузки или создания моделей для выбранной задачи

Дальнейшие действия зависят от пользователя. При загрузке моделей указывается место расположения цифровой модели и после нажатия кнопки «Загрузить модель» «ПреПостпроцессор» отправляет запрос в «Ядро» о запуске алгоритма формирования кода загрузки цифровой модели в проект ANSYS. Данный алгоритм расположен в блоке «Макросы». Блок «Макросы» представляет собой файл со структурой, которая позволяет удобно хранить внутри себя алгоритмы для выполнения тех или иных действий, в основном это сформированные функции или процедуры, которые добавляются в код работы программной среды через обращение к ним по ссылке.

```
Private Sub button7_Click(sender As Object, e As RoutedEventArgs) Handles
button7.Click
    Dim OFD2 As New Microsoft.Win32.OpenFileDialog()
    OFD2.ShowDialog()
    textBox3.Text = OFD2.FileName
```



```

IssGeomDown = OFD2.FileName
If (IssGeomDown <> " " Or IssGeomDown <> Nothing) Then
    IssCr = 1
End If
End Sub

```

После формирования адреса расположения модели следует пройти этап загрузки модели в программный комплекс ANSYS. Для этого необходимо сформировать пакетный файл на языке программирования Python с помощью функций VB .NET. Для этого проходит запрос введенных данных пользователем на первом шаге, подвергаются обработке, а затем формируется пакетный файл с указанием действий, которые должны быть выполнены внутри программного комплекса ANSYS.

При желании пользователя создать, например, окружающее пространство согласно выбранной технической задаче, тогда блок «Пользователь» выводит отдельное окно, которые показывает какие параметры ему необходимо ввести (Рисунок 4.6).

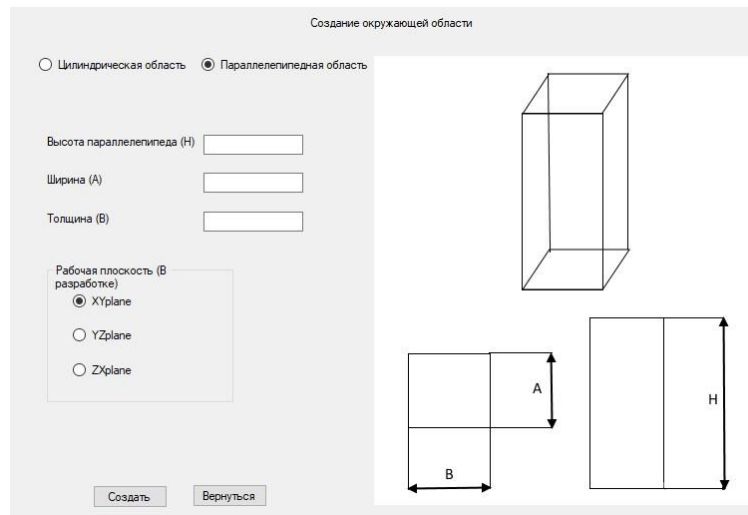


Рисунок 4.6. Создание окружающего пространства согласно типу выбранной задачи

Создание геометрии проводится с помощью языка программирования JavaScript, который работает внутри модуля построения геометрии в программном комплексе ANSYS. После ввода параметров запускается процедура формирования пакетного файла из языков Python и JavaScript для создания выбранной пользователем модели.

```

If (Par.TextBox1.Text <> Nothing And Par.TextBox2.Text <> Nothing And
Par.TextBox3.Text <> Nothing) Then

    PythonFileName2 = Form1.step2.WorkingDir + "Packej.py"
    Form1.step2.PythonWorkingDir
    Replace (Form1.step2.WorkingDir, "\", "/")
    WBCommand2 = PathExe + PythonFileName2
    NamePro = Form1.step2.PythonWorkingDir + Form1.NameP

    H = CStr (Par.TextBox1.Text)
    A = CInt (Par.TextBox2.Text) / 2

```

```

A = CStr(A)
A = Replace(A, ",", ".")
B = CInt(Par.TextBox3.Text) / 2
B = CStr(B)
B = Replace(B, ",", ".")

Dim FilePyt As New IO.StreamWriter(PythonFileName2, False)

FilePyt.WriteLine("# encoding: utf-8")
FilePyt.WriteLine("SetScriptVersion(Version=""15.0"")")
FilePyt.WriteLine("")
FilePyt.WriteLine("Open(FilePath="" + NamePro + ".wbpj"")")
FilePyt.WriteLine("")

FilePyt.WriteLine("def DCommand():")
FilePyt.WriteLine(" com = """)
FilePyt.WriteLine(" com += ""var p=new Object();\n""")
FilePyt.WriteLine(" com += ""var Height="" + A + ";\n""")
FilePyt.WriteLine(" com += ""var Weight="" + B + ";\n""")
FilePyt.WriteLine(" com += ""p.Plane =")
agb.GetActivePlane();\n""")
FilePyt.WriteLine(" com += ""p.Origin =")
p.Plane.GetOrigin();\n""")
FilePyt.WriteLine(" com += ""p.XAxis =")
p.Plane.GetXAxis();\n""")
FilePyt.WriteLine(" com += ""p.YAxis =")
p.Plane.GetYAxis();\n""")
FilePyt.WriteLine(" com += ""p.Sk2 =")
p.Plane.NewSketch();\n""")
FilePyt.WriteLine(" com += ""p.Sk2.Name =")
\"Sketch2\";\n""")
FilePyt.WriteLine(" com += ""with (p.Sk2)\n""")
FilePyt.WriteLine(" com += ""{\n""")
FilePyt.WriteLine(" com += ""p.Ln1 = Line(-" + B + ", -" + A")
+ ", " + B + ", -" + A + ");\n""")
FilePyt.WriteLine(" com += ""p.Ln2 = Line(" + B + ", -" + A")
+ ", " + B + ", " + A + ");\n""")
FilePyt.WriteLine(" com += ""p.Ln3 = Line(" + B + ", " + A +")
", -" + B + ", " + A + ");\n""")
FilePyt.WriteLine(" com += ""p.Ln4 = Line(-" + B + ", " + A")
+ ", -" + B + ", -" + A + ");\n""")
FilePyt.WriteLine(" com += ""}\n""")
FilePyt.WriteLine(" com += ""\n""")
FilePyt.WriteLine(" com += ""with (p.Plane)\n""")
FilePyt.WriteLine(" com += ""{\n""")
FilePyt.WriteLine(" com += ""agb.Extrude(agc.Add, p.Sk2,")
agc.DirNormal, agc.ExtentFixed, " + H + ", agc.ExtentFixed, 0.0, agc.No, 0.0,")
0.0);\n""")
FilePyt.WriteLine(" com += ""}\n""")
FilePyt.WriteLine(" com += ""\n""")
FilePyt.WriteLine(" com += ""p.Plane.EvalDimCons();\n""")
FilePyt.WriteLine(" com += ""\n""")
FilePyt.WriteLine(" com += ""agb.Regen();\n""")
FilePyt.WriteLine(" com += ""\n""")
FilePyt.WriteLine(" com += ""var body =")
ag.agApplet.FeatureMgr.Body(0);\n""")

```

```

FilePyt.WriteLine(" com += ""body.Name =
\""Parallelepiped\"";\n""")
FilePyt.WriteLine(" return com")

FilePyt.WriteLine("template3 = GetTemplate(TemplateName =
""Geometry""")")
FilePyt.WriteLine("system3 = template3.CreateSystem()")
FilePyt.WriteLine("template4 = GetTemplate(TemplateName =
""Mesh""")")
FilePyt.WriteLine("system4 = template4.CreateSystem()")
FilePyt.WriteLine("geometry2 =
system3.GetContainer(ComponentName = ""Geometry""")")
FilePyt.WriteLine("geometry2.Edit()")

FilePyt.WriteLine("geometry2.SendCommand(Command=DCommand()")")
FilePyt.WriteLine("geometry2.Exit()")
FilePyt.WriteLine("component4 = system3.GetComponent(Name =
""Geometry""")")
FilePyt.WriteLine("component5 = system4.GetComponent(Name =
""Geometry""")")
FilePyt.WriteLine("component5.ReplaceWithShare(TargetSystem
= system4, ComponentToShare = component4, SourceSystem = system3)")
FilePyt.WriteLine("")
FilePyt.WriteLine("Save(Overwrite = True)")
FilePyt.Close()

Shell(WBCommand2)

```

Подобную процедуру проходит и импорт модели, выбранной пользователем, однако в этом случае без использования JavaScript.

После проверки загрузки исследуемого тела и окружающего пространства в программный комплекс ANSYS. Пользователю предлагается выбрать материалы, которые будут применяться к его модели. Имеется возможность создать материал со индивидуальными свойствами и внести его в библиотеку материалов проекта, что позволит в дальнейшем пользоваться им в любом вычислительном блоке проекта (Рисунок 4.7).

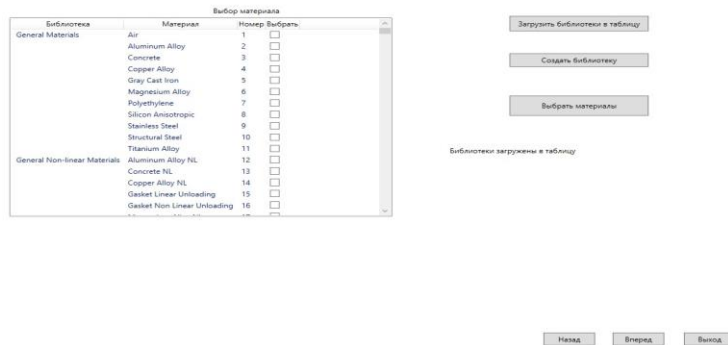


Рисунок 4.7. Окно выбора или создания материала проекта.

После подтверждения пользователем о выборе или создании материала идет проверка правильности введенных данных и на повтор существующих библиотек. Если

процесс проверки закончился без ошибок запрос делается «Ядру», чтобы запустить этап формирования алгоритма из модуля «Макрос» для внесения новых или выбранных материалов в проект программного комплекса ANSYS. После формирования алгоритма наступает очередь работы модуля «Convert», который преобразует алгоритм в структуру пакетного файла для передачи в программный комплекс. После этого идет отправка через «Ядро» в программный комплекс ANSYS, где создается необходимый материал или добавляется существующий из библиотеки материалов.

Следующий этап представляет собой наложение сетки конечных элементов на загруженные модели. На этом этапе происходит внедрение нового конечного элемента с ОФФ в библиотеку КЭ. Конечный элемент накладывается на исследуемую модель с помощью языка программирования APDL в виде команд, однако как было сказано в главе 2, эти команды должны уже быть расположены в вычислительном модуле. Доступа к коду существующих вычислительных модулей нет, однако есть возможность создать свои используя вычислительные мощности существующих. Поэтому для наложения сетки на исследуемую модель используется компонент вычислительного модуля «Transient Structure» - «Mesh». Кроме этого блок «Пользователь» делает запрос к «Ядру» для формирования внутри проекта необходимых вычислительных модулей, после этого «Ядро» запускает процедуры формирования пакетного файла на основе функций и кода, расположенного в модуле программной среды «Модули». В этом модуле содержатся коды на языке программирования XML – для формирования самих вычислительных модулей в программном комплексе ANSYS,

```

<interface context="Project">
  <images>D:\ANS\ANSYS
Inc\v150\Addins\ACT\extensions\Dis\images</images>
</interface>
<interface context="Mechanical">
  <images>D:\ANS\ANSYS
Inc\v150\Addins\ACT\extensions\Dis\images</images>
  <toolbar name="ToolBar1" caption="Resonance">
    <entry name="Resonance" icon="s1" caption="Resonance">
      <callbacks>
        <onclick>OnClickTB1Resonance1</onclick>
      </callbacks>
    </entry>
    <entry name="Aerodynamics" icon="s1" caption="Aerodynamics">
      <callbacks>
        <onclick>OnClickTB1Aerodynamics</onclick>
      </callbacks>
    </entry>
  </toolbar>
  <toolbar name="ToolBar2" caption="ModalTable">
    <entry name="SaveModalTable" icon="s1" caption="SaveModalTable">
      <callbacks>
        <onclick>OnClickTB1SaveModalTable1</onclick>
      </callbacks>
    </entry>
  </toolbar>
</interface>

```

и создание в каждом из модулей необходимые компоненты для решения выбранной технической задачи.

```

    <workflow name="wf1" context="Project" version="1">
      <taskgroups>
        <taskgroup name="ModalTable" caption="ModalTable" icon="s1"
category="Analysis Systems" abbreviation="Thesis" version="1">
          <includeTask external="True"
name="EngDataCellTemplate" caption="Solid materials"/>
          <includeTask external="True"
name="GeometryCellTemplate" caption="Body and surroundings"/>
          <includeTask external="True"
name="SimulationModelCellTemplate" caption="CE grid"/>
          <includeTask external="True"
name="SimulationSetupCellTemplate_StructuralANSYS" caption="Formulation of the
problem"/>
          <includeTask external="True"
name="SimulationSolutionCellTemplate_StructuralANSYS" caption="Decision"/>
          <includeTask external="True"
name="SimulationResultsCellTemplate_StructuralANSYS" caption="Results"/>
          <attributes SolverType="ANSYS" SystemName="Modal
(ANSYS)"/>
        </taskgroup>
        <taskgroup name="Resonance" caption="Resonance" icon="s1"
category="Analysis Systems" abbreviation="Thesis" version="1">
          <includeTask external="True"
name="EngDataCellTemplate" caption="Solid materials"/>
          <includeTask external="True"
name="GeometryCellTemplate" caption="Body and surroundings"/>
          <includeTask external="True"
name="SimulationModelCellTemplate" caption="CE grid"/>
          <includeTask external="True"
name="SimulationSetupCellTemplate_StructuralStaticANSYS" caption="Formulation of
the problem"/>
          <includeTask external="True"
name="SimulationSolutionCellTemplate_StructuralStaticANSYS" caption="Decision"/>
          <includeTask external="True"
name="SimulationResultsCellTemplate_StructuralANSYS" caption="Results"/>
          <attributes SolverType="ANSYS" SystemName="Transient
Structural (ANSYS)"/>
        </taskgroup>
        <taskgroup name="Aerodynamics" caption="Aerodynamics"
icon="s1" category="Analysis Systems" abbreviation="Thesis" version="1">
          <includeTask external="True"
name="EngDataCellTemplate" caption="Solid materials"/>
          <includeTask external="True"
name="GeometryCellTemplate" caption="Body and surroundings"/>
          <includeTask external="True"
name="SimulationModelCellTemplate" caption="CE grid"/>
          <includeTask external="True"
name="SimulationSetupCellTemplate_CFXANSYS" caption="Formulation of the problem"/>
          <includeTask external="True"
name="SimulationSolutionCellTemplate_CFXANSYS" caption="Decision"/>

```

```

        <includeTask                                external="True"
name="SimulationResultsCellTemplate_CFXANSYS" caption="Results"/>
        <attributes                                SolverType="ANSYS"    SystemName="CFX
(ANSYS)"/>
        </taskgroup>
    </taskgroups>
</workflow>

```

После формирования пакетного файла с необходимыми кодами он передается в программный комплекс ANSYS, в котором уже формируются модули (Рисунок 4.8)

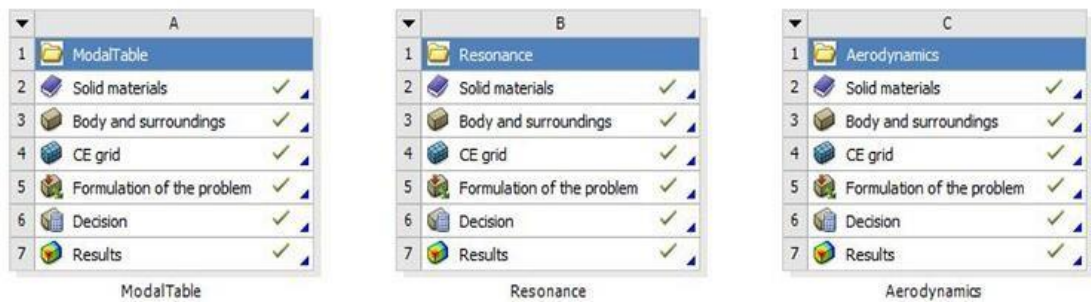


Рисунок 4.8. Вычислительные модули, создаваемые в программном комплексе ANSYS

Помимо XML используется язык Python – для формирования отдельных функций и процедур внутри этих модулей для работы новой методики. Ниже представлен фрагмент кода пакетного файла для формирования сгущения сетки в области исследуемого тела для получения высоких показателей точности результатов. Кроме этого, согласно методике расчета сетка исследуемого тела будет подвергаться изменению в соответствии с его деформацией, что приводит к изменению сетки воздушного пространства для сохранения качества в области тела. Параметры сетки, которые представлены ниже используются для наложения на окружающее пространство, и они будут подвергаться изменению на основе показателей, которые будут поступать из программного комплекса ANSYS. Стоит отметить, что в этом случае в действие вступает модуль «Transmitter», который запускается, как только «Ядро» получает информацию от ANSYS о том, что завершился первый шаг по времени в Aerodynamics и получены результаты давления на поверхности исследуемого тела, которое приводит к его перемещению. Модуль «Transmitter» считывает данные по перемещению из файла-результатов проекта, так как среда знает где расположены все файлы проекта, и формирует файл, в котором делает привязку значений перемещений узлов сетки исследуемого тела в каждый момент времени расчета.

```

FilePyt.WriteLine("# encoding: utf-8")
FilePyt.WriteLine("SetScriptVersion(Version=""15.0"")")
FilePyt.WriteLine("")
FilePyt.WriteLine("Open(FilePath="" + NamePro + ".wbpj"")")

```

```

        FilePyt.WriteLine("")

        FilePyt.WriteLine("def MCommand():")
        FilePyt.WriteLine(" com = """)
        FilePyt.WriteLine("     com     +=     ""var     Mesh_Mod     =
DS.Tree.FirstActiveBranch.MeshControlGroup;\n""")
        FilePyt.WriteLine("         com         +=
""DS.Script.SelectItems(""+Mesh_Mod.ID);\n""")
        FilePyt.WriteLine(" com += ""DS.Script.doInsertMeshSize(1)\n""")
        FilePyt.WriteLine(" com += ""ListView.ActivateItem(""Scoping
Method"");\n""")
        FilePyt.WriteLine(" com += ""ListView.ItemValue = ""Named
Selection"" ;\n""")
        FilePyt.WriteLine(" com += ""ListView.ActivateItem(""Named
Selection"");\n""")
        FilePyt.WriteLine(" com += ""ListView.ItemValue = ""Cilinder""
;\n""")
        FilePyt.WriteLine("         com         +=
""ListView.ActivateItem(""Type"");\n""")
        FilePyt.WriteLine(" com += ""ListView.ItemValue = ""Sphere of
Influence"" ;\n""")
        FilePyt.WriteLine(" com += ""ListView.ActivateItem(""Sphere
Center"");\n""")
        FilePyt.WriteLine(" com += ""ListView.ItemValue = ""Global
Coordinate System"";\n""")
        FilePyt.WriteLine(" com += ""ListView.ActivateItem(""Sphere
Radius"");\n""")
        FilePyt.WriteLine(" com += ""ListView.ItemValue = ""0,25""\n""")
        FilePyt.WriteLine(" com += ""ListView.SelectedItem.IsChecked =
""true""\n""")
        FilePyt.WriteLine(" com += ""ListView.ActivateItem(""Element
Size"");\n""")
        FilePyt.WriteLine(" com += ""ListView.ItemValue = ""0,01""\n""")
        FilePyt.WriteLine(" com += ""ListView.SelectedItem.IsChecked =
""true""\n""")
        FilePyt.WriteLine(" return com")

        FilePyt.WriteLine("template4 = GetTemplate(TemplateName =
""Mesh"")")

        FilePyt.WriteLine("system4 = template4.CreateSystem()")
        FilePyt.WriteLine("geometry2 =
system4.GetContainer(ComponentName = ""Mesh"")")
        FilePyt.WriteLine("geometry2.Edit()")
        FilePyt.WriteLine("geometry2.SendCommand(Command=MCommand())")
        FilePyt.WriteLine("geometry2.Exit()")
        FilePyt.WriteLine("")
        FilePyt.WriteLine("Save(Overwrite = True)")
        FilePyt.Close()

```

По завершению этапа наложения сетки наступает этап указания граничных условий и постановки самой задачи. От пользователя требуется указать что указывается на входе, выходе или на открытых областях и указать численное значение выбранного типа. Помимо этого, необходимо указать угловую скорость вращения или её отсутствие.

Так как вычислительный модуль формируется с помощью готового кода, то в отличие от стандартного CFX, в новом модуле уже содержатся необходимые для работы методики формулы и переменные, а также выбраны требуемые ссылки на файлы и параметры расчета, то есть пользователю не требуется заходить в программный комплекс, чтобы настроить всё вручную, но данная возможность имеется.

При желании пользователь может сохранить проект на этапе выполнения в программной среде и вручную открыть программный комплекс, зайти в проект, затем сохранить его отредактировав необходимые параметры, и затем обновлением данных, программная среда сделает перерасчет параметров проекта и перезапишет уже новые, введенные пользователем непосредственно в сам проект.

Это становится возможным благодаря модулю «Память», она начинает действовать, когда «Ядру» приходит запрос на остановку программной среды. В этом случае «Память» сохраняет все параметры, введенные пользователем в среде, в матрицу данных – идентификационный номер (ID), тип решаемой задачи, набор интерфейсов и что было введено в каждый элемент на интерфейсе. После ответа от «Памяти», «Ядро» блокирует кнопки и поля программной среды и открывает проект пользователя, запускается процесс ожидания его закрытия с сохранением.

Когда проект закрывается «Ядро» получает об этом оповещение и запускает процесс в модуле «Transmitter» для выгрузки новых параметров из файлов проекта. Когда происходит выгрузка параметров модуль «Память» сравнивает их с сохраненными и заменяет их в программной среде.

Далее после указания всех параметров в вычислительном модуле Aerodynamics модуль «Макрос» собирает пакетный файл на языке программирования CCL, который является скриптовым языком вычислительного модуля CFX. Так как Aerodynamics основывается на вычислительных мощностях CFX, то данный язык подходит и для нового модуля.

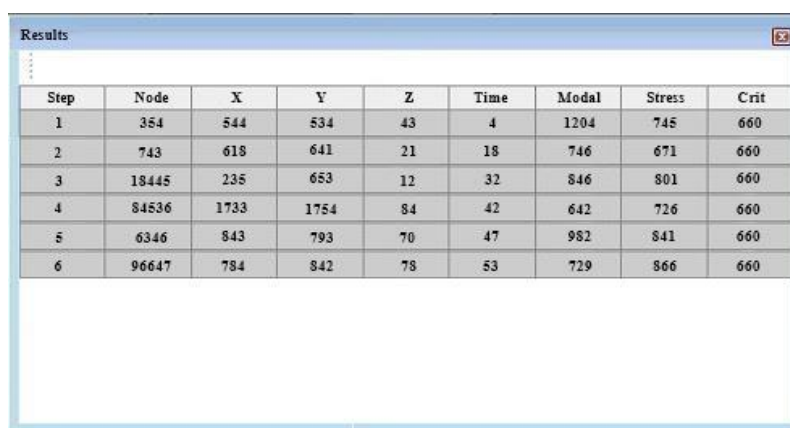
Как было сказано, на этом этапе происходит перерасчет двух сеток согласно движению исследуемого тела. Каждое положение узла сетки фиксируется в файле программной среды для дальнейшего учета значений деформаций и напряжений при комбинировании с частота собственных колебаний.

В модуле «Resonance» происходит подгрузка значений частот собственных колебаний и их форм. При этом данная подгрузка проводится не как повтор расчета частот, а определяется частота и затем - значения деформаций и напряжений соответствующие этой частоте.

Вторым этапом является загрузка средствами языка программирования Python файла с данными о перемещениях узлов исследуемого тела. Иначе говоря, в вычислительном модуле с уже известной сеткой и номерами узлов, на каждом узле создается таблица перемещений по времени. Такие первоначальные граничные условия создаются, но они накладываются один раз на каждый момент времени. То есть перед тем, как пробежать по всем частотам собственных колебаний рассматривается одно положение узлов исследуемого тела со значениями напряжения в этот момент.

Следующий этап - определение максимально допустимых значений напряжений, возникающих в исследуемом теле. Так как происходит наложение всех форм частот на исследуемое тело в момент времени, то происходит усиление или уменьшение значений напряжения в каждом узле. То есть наше тело уже имеет деформированную форму в первый момент снятия показаний и, следовательно, обладает остаточными напряжениями от этой деформации. Программная среда с помощью языка программирования Python задает координаты всех узлов сетки исследуемого тела и накладывает на него значения напряжения из результатов расчета вычислительного модуля с помощью решателя CFX.

В результате получают фильтрованные значения напряжений, которые отвечают критерию прочности. Они заносятся в таблицу результатов с привязкой к моменту времени, частоте свободных колебаний и к координатам на исследуемой модели, где возник скачок напряжения в теле (Рисунок 4.9).



Step	Node	X	Y	Z	Time	Modal	Stress	Crit
1	354	544	534	43	4	1204	745	660
2	743	618	641	21	18	746	671	660
3	18445	235	653	12	32	846	801	660
4	84536	1733	1754	84	42	642	726	660
5	6346	843	793	70	47	982	841	660
6	96647	784	842	78	53	729	866	660

Рисунок 4.9. Таблица результатов колебаний лопастей винта, превышающих установленный критерий

4.5 Алгоритм и блок-схема разработанной программной среды

Рассмотрим алгоритм взаимодействия каждого блока или модуля с другими и какими данными они пользуется, на основе этого сформируется блок-схема программной среды.

Первым для рассмотрения является ядро. Это процедура, которая начинает работать в момент запуска программной среды. Она запускает внутри себя цикл, выходом из которого является событие закрытия программной среды.

В этом цикле на первом шаге запускается проверка наличия всех необходимых для работы программной среды файлов. Если какие-либо файлы отсутствуют, то программная среда выдает предупреждение об ограничениях ее работы.

Следующей функцией после проверки является задействование модуля «Пользователь» со стандартными интерфейсами ввода параметров расположения исполнительного файла программного комплекса, названия проекта и расположения рабочей папки, а также выбор типа решаемой технической задачи. При отсутствии модуля

«Пользователь» программная среда прекращает работу с выводом информации о причинах закрытия.

Пока работает модуль «Пользователь» «Ядро» запускает второй поток обработки, направленный на проверку блока «Память» и какая, последняя запись в ней храниться. В свою очередь блок «Память» обращается к базе данных, которая хранит всю информацию о решенных проекта с введенными параметрами и полученными результатами.

Когда пользователь завершает ввод данных на первом интерфейсном окне модуль «Пользователь» возвращает значение завершенности в «Ядро». При этом, если второй поток, открытый для проверки памяти, не завершился, «Ядро» запускает третий поток, который направлен на запуск модуля «Макрос» и «Convert», которые в свою очередь внутри своих циклов вызывает другие модули и блоки для работы.

Любое действие в программной среде или ответ от программного комплекса поступает в виде запроса к «Ядру», которое уже на основе цикла «switch» фильтрует запросы и запускает необходимые процедуры.

Теперь рассмотрим работу модуля «Пользователь» она заключается в выводе в родительское окно программной среды, которое создается «Ядром», интерфейсных окон. Интерфейсные окна формируются на основе выбранной технической задачи. Информация о количестве элементах и их типах на каждом окне и в каждой решаемой задаче уже содержаться в блоке «Память». Однако, только модуль «ПреПостпроцессор» может сформировать последовательность вывода этих окон, а затем привязать к каждому элементу переменную для обработки и передать готовый порядок вывода модулю «Пользователь», который в свою очередь имея порядок вывода обращается к памяти и выгружает их. Всё что вводит или выбирает пользователь в программной среде храниться в модуле «Пользователь» и только после закрытия программной среды возвращает ответ в «Ядро» и происходит передача данных от модуля «Пользователь» в блок «Память».

Блок «Память» - отдельно создаваемая процедура с собственной структурой для хранения действий, параметров, строк кода, их обработки и вывода. Однако, блок «Память» является функциональной процедурой, которая выполняет действия над данными, сами же данные хранятся в базе данных, согласно структуре заложенной в блоке. Блок «Память» контактирует со всеми модуля и блоками программной среды, но только с «разрешения» «Ядра», если разрешение получено, то происходит прямая передача данных от этого блока в другие процедуры и функции (Рисунок 4.10).

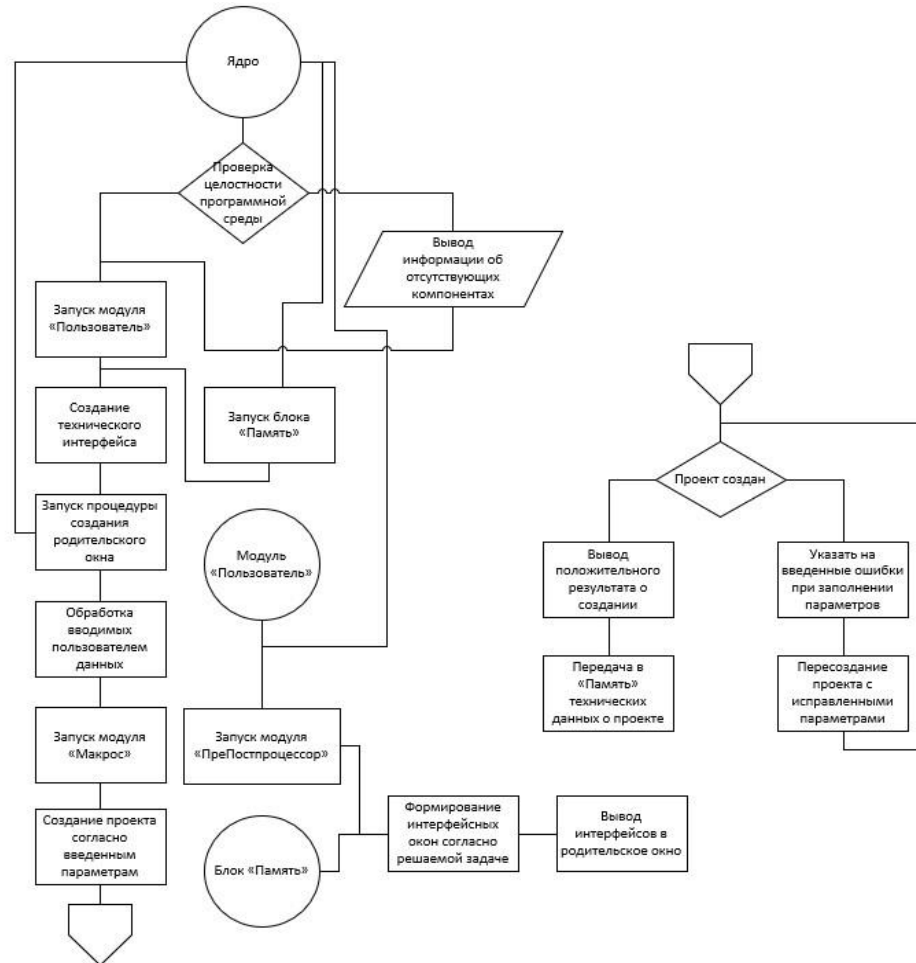


Рисунок 4.10. Алгоритм первого этап работы программной среды

Следующий этап заключается в работе с интерфейсами в родительском окне. Все интерфейсы выводят одновременно в родительское окно и имеется возможно переключаться между ними. При завершении всех требуемых этапов пользователю предлагается запустить расчет и проанализировать результаты.

В интерфейсе по созданию или загрузке геометрии задействована работа модуля «Пользователь» и «ПреПостпроцессор», при подтверждении о загрузке геометрии «ПреПостпроцессор» запрашивает у «Ядра» разрешение на формирование вычислительных модулей в проекте и осуществление импорта геометрии. Перед импортом «Ядро» запускает модуль «Модуль», и он получает запрос о формировании в проекте необходимых вычислительных модулей согласно выбранной технической задаче и возвращает ответ об успешном создании и загрузке геометрии в проект.

Затем открывает библиотека материалов, в которой выбирается уже готовый материал или создается новый на основании вводимых параметров. С помощью уже запущенной процедуры «Converter» проверяет правильность строения материала и передает их на формирование пакетного файла. В ответ «Transmitter» считывает результат об успешном добавлении материала.

Сетка конечных элементов накладывается только в том случае, если выполнено создание вычислительных модулей в проекте и загружена необходимая геометрия. До наложения есть возможность задать параметры наложения и сгущения сетки в области

исследуемого тела. Кроме этого, «Ядро» запускает модуль «LA», который содержит в себе функции внедрения нового конечного элемента с ОФФ на исследуемое тело. С помощью модуля «Макрос» происходит наложение сетки и возвращается результат успешного наложения.

Задание граничных условий и постановка задачи по расчету динамического взаимодействия исследуемого тела с окружающим пространством начинается с указания типа границы воздушного пространства и численного его значения. После этого указывается что делает исследуемое тело в воздушном пространстве, затем происходит настройка технических параметров, в том случае если пользователь не знает, что требуется вводить, то программная среда подбирает стандартные параметры под тип решаемой задачи. Параллельно внедряется код, описывающий изменение идеального газа по всей области пространства на вязкий газ согласно уравнению Навье-Стокса (23). После указания всех параметров формируется CCL код, который отправляется в решатель для расчета с указанием его параметров. Последним шагом на данном этапе является запуск расчета динамического взаимодействия исследуемого тела с газами. Интерфейс анализа результатов становится активным только в том случае, когда будет получен результат завершения расчета (Рисунок 4.11).

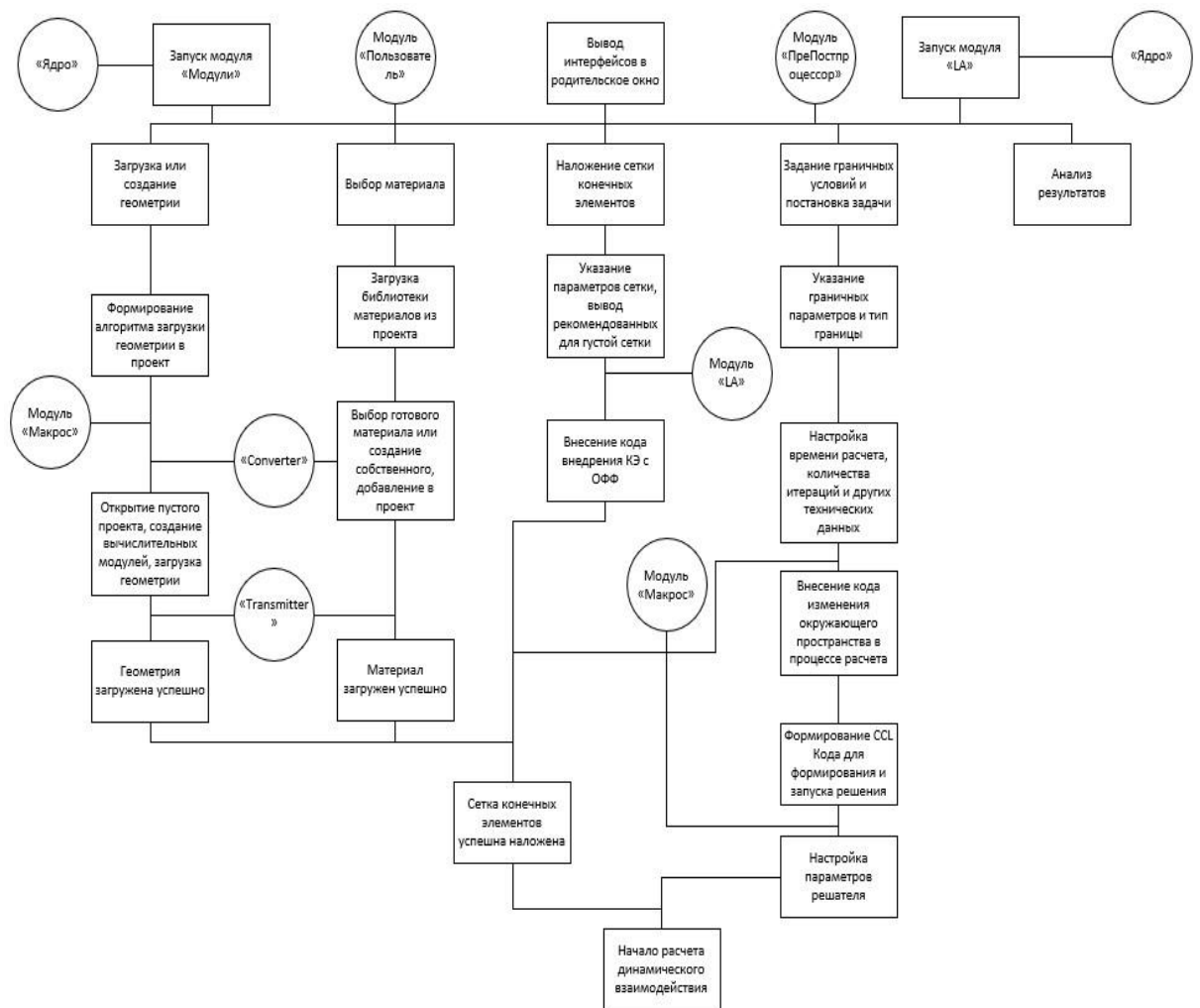


Рисунок 4.11. Второй этап работы программной среды с описанием алгоритма взаимодействия

На последнем этапе производится расчет динамического взаимодействия исследуемого тела с окружающей средой. При запуске решателя требуется дождаться окончания расчета на первом промежутке времени, чтобы определить как влияет давление, создаваемое воздушным потоком на перемещение исследуемого тела. На выходе будут получены координаты узлов сетки в результате перемещения, а также значения напряжения, возникающие в данный момент. Когда получен ответ о выводе результатов первого промежутка «Transmitter» считывает координаты узлов из файла расчета решателя CFX и заменяет их в исходном файле сетки исследуемого тела, затем новая сетка перегружается в вычислительном модуле Aerodynamics для изменения густоты сетки в области нового расположения тела и цикл перерасчета динамического взаимодействия повторяется до конца по времени, которое было задано в постановке задачи.

После завершения расчета запускается вычислительный модуль Resonance, в котором размещается исследуемое тело, таблица перемещений узлов и таблица частот собственных колебаний. У пользователя запрашивается критерий по напряжению, чтобы отфильтровать не нужные данные. Результатом работы данного модуля будет являться таблица, в которой размещается информация о частоте, времени, координате и значение напряжения (Рисунок 4.12).

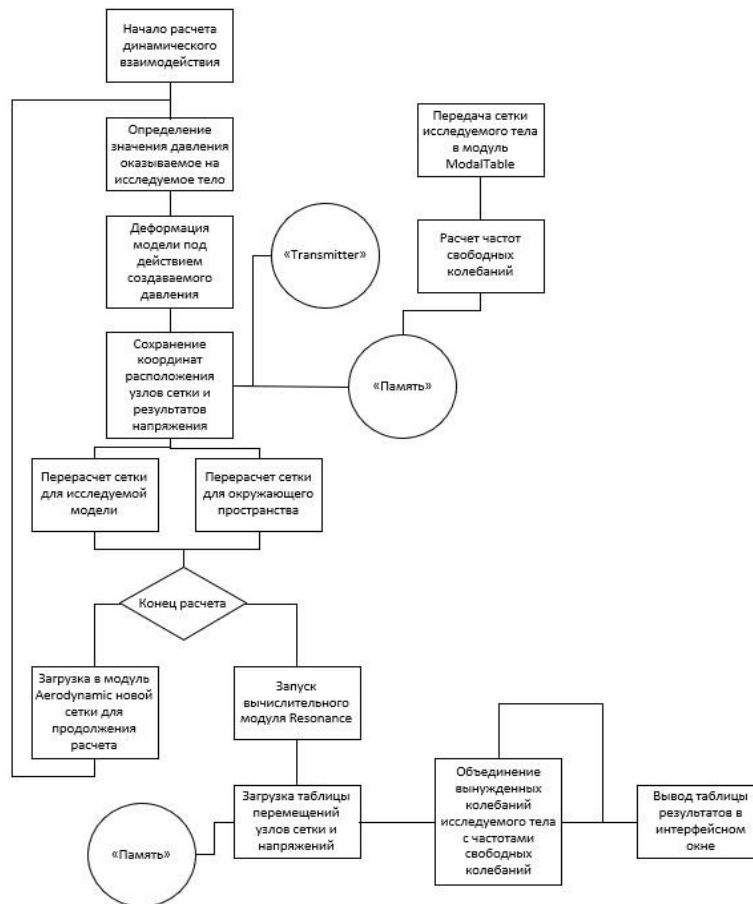


Рисунок 4.12. Третий этап работы программной среды с описанием алгоритма

4.6 Выводы. Инструкция по использованию программной среды

Разработанная программная среда используется для реализации работы нового метода расчета динамического взаимодействия твердого тела с газами, упрощает взаимодействие с программным комплексом ANSYS и автоматизирует большинство процедур подготовки проекта к решению задачи. Благодаря применению программной среды нет необходимости постоянно пересохранять проект при опасении его закрытия, удобство и доступность параметров настройки проекта, так как они все расположены в родительском окне, а не в каждом компоненте каждого вычислительного модуля.

Использование программной среды заключается в необходимости выделить доступное для записи и удаления технически необходимых файлов, называемые пакетными файлами, которые удаляются после выполнения поставленной перед ними задачи.

Необходимо учитывать форматы загружаемых цифровых моделей в проект. Описание доступных форматов было дано в главе 2.

Подготовка нового материала возможна только в созданной новом массиве материалов, это сделано для предотвращения нарушения целостности библиотеки материалов, расположенной внутри проекта в случае возникновения конфликта имен переменных. Свойства материалов рассчитываются согласно указанным в руководстве библиотеки функциям, при указании модуля Юнга и коэффициента Пуассона, остальные параметры автоматически заполняются, или наоборот.

Наложение сетки конечных элементов основывается на использование стандартных средств разбиения модели на конечный области, к которым применяется настройки конечного элемента. По этой причине программная среда не влияет на скорость наложения сетки на цифровую модель, поэтому при задании крайне маленького параметра размера КЭ, то программная среда на основе сравнения размера тела и сетки выдаст предупреждение о достаточно долгом процессе наложения, но и о процессе перестройки в дальнейшем.

ГЛАВА 5. ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ СОЗДАННОЙ ПРОГРАММНОЙ СРЕДЫ. РЕШЕНИЕ АКТУАЛЬНЫХ ТЕХНИЧЕСКИХ ЗАДАЧ

5.1 Обзор типов технических задач динамики взаимодействия винта с окружающей газовой средой

Применение разработанной программной среды позволяет решать ряд задач, а именно расчет начала движения тела под воздействием набегающего воздушного потока, моделирование установившегося вращения упругого винта, моделирование потока газа в аэродинамической трубе, моделирование передачи энергии воздушного потока с одного тела на другое, определение резонансных колебаний в пределах критерия по напряжению.

Рассмотрим пример решения задачи, в котором винт имеет фиксированную угловую скорость вращения и набегающий поток направлен под прямым углом к оси вращения. При запуске программной среды требуется указать размещение исполняемого файла программного комплекса ANSYS – RunW2.exe. После указания ссылки на этот файл необходимо указать рабочую папку проекта и его название, при завершении этого процесса запуститься в фоновом режиме проверка на правильность расположения исполняемого файла и на доступ к рабочей папке для записи и чтения файлов. При выборе типа решаемой задачи в рабочей папке проекта создаются файлы программного комплекса ANSYS с размещенными внутри него новыми вычислительными комплексами.

Разобьем весь процесс расчета от загрузки геометрии до анализа результатов:

1. Загрузка, создание цифровых моделей в проекте, выбор материала и наложение сетки конечных расчетов.
2. Расчет частот собственных колебаний и постановка задачи для расчета динамического взаимодействия.
3. Расчет резонансных частот и сравнительный анализ данных.

На первом этапе моделирования необходимо выбрать цифровую модель исследуемого тела, а затем либо загрузить готовую модель окружающего пространства, в этом случае требуется вручную проименовать плоскости у загруженной модели с обозначением входа, выхода, стенок и т.д, либо воспользоваться окном создания простой геометрии по заданным параметрам. После этого программная среда установит связь между именем поверхности и настройками по постановке задачи (Рисунок 5.1).

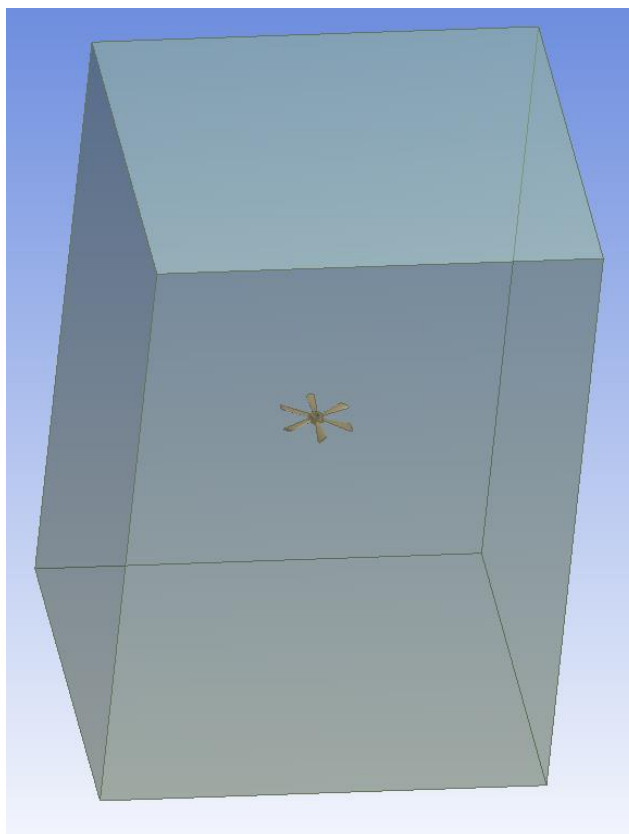


Рисунок 5.1. Загруженная цифровая модель винта и окружающего пространства

После подтверждения окончания процесса загрузки двух цифровых моделей программная среда переходит к наполнению библиотеки материалов в программном комплексе ANSYS.

Перед пользователем открывается окно, в которое загружаются все доступные материалы из библиотеки, и он может выбрать любой из них подвергнуть изменению или создать свой материал. Пользователь может в ручном режиме создавать композиционный материал через программный комплекс ANSYS и указав о его создании в программной среде, она подгрузит его в библиотеку проекта.

Так как либо автоматически, либо вручную именуется необходимые области, участвующие при решении задачи, то программная среда работает с именами поверхностей. При загрузке двух моделей можно заметить, что исследуемое тело расположено внутри окружающего пространства, следовательно, необходимо получение густой сетки вблизи этого тела, в то время как сетка на границе области может иметь более низкое качество.

Сетка накладывается в соответствующем для этого компоненте вычислительного модуля, в котором создаются дополнительные функции для настройки параметров сетки. Сами параметры подбираются пользователем с подробным описанием за что отвечает каждый из них. После определения параметров сетки наступает этап её генерации, в программной среде имеются условия проверки по параметрам, то есть если какой-то из параметров превышает установленный критерий, тогда программная среда выдает предупреждение о том, что процесс наложения сетки может быть длительным из-за такого-то параметра (Рисунок 5.2).

Вторым этапом решения задачи является расчет частот свободных колебаний в вычислительном модуле. В программной среде пользователь может указать диапазон поиска частот, а также их количество. Перед этим требуется либо в ручную указать место жетской задержки исследуемого тела, либо дать имя плоскости и программная среда сама установить необходимые граничные условия.

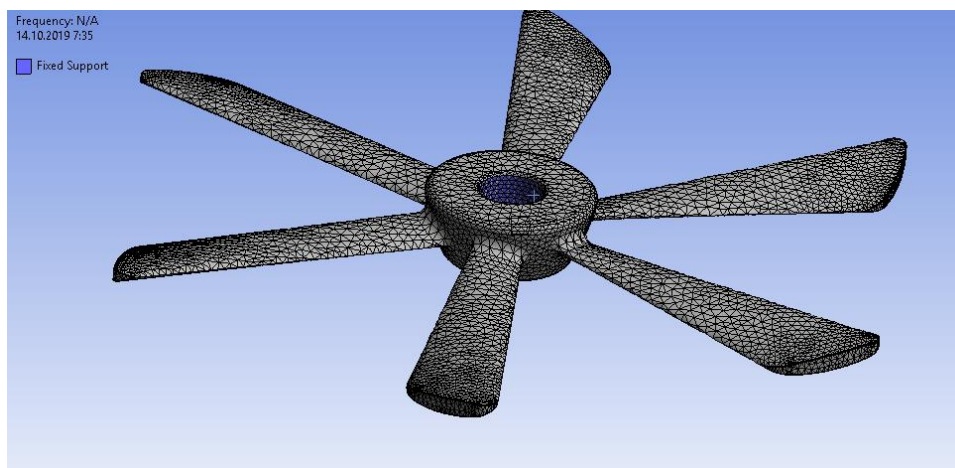


Рисунок 5.2. Сетка конечных элементов и жесткая заделка согласно проименованным поверхностям

Настройки и выбор решателя следует оставлять в программном выборе, так как новый вычислительный модуль использует не собственный решатель, а основывается на решателе модуля Modal для получения результатов расчета (Рисунок 5.3).

Details of "Analysis Settings"	
Options	
Max Modes to Find	13
Limit Search to Range	Yes
Range Minimum	0, Hz
Range Maximum	1,e+008 Hz
Solver Controls	
Damped	No
Solver Type	Program Controlled
Rotordynamics Controls	
Output Controls	
Analysis Data Management	

Рисунок 5.3. Настройка параметров решателя и поставка задачи

После получения таблицы частот свободных колебаний можно увидеть форму колебаний каждой частоты, а также значения деформаций и напряжений, которые при этом возникают. С помощью встроенной новой функции происходит выгрузка этой таблицы и сохранение всех частот с формами колебаний в отдельный файл с собственной

структурой для удобства пользования. За создание файла и выгрузку частот отвечает программная среда, пользователь не участвует в выполнении подобных операций. В результатах расчета можно помимо таблицы частот можно увидеть и результаты напряжений в теле (Рисунок 5.4).

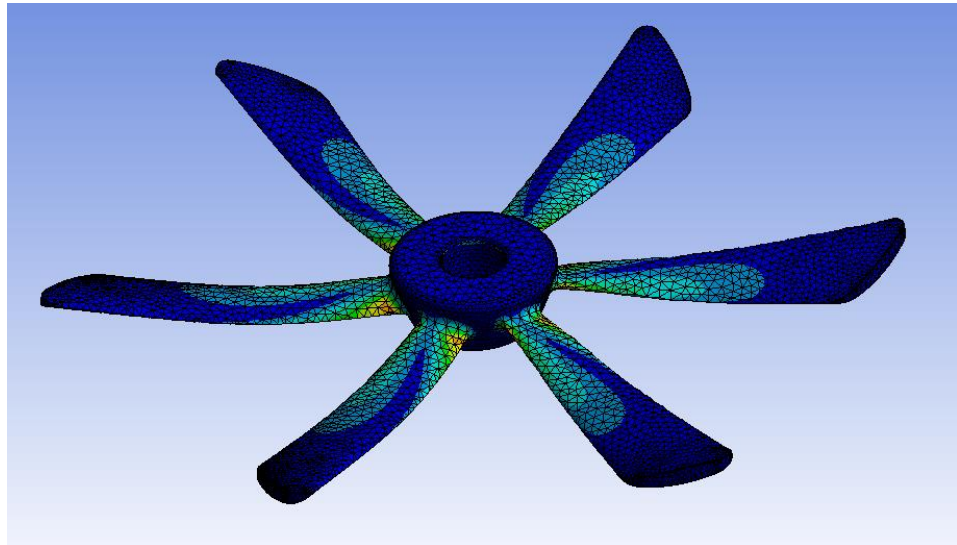


Рисунок 5.4. Результаты напряжения, возникающие на одной из частот свободных колебаний

При постановке задачи в вычислительном модуле Aerodynamics от пользователя в программной среде требуется выбрать тип граничного условия на входе и выходе, а также численное значение для этого условия. Для запуска решения необходимо указать значения на вход, выход и стенки, остальные типы выбираются пользователем вручную в проекте. Согласно методике динамического взаимодействия, программная среда автоматически вносит необходимые уравнения расчета в решатель вычислительного модуля и производит настройки этого решателя. При этом программная среда формирует алгоритм работы решателя на языке CCL для выгрузки и хранения результатов расчета на каждом промежутке времени. Пользователю выводится информация о том, сколько реального времени прошло с начала запуска решателя и сколько приблизительно осталось ожидать. В конце каждого промежутка времени программная среда обращается к компоненту программного комплекса ANSYS для перестройки первоначальной сетки и подгрузки её в решатель динамики (Рисунок 5.5).

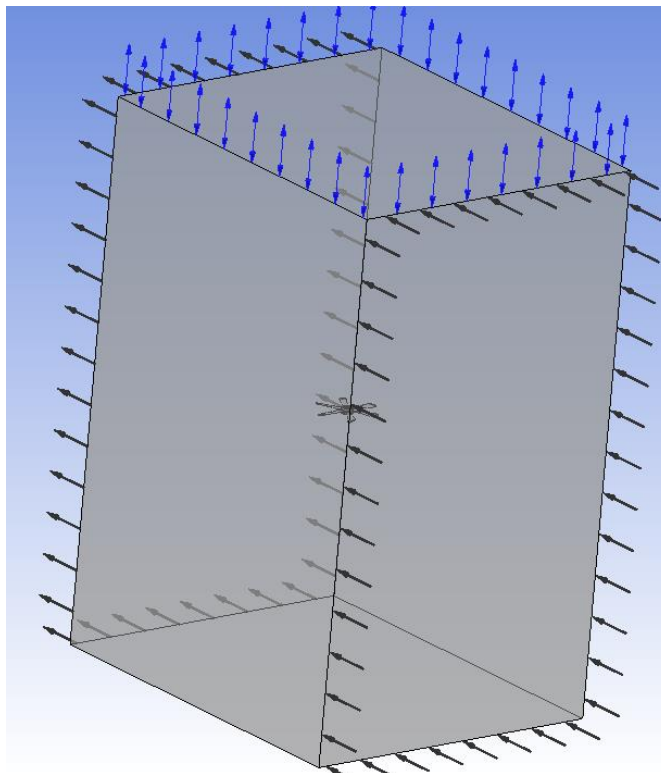


Рисунок 5.5. Постановка задачи и задание граничных условий

Последний этап расчета в программной среде является автоматизированным. После завершения этапа расчета динамического взаимодействия твердого тела с окружающей средой, происходит запуск вычислительного модуля Resonance, в котором подгружается сетка тела в конце первого промежутка времени с полученными значениями напряжения, а затем на неё накладываются значения напряжений от каждой частоты свободных колебаний, тем самым определяется как влияет каждая частота на исследуемую модель в определенный промежуток времени.

5.2 Техническая задача 1. Моделирование динамики лопастей винта, находящихся в воздушном потоке. Анализ достоверности

Апробация метода моделирования проводилась на основе результатов эксперимента «Вращение трехлопастного винта под действием набегающего потока воздуха» на базе МАИ в научно-образовательном центре «Конструирование и проектирование авиационной техники» Рипецким Андреем Владимирович доцентом кафедры 904.

При моделировании и проведении эксперимента было принято решение о изготовлении винта из спрессованных досок с последующей механической доработкой для придания окончательного профиля лопасти.

Цифровая модель лопасти для моделирования создавалась на основе изготовленной с помощью координатно-измерительной машины. В результате измерения были получены параметрические данные: длина лопасти – 1152.0511 мм;

ширина лопасти у основания 219.7883 мм, у законцовки 114.1934 мм; профиль внешней поверхности дорабатывался по точкам в программном продукте PC-DMIS CMM для КИМ (Рисунок 5.6).

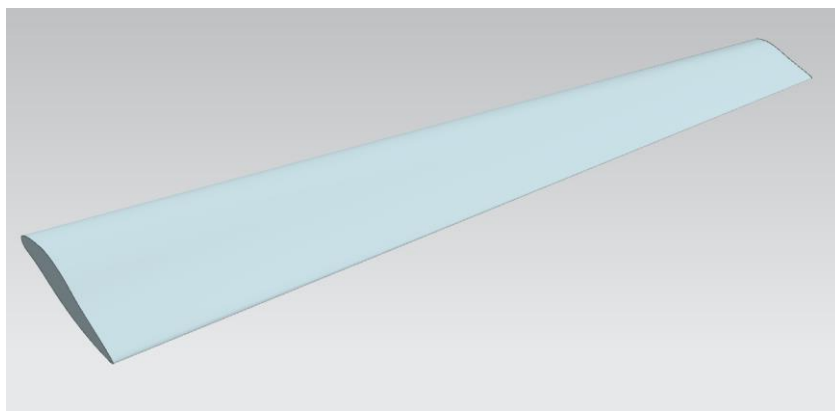


Рисунок 5.6. - Цифровая модель лопасти винта

Лопасты фиксируются в основании, которое вращается вокруг шарнирной оси. Основание представляет собой стальной короб в форме усеченного конуса с настройкой сглаженности по вершине (Рисунок 5.7).

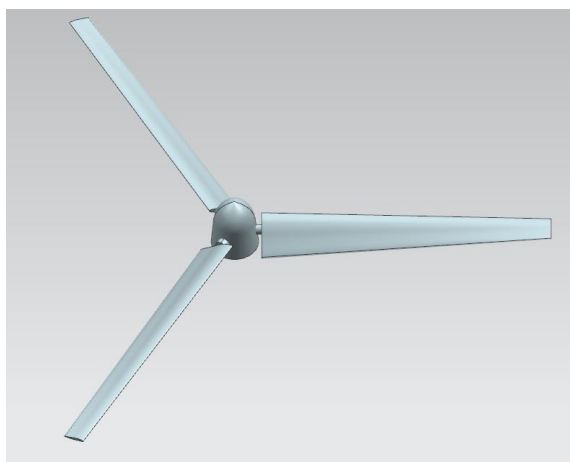


Рисунок 5.7 – Цифровая модель лопасти винта

Местом для установки оси вращения было выбрано цилиндрическое помещение с установленными вентиляционными решетками. В верхней части помещения располагается вентилятор ВР 80-75 с производительностью до 1000 м³/ч, полное давление до 180 Па. При проведении эксперимента с помощью анемометра была зафиксирована скорость потока для дальнейшего моделирования, которая равнялась 5 метров в секунду.

Расстояние между входной решеткой и расположенным винтом составляет 2 метра, высота расположения винта от поверхности равна 2 метрам, общий диаметр помещения равен 4.5 метра. В нижней части помещения располагаются выходные решетки для выхода воздушного потока из помещения. Данное помещение использовалось для покраски изделий или нанесения дополнительных покрытий на различные поверхности.

При создании цифровой модели помещения было принято решение о создании цилиндрической области и было сделано допущение, что в области отсутствует горизонтальный пол, эксперимент и моделирование проводились для расчета вращательного движения лопасти вокруг оси винта за счет набегающего потока. Размеры области и форма стенок не оказывают заметного влияния на результаты. Винт расположен под входной решеткой так, что, в моделировании не влияют потоки, отраженные от стен (Рисунок 5.8).

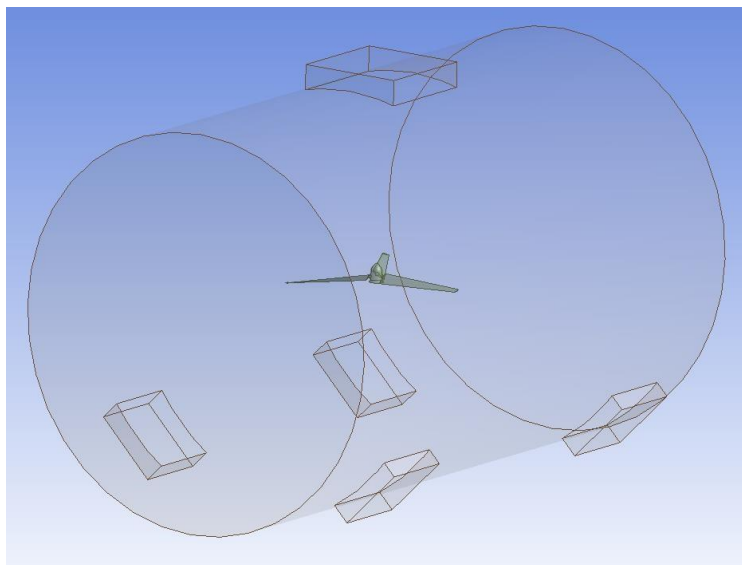


Рисунок 5.8 – Цифровая модель области и расположенного внутри исследуемого винта.

При моделировании моделей для помещения задавался тип Fluid, для винта – Solid. Новый метод основывается на использовании связанной задачи динамики взаимодействия набегающего потока и лопастей. Метод определяет перемещение каждой точки лопасти. В каждой точке, выбранной на поверхности лопастей, снимаются значения деформаций и напряжений, получаемые при взаимодействии с потоком, и передаются в авторский модуль перерасчета, который останавливает расчет на последней итерации и изменяет геометрию лопастей винта на основании получаемых данных. Такой способ впервые применяется в модуле аэрогидродинамики, благодаря созданной возможности экспорта деформированной модели в конце шага расчета. Для использования этого способа необходимо было внедрить в структуру расчетного модуля код дополнительных уравнений отслеживания линейной и угловой скорости перемещения точек в пространстве. В модели лопасти используются:

- тензорное уравнение движения (три скалярных уравнения)

$$\rho \left(\frac{d^2 \bar{u}}{dt^2} - \bar{F} \right) = \text{div} \bar{\sigma}$$

- шесть кинематических уравнений, связывающих компоненты ε_{ij} тензора деформации с компонентами u_i вектора перемещений (геометрически-нелинейная модель)

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i} \frac{\partial u_k}{\partial x_j} \right)$$

- закон Гука в тензорной форме (шесть скалярных уравнений) для изотропного материала (с гомогенизацией композиционной структуры лопасти)

$$\bar{\sigma} = \lambda I_1(\bar{\varepsilon}) \bar{g} + 2\mu \bar{\varepsilon}.$$

Уравнение аэроупругих колебаний вращающейся лопасти винта, полученное методом конечных элементов, записывается в матричном виде:

$$M\ddot{q} + D\dot{q} + (K + \Omega^2 S + B)q = 0$$

где M , K - матрицы инерции и жесткости всей лопасти; B и D - матрицы аэродинамической жесткости и аэродинамического демпфирования, зависящие от времени; S - матрица, учитывающая влияние центробежных сил.

Однако, данная матричная запись не учитывает воздействие динамического потока воздуха на твердое тело в данный момент времени с передачей результата этого воздействия на следующий шаг, в свою очередь это приводит к невозможности определения скорости и перемещения узлов сетки в каждый момент времени с учетом постоянного взаимодействия лопасти с окружающим пространством.

Для учета перемещений точек лопасти при взаимодействии с окружающим пространством требуется вычислять радиус-вектор, скорость и ускорение в каждой точке тела.

В расчетном модуле CFX программного комплекса ANSYS создается дискретная модель деформируемого тела, расположенную в воздушном пространстве и наблюдать линейную и угловую скорости каждого узла конечноэлементной сетки тела в течении промежутка времени.

$$v_k = \Lambda_k r_k \dot{r}_k; \omega_k = A_k \theta_k,$$

где Λ_k, A_k – ортогональные матрицы преобразования с учетом углов наклона точек на профиле лопасти, $r_k \dot{r}_k$ – радиус-вектор производный по времени для k -го узла, θ_k – производная по времени угла между начальным и конечным расположением k -узла.

Комбинируя результаты вычисления деформаций и напряжений с результатами вычисления перемещений узлов накладываемой сетки, получаем расположение лопастей для следующего момента времени. Кроме того, имеются результаты определения напряженно-деформированного состояния, которые позволяют определить места разрушения тела в момент резонансных колебаний (при учете частот свободных колебаний в каждый момент времени) или в результате иного воздействия.

Время и точность измерений и перестройки геометрии связаны с перестройкой сетки КЭ и влияет на структуру расчета. Так как за время решения одной задачи таких перестроений может быть более сотни и при этом следует проводить расчет с учетом частот свободных колебаний, то возможность сокращения времени и повышения точности является актуальной задачей.

Как было сказано выше, материал лопастей представляет собой спрессованные доски с отвердителем между ними и последующей доработкой, свойства материала для библиотеки ANSYS подбирались наиболее близкие к реальным свойствам досок.

Воздушное пространство представляет собой воздух с характеристиками, соответствующими температуре 25 градусов Цельсия при давлении в 1 атм. На деформируемое тело и на воздушное пространство накладывается сетка конечных элементов (Рисунки 5.9-5.10).

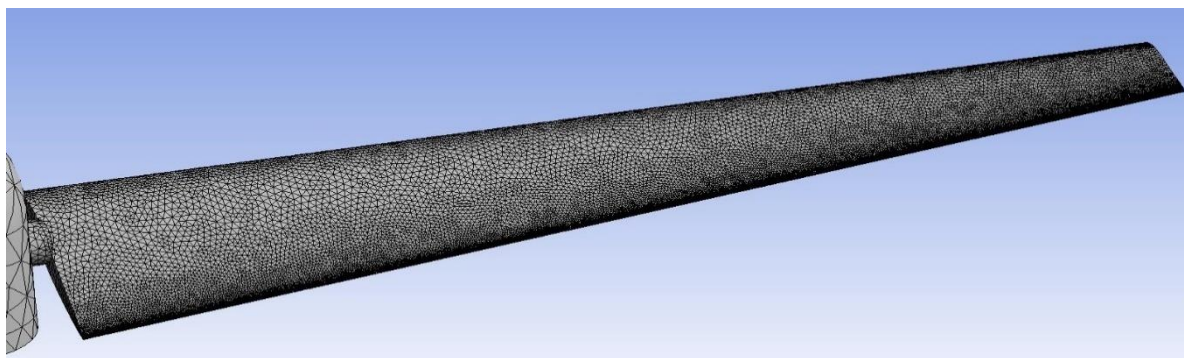


Рисунок 5.9 – Сетка конечных элементов на одной из лопастей винта

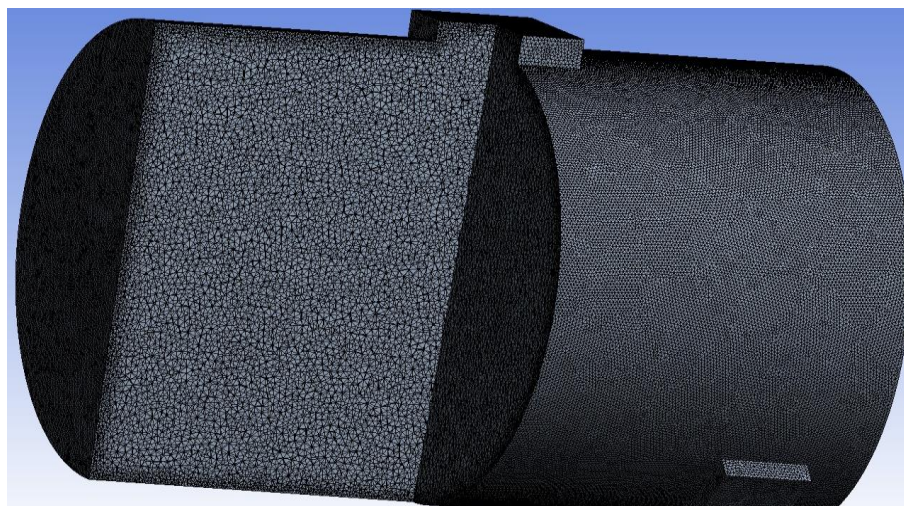


Рисунок 5.10 – Сетка конечных элементов для окружающего пространства

В результате наложения сетки КЭ было получено для воздушного пространства 5 млн. КЭ -тетраэдров, а для винта с лопастями – 1.5 млн. тетраэдров. Время наложения сетки составляет 3 часа с ручной настройкой параметров.

Обычно при расчете динамики исследуемый объект вычитается из окружающего пространства и расчет деформаций и напряжений производится в другом модуле, отличном от модуля расчета аэрогидродинамики. Метод моделирования с внедренным новым математическим обеспечением позволяет создавать новые дискретные модели, которые позволяют размещать тело внутри пространства и производить расчет деформаций и аэродинамики внутри одного модуля.

Для сравнения, в библиотеке модулей ANSYS для расчета влияния твердого тела на окружающее пространство с учетом заранее прикладываемой силы необходимо использовать три модуля: Transient Structure, Fluid Flow (CFX, Fluent) и System Coupling, без учета стандартных модулей. При передаче результатов из одного модуля в другой

происходит изменение структуры файла для возможности его чтения в модуле, где будет проходить расчет.

При использовании нового метода были созданы собственные вычислительные модули для расчета резонансных колебаний лопастей винта, при этом чтение полученных деформаций и напряжений, а также перемещений, осуществляется в одном авторском модуле без последующих преобразований файлов, содержащих результаты, для передачи в другие модули (Рисунок 5.11).



Рисунок 5.11 – Авторские модули нового метода моделирования (ModalTable – модуль расчета частот свободных колебаний и их сохранения в авторском файле для дальнейшего расчета, Resonance – модуль, совмещающий влияние свободных колебаний с деформациями и напряжениями тела в каждый момент времени, Aerodynamics – модуль расчета динамики взаимодействия лопастей винта с окружающей средой с сохранением информации о деформациях и напряжениях в каждый момент времени)

Формы для задания свойств граничных условий внутри модуля аэродинамики остались в основном без изменений. Отличие заключается в том, как эти параметры будут использоваться в процессе расчета, а также в настройках наложения решетки для снятия результатов.

При моделировании численного эксперимента были указаны граничные условия, в которых такие моменты как: шероховатость поверхностей-стенок, математической модели расчета турбулентных потоков и изменения температуры воздуха в процессе работы вентилятора, не учитывались.

Были указаны следующие входные параметры:

1. Скорость потока через входную решетку составляет 5 метров в секунду
 2. На границе выходных решеток задавалось среднее давление равное 0 Па. За пределами выходных решеток находится пространство с нулевым давлением и поток, создаваемый в помещении, может проходить через эти решетки с любым давлением.

3. Турбулентные потоки рассчитывались по модели k-epsilon. Уравнение движения преобразуется к виду, в котором добавлено влияние средней скорости (в виде турбулентной кинетической энергии) и процесса уменьшения её за счёт вязкости. Наиболее часто используемая модель при решении реальных инженерных задач.

4. Время вращения винта составляет 60 секунд, каждую 1 секунду снимаются показания с решетки и происходит перестроение геометрии.

5. Поверхности лопастей считаются шероховатыми.

Необходимо учитывать, что в новом методе тело не вырезается из окружающего пространства и можно наблюдать его деформации в одном модуле под влиянием воздушного потока (Рисунок 5.12).

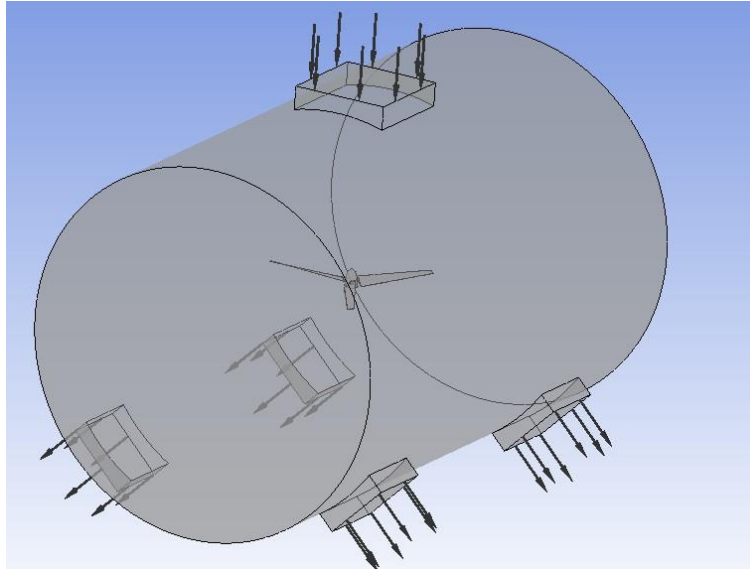


Рисунок 5.12 – Постановка задачи и заданные граничные условия

Из физического эксперимента получаем график угловой скорости вращения винта под воздействием набегающего потока, а также значения деформаций в точках расположения с тензодатчиков, расположенных на поверхности лопастей.

Тензодатчики при численном моделировании заменяются мониторинговыми точками (мон. точки), которые располагаются в тех же местах, что и на реальной лопасти (Рисунок 5.13, Таблица 35).

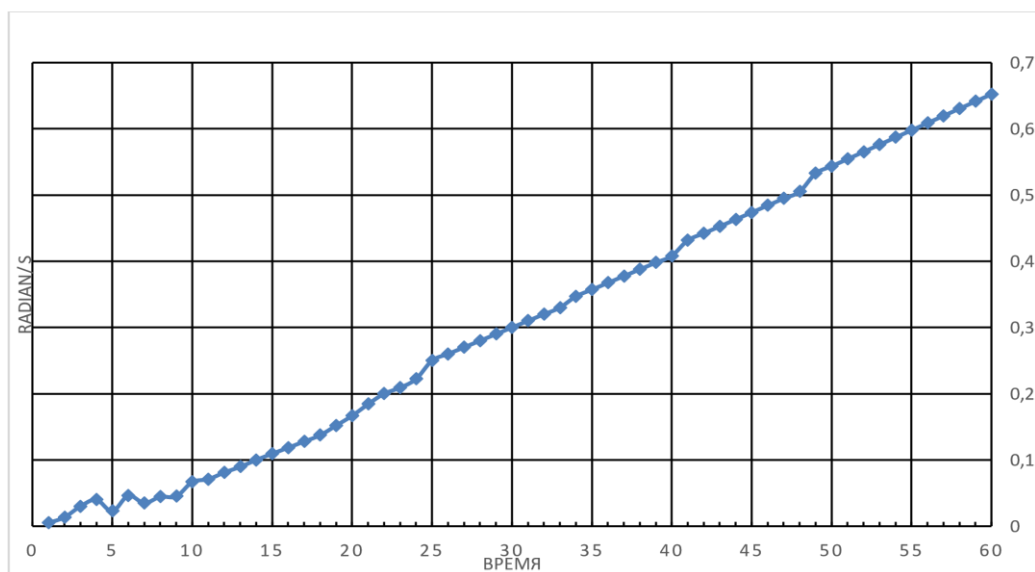


Рисунок 5.13 – Экспериментальная угловая скорость вращения винта в течении 60 секунд.

Значения деформаций лопастей с тензодатчиков.

Время , сек	Лопасть 1		Лопасть 2		Лопасть 3	
	Тензодатчи к 1, мм	Тензодатчи к 2, мм	Тензодатчи к 1, мм	Тензодатчи к 2, мм	Тензодатчи к 1, мм	Тензодатчи к 2, мм
1	+22	+10	+25	+9	+20	+9
2	+12	+5	-11	+7	-10	+5
3	+5	0	+9	+5	+7	+1
4	-3	-2	-3	0	-1	-2
5	-9	0	-5	0	-4	-2
6	-5	+3	-1	-2	0	0
7	+2	+5	+4	-4	+1	+2
8	+7	+1	+9	0	+5	+3
9	+7	-3	+8	+1	+2	+2
10	-5	-5	+6	+1	+6	+4
11	-9	-4	+5	+2	+8	+1
12	-8	-5	+1	-1	+8	-1
13	+1	-1	-5	-5	+4	-4
14	+10	-5	-1	-3	-2	-5
15	+9	-5	-6	-4	-1	-1
16	+2	-2	-6	-3	-3	-2
17	-4	+2	-6	-5	+4	+2
18	-9	+5	-4	-5	+10	+4
19	-8	+5	-1	-4	+3	+3
20	-4	+5	+7	-1	+5	+3
21	+1	+4	+9	+3	+4	+3
22	-1	+4	+7	+2	-3	-1
23	-7	+1	+9	+2	-2	-5
24	-2	-1	+8	+5	-8	-3
25	+1	-4	+4	+2	-6	-3
26	+7	-1	-1	-2	-9	-4
27	+4	+3	-1	-3	-5	-3
28	+7	+2	-5	-4	-1	-1
29	+5	+2	-9	-3	-9	+1
30	+5	+5	-4	+1	-2	+4
31	+1	+1	-1	+5	-4	+1
32	-1	-2	+6	+3	+3	+3
33	-4	-2	+3	+4	+10	+3
34	-2	-4	+7	+5	+2	+5
35	-6	-4	+7	+1	+3	+1
36	-7	-2	+4	+5	+9	+5

37	-1	+1	+5	+2	+5	+5
38	+8	+2	+4	+5	-1	+4
39	+3	+3	+8	+5	-5	+3
40	+5	+4	+9	+3	-9	-2
41	+8	+5	+2	-1	-10	-4
42	+8	+1	-1	-4	-8	-1
43	+9	-3	-1	-4	-2	-2
44	+9	-3	-8	-3	+7	-4
45	+6	-1	-7	-5	+5	-5
46	+1	-5	-10	-3	+3	-3
47	-5	-3	-3	-4	-4	-3
48	-7	-3	-5	-2	-6	-5
49	-6	-3	-1	+1	-8	-2
50	-3	+2	0	+2	-4	+4
51	-6	+4	+1	+3	+2	+5
52	+2	+2	+2	+5	+2	+3
53	+7	+5	+5	+1	+8	+5
54	+7	+2	+2	+4	+3	+3
55	+8	+5	-2	+3	+7	+4
56	+4	+1	-6	-1	+9	+3
57	+1	-2	-10	-3	+7	+3
58	-4	-5	-9	-4	+5	+2
59	-7	-1	-8	-5	+8	-1
60	-5	-5	-6	-1	+6	-4

Результатом моделирования также является график угловой скорости вращения, которые строятся в процессе расчета, и таблица с перемещениями, которые фиксировали мониторинговые точки, установленные на цифровой модели лопасти винта (Рисунок 5.14).

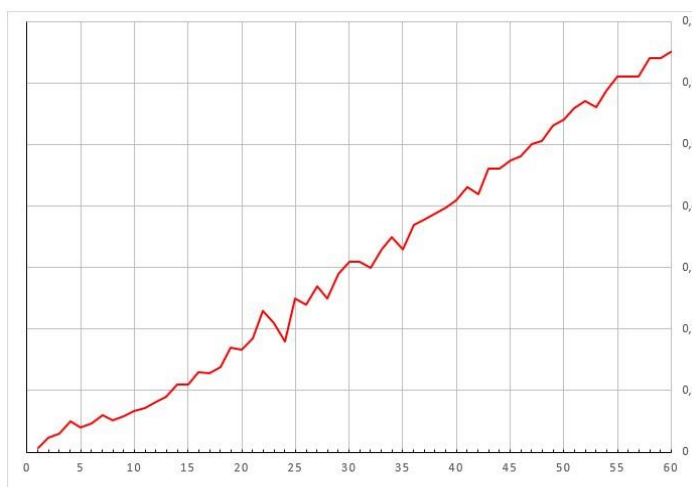


Рисунок 5.14 - Угловая скорость вращения винта в течении 60 секунд, полученная при моделировании.

Погрешность результатов по угловой скорости вращения между экспериментом и методом моделирования составляет 3%, что обусловлено недостаточно густой сеткой конечных элементов на лопастях винта, а также приближенными свойствами материала. Однако, метод позволил моделировать динамику винта за счет набегающего воздушного потока, кроме этого, стало возможно определять деформации тела в процессе взаимодействия в течении заданного промежутка времени (Рисунок 5.15-5.18, Таблица 36).

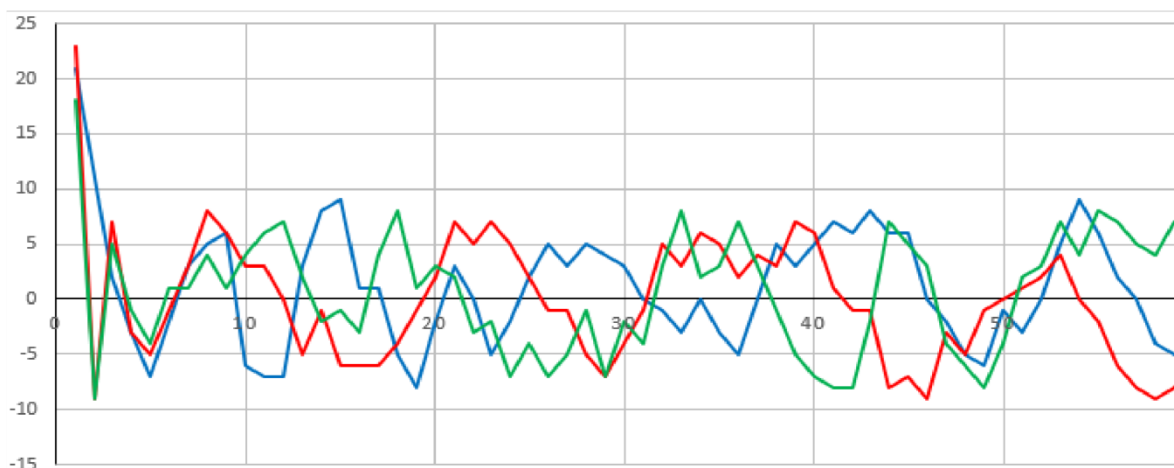


Рисунок 5.15 – Значения деформации для первой мониторинговой точки на лопастях (Синяя кривая – первая лопасть, красная кривая – вторая лопасть, зеленая кривая – третья лопасть)

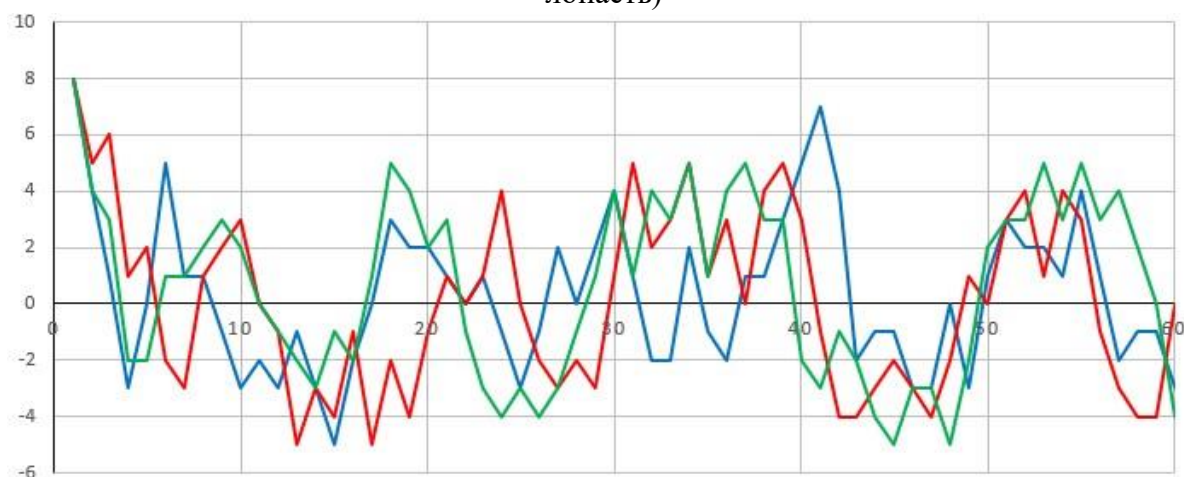


Рисунок 5.16 – Значение деформации для второй мониторинговой точки на лопастях (Синяя кривая – первая лопасть, красная кривая – вторая лопасть, зеленая кривая – третья лопасть)

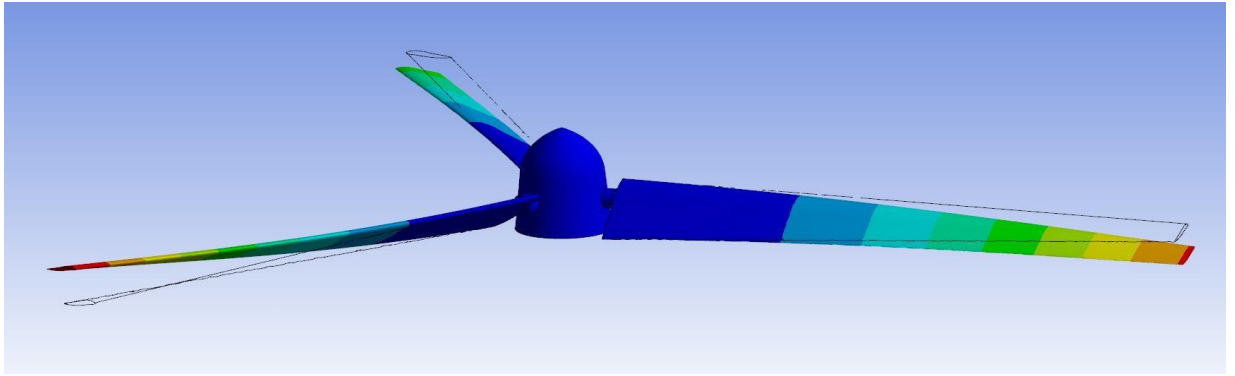


Рисунок 5.17 – Деформация лопастей винта на 30 секунде расчета

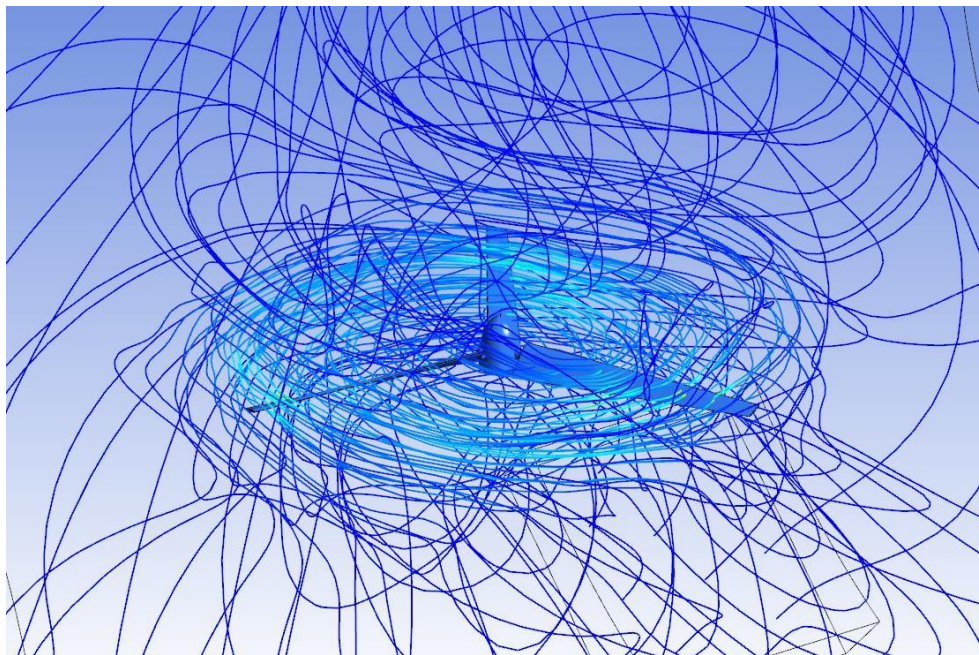


Рисунок 5.18 – Линии тока движения воздушного потока в области вращения лопастей винта на 60 секунде

Таблица 36

Погрешность ε по результатам деформации между экспериментом и методом моделирования

№	Лопасть 1		Лопасть 2		Лопасть 3	
	ε между тенз. 1 и мон. точкой 1	ε между тенз. 2 и мон. точкой 2	ε между тенз. 1 и мон. точкой 1	Погрешность между тенз. 2 и мон. точкой 2	Погрешность между тенз. 1 и мон. точкой 1	Погрешность между тенз. 2 и мон. точкой 2
1	3%	2%	4%	1%	2%	4%

Конечный процент погрешности вычислялся, как среднее арифметическое погрешностей на каждой секунде.

5.3 Техническая задача 2. Моделирование аэродинамики ветросолнечной установки

Целью исследования является определение максимальной скорости вращения лопаток генератора, прочностные показатели при заданном материале в процессе продувки (Рисунок 5.19). Данная задача – пример возможности применения созданного математического аппарата и программной среды для решения задач не только авиастроения.

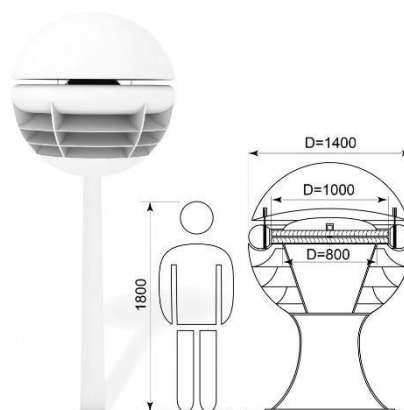


Рисунок 5.19. Внешний вид ветросолнечной установки

Требования:

1. Разрабатываемое изделие должно удовлетворять требованиям ГОСТ 15150-69 по категории УХЛ 1.1 (для эксплуатации на открытом воздухе, умеренного и холодного климата).
2. Структура материала эластично-жесткая, электроизоляционная, с пыле-влаго- и жироталкивающими свойствами, модуль упругости 8 000 - 10 000 МПа (81 600 - 102 000 кгс/см²).
3. Устойчивость к разрушению от ультрафиолетового излучения.
4. Диапазон рабочих температур установка 50...+50С.
5. Максимальная относительная влажность воздуха 100%.

Ветро-солнечная установка содержит в себе два лопаточных генератора расположенных в центре сферы с количеством лопаток равным 23 по всей длине окружности. Воздушный поток, заходя внутрь сферы, попадает на направляющий аппарат, который направляет его в кольцевой обтекатель. В дальнейшем воздушный поток движется по кольцевому обтекателю за счет вращения направляющего аппарата. Рабочее колесо начинает своё вращение за счет набегающего потока, скорость которого зависит от угловой скорости вращения направляющего аппарата и скорости потока на входе через нижний обтекатель (Рисунок 5.20).

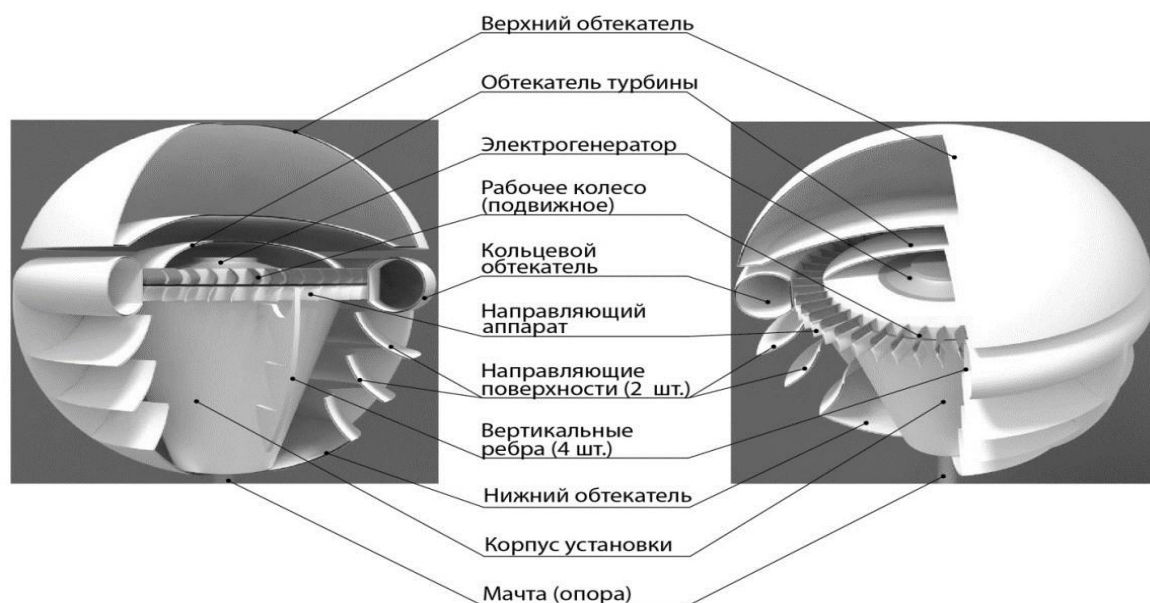


Рисунок 5.20 – внутреннее устройство ветро-солнечной установки

Необходимо провести ряд расчетов с различной скоростью движения воздушного потока, создаваемого направляющим аппаратом, при этом его вращение считается постоянным и основывается на различном потреблении энергии от солнечной батареи. Задача является решенной при условии обнаружения максимального значения КПД с учетом потребляемой от солнечной батареи энергии и выделяемой за счет вращения рабочего колеса на электрогенераторе

По чертежам созданы модели рабочего колеса и направляющего аппарата с заданной формой лопаток (Рисунок 5.21).

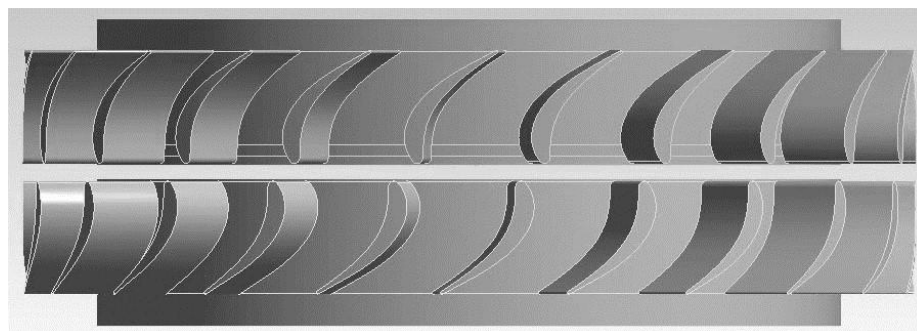


Рисунок 5.21 – Цифровые модели рабочего колеса и направляющего аппарата

При загрузке в авторский программный комплекс созданных моделей и воздушного пространства указывались граничные условия непроницаемости для стенок внутреннего строения обтекателя. Входным параметром, изменяемым с течением времени, является скорость воздушного потока, создаваемая направляющим аппаратом. Продолжительность работы при каждом входном параметре составляет 24 часа, с получением результатов выделяемой мощности за счет вращения рабочего колеса и потребляемой направляющим аппаратом, также для определения прочности опоры ветро-солнечной установки рассчитывалась осевая нагрузка, создаваемая совместным вращением двух элементов конструкции.

В результате расчета получены графики КПД, мощности и осевой нагрузки от скорости воздушного потока, создаваемого направляющим аппаратом с течением времени (Рисунок 5.22). Весь процесса расчета проходил в автоматизированном состоянии при изменении скорости потока в пределах, указанных пользователем. Каждый получаемый результат сохранялся в базу данных программного комплекса, параметры которых комбинировались на последующих расчетах для получения максимального значения искомой величины.

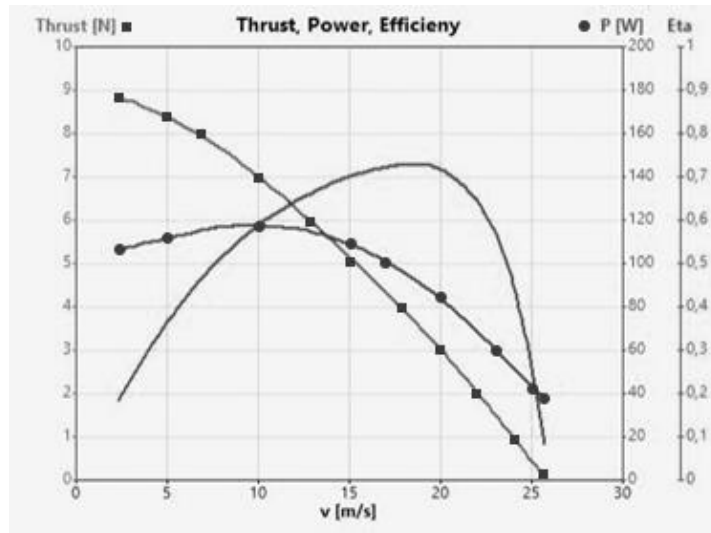


Рисунок 5.22 – График скорости потока, мощности и КПД

5.4 Техническая задача 3. Моделирование резонансных колебаний лопастей воздушного винта, анализ образования вихрей

Совместно с Конструкторским бюро «Беркут» решалась задача обнаружения резонансных колебаний на лопастях винта вертолета.

Цель работы заключается в:

1. Определение турбулентных потоков в области движения лопастей винта и их влияния на прочностные характеристики винта.
2. Определение турбулентных потоков для двух видов вращения винта: прямой ход и реверсный ход.
3. Поиск возможных частот резонансных колебаний и прочностных характеристик исследуемого тела для оптимизации профиля и геометрии лопасти винта.

Исследуемым объектом является лопасть винта вертолета, которая подвергается разрушению при наборе начальной угловой скорости вращения, а также при посадке. Дополнительно требовалось рассчитать степень влияния набегающего воздушного потока на лопасти вертолета при его перемещении на различных высотах полета.

Моделирование проводилось винта с тремя лопастями и диаметром 3,9 метров.

В результате моделирования были обнаружены участки турбулентных завихрений воздушного потока, которые формировали на лопасти периодические

нагрузки, приводящие к ее колебаниям. Это приводит к образованию эффекта вихревого кольца, способствующего снижению подъемной силы.

Максимальное напряжение, возникающее в лопасти при ее крутильных колебаниях в процессе вращения с учетом воздушной среды, достигает на краях 590 МПа на 21 секунде расчета при частоте свободных колебаний 1339 Гц с учетом результатов перемещения точек тела из авторского вычислительного модуля CFXDisp (Рисунок 5.23).

Предел прочности для анизотропного материала лопасти:

- При растяжении – 380 МПа,
- При сжатии – 245 МПа
- При статическом изгибе – 430 МПа
- При межслоевом сдвиге – 70 МПа

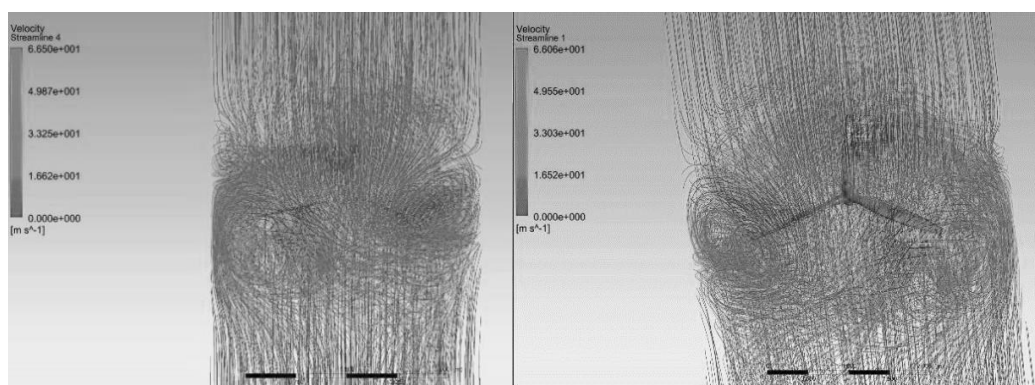


Рисунок 5.23 – Вихри, возникающие в окрестностях концов лопастей винта и снижающие эффективность винта

По полученным изображениям видно, то что воздушные вихре, образованные впереди движущимися лопастями воздействуют на наступающие лопасти, что приводит к возникновению колебаний и торможению лопасти.

Из результатов моделирования с первоначальной геометрией лопасти возникают колебательные движения с высокой амплитудой, показывающее резкий скачок перемещения, данный скачок был зафиксирован программной средой и дальнейший процесс записи был остановлен, так как произошло превышение значения прочности материала на растяжение (Рисунок 5.24).

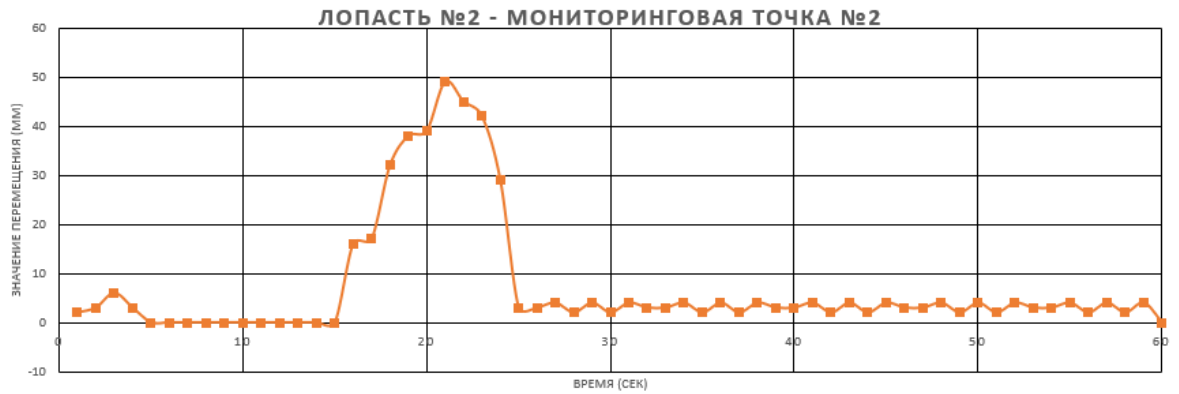


Рисунок 5.24. График снятия показаний

При перерасчете значений напряжения определилось, что концентраторами напряжения становятся кромки лопасти по обеим сторонам, результаты показали о том, что напряжение в данных кромках достигает 590 МПа (Рисунок 5.25).

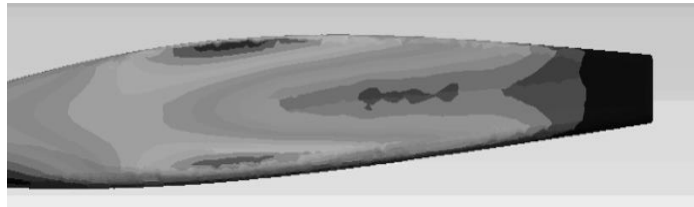


Рисунок 5.25 – Концентраторы напряжений на поверхности лопасти

В результате работы были получены значения частот свободных колебаний, при которых достигаются высокие значения напряжений и перемещений точек тела лопасти. Было замечено, что данные значения возникают при прохождении лопасти через турбулентный вихрь, создаваемый впереди движущейся лопастью, а также в процессе набора рабочей угловой скорости вращения, возникают малые изгибные и крутильные колебания при взаимодействии с окружающим пространством. На основе полученных результатов было предложено изменение профиля и геометрии лопасти и установка дополнительных конструктивных элементов на её поверхности для увеличения прочностных показателей.

ЗАКЛЮЧЕНИЕ

При выполнении поставленных задачи диссертационной работы были достигнуты основные результаты.

1. Исследованы существующие методы моделирования динамического взаимодействия твердых тел с окружающим пространством с целью обнаружения частот их резонансных колебаний. Для решения технической задачи в области аэродинамики были рассмотрены исследованы вариационно-сеточный статистический метод анализа и вычислительный метод для моделирования динамики жидкости и газов (SPH) для определения напряженно-деформированного состояния.

2. Разработаны новые конечно-элементные дискретные модели, а также численный метод моделирования динамического взаимодействия твердых тел с окружающей средой в программном комплексе ANSYS, позволяющий определять частоты резонансных колебаний твердых тел.

3. Разработаны авторские вычислительные модули для определения значений перемещения точек тела внутри программной оболочки ANSYS WorkBench с сохранением информации в авторском формате файла, который используется для поиска резонансных колебаний.

4. Созданы структуры нового метода конечных элементов с использованием ортогональных финитных функций, который позволяет устранить проблемы, связанные с высокими требованиями к вычислительной мощности и необходимой точности результатов. Методы конечных элементов с применением ортогональных финитных функций впервые построены на языке программирования Fortran в среде ANSYS WorkBench.

5. Получены верифицирующие решения плоских задач и трехмерных задач упругости с использованием новых 2D и 3D конечных элементов. Получены таблицы и графики эффективности, разработанных КЭ при решении задачи теории упругости.

6. Создан авторский программный комплекс, основанный на использовании алгоритмов нового метода моделирования при помощи конечных элементов на ОФФ, для исследования и решения актуальных технических задач аэродинамики.

7. Создана авторская база данных сбора, хранения и обработки параметров о постановке задачи аэрогидроупругости и процедура выдачи рекомендаций по формированию граничных условий на основе параметров решенных технических задач.

8. Создан препостпроцессор, предназначенный для визуализации результатов решения трехмерных задач теории аэрогидроупругости.

9. Выполнены решения актуальных технических задач динамического взаимодействия упругих лопастей воздушного винта с газовой средой.

10. На основе проведенного математического моделирования получены картины напряженно-деформированного состояния упругих лопастей воздушного винта в каждый момент времени при их динамическом взаимодействии с окружающей средой.

11. На основании результатов решения актуальных технических задач от применения авторского программного комплекса с использованием разработанного метода расчета аэродинамики и новых КЭ с ОФФ получены акты о внедрении результатов диссертационной работы от ООО КБ «Беркут» (от 26.11.2017), занимающегося

производством элементов конструкций летательных аппаратов и соответствующего оборудования, и ЗАО «КАДФЕМ Си-Ай-Эс» (от 16.10.19), деятельность которого связана с использованием вычислительной техники и информационных технологий. Акты о внедрении подтверждают научную новизну и практическую значимость диссертационной работы.

СПИСОК ЛИТЕРАТУРЫ

1. Yu, S.D., Warwick, S.A., Zhang, X. (2009). Nonlinear dynamics of a simplified engine-propeller system. *Communications in Nonlinear Science and Numerical Simulation*, 14 (7), 3149–3169.
2. Kamarlouei, M., Ghassemi, H., Aslansefat, K., Nematy, D. (2014). MultiObjective Evolutionary Optimization Technique Applied to Propeller Design. *Acta Polytechnica Hungarica*, 11, 163–182.
3. Михеев С.В. Пути совершенствования винтокрылых летательных аппаратов Монография. — М.: МАИ, 2006. — 64 с.
4. Панасенко Б.А. Конструирование эластомерных подшипников Учебное пособие по курсовому и дипломному проектированию. — Харьков: ХАИ, 1982. — 51 с.
6. Проскурин В.Д. Расчет параметров вертолета на этапе предварительного проектирования Учебное пособие.— Оренбург : ОГУ, 2014 - 199 с.
7. Резниченко В.И. Изготовление лопастей вертолетов из неметаллических материалов М.: Изд-во МАИ, 1977. — 61 с 8. Ружицкий Е.И. (ред) Авиастроение. Том 10. Зарубежные вертолёты М.: ВИНТИ, 1989. — 116 с.
9. Русанович Н.Г. Материалы по проектированию муфт включения винтокрылых аппаратов Технические отчеты. №91 —Труды ЦАГИ, издательство Бюро новой техники, 1947. — 17 с.
10. Armendáriz, I., Olarrea, J., García-Martínez, J. (2018). Engine to wing structural design under critical loads caused by a propeller blade loss. *Engineering Structures*, 158, 155–163.
11. Слюсарь Б.Н., Флек М.Б. и др. Технология вертолетостроения. Технология производства лопастей вертолетов и авиационных конструкций из полимерных композиционных материалов Ростов-на-Дону, Издательство ЮНЦ РАН, 2013. — 230 с.
12. Сохань О.Н. Конструирование втулок несущих винтов вертолетов М.: МАИ, 1981. - 53 с. 12. Сохань О.Н. Конструирование втулок несущих винтов вертолетов М.: МАИ, 1981. — 56 с.
13. Bertoglio, C., Gaggero, S., Rizzo, C. M., Vaccaro, C., Viviani, M. (2015). Influence of propeller characteristics on propeller structural design. In *Proceedings of the 16th International Congress of the International Maritime Association of the Mediterranean, IMAM*.
14. Klyavin, O., Borisenko, V., Gichev, Y., Boldyrev, Y. (2018). Mathematical Modeling Step Responses of a Pitch-Change Mechanism for the Altitude Propeller. *J. of Adv. Research in Dynamical & Control Systems*, 10, 2085–2089.
15. Aleshin, M., Smirnov, A., Murzina, M., Boldyrev, Y. (2018). On Structural Optimization of the Propeller Blade. *Int. J. of Engineering & Technology*, 7 (4.36), 1104– 1109.
16. Зленко Н.А., Кедров А.В., Кишалов А.Н. Оптимальное аэроакустическое проектирование воздушного винта ченые записки ЦАГИ Тол XLII 6, 2011 //Москва с. 92 — 102.

17. Лен С.А. Разработка метода расчета параметров воздушного винта//Барнаул, 2017г.
18. Левшонков Н.В. Методика проектировочного расчета и рациональный выбор параметров воздушного винта при разработке многорежимных летательных аппаратов// Казань, 2015. 47 Шувалов А.А. Аэродинамическое проектирование воздушного винта// АОН, 1999.
19. Герасимов О.В. Крицкий Б.С. Расчет воздушного винта беспилотного летательного аппарата с учетом числа Рейнольдса и степени редукиции. // Научный вестник МГТУ ГА (200). –2014. –№2. с. 79 – 85.
20. Лысенков, А.В. Разработка методологии расчёта аэродинамических характеристик воздушных винтов. [Электронный ресурс] / А.В. Лысенков, С.В. Павлик. – Электрон. текст. дан. // ТРУДЫ МФТИ: науч. журн. / – 2013. – Т. 5. – С.174 – 186. – Режим доступа:<https://mipt.ru/upload/e67/174-186-arphj8g0g1k.pdf>, свободный. (Дата обращения: 28.03.2019).
21. Ganguli R. Smart Helicopter Rotors: Optimization and Piezoelectric Vibration Control Springer, 2016. — 264 p. — ISBN 978-3-319-24766-3 ISBN 978-3319-24768-7 (eBook).
22. Guowei Cai, Ben M. Chen, Tong Heng Lee. Unmanned Rotorcraft Systems London: Springer-Verlag London Limited, 2011. — 288 p.
23. Dover Publications, 1994. 1120 p. ISBN:0486682307
24. Stepniewski W.Z. A comparative study of Soviet vs. Western helicopters. Part 1. General comparison of designs International Technical Associates, Ltd. Upper Darby, Pennsylvania, 1983. — 298 p.
25. Stepniewski W.Z. A comparative study of Soviet vs. Western helicopters. Part 2. Evaluation of Weight, Maintainability, and Design Aspects of Major Components International Technical Associates, Ltd. Upper Darby, Pennsylvania, 1983. — 175 p.
26. Watkinson J. Art of the Helicopter Oxford: Butterworth-Heinemann, 2004. — 390 p. — ISBN:0750657154 .
27. Александрин Ю.С., Кудряшов А.Б. и др. Использование программного обеспечения MSC.Software для сопровождающего моделирования испытаний на усталость и живучесть вертолетных конструкций Москва: 2009 - 19 с.
28. Ананьев И.В. Расчет вибраций и выносливость элементов конструкции геликоптера М.: ЦАГИ, 1950. — 68 с
29. Баршевский В.Б. Вертолет в полете Монография. — М.: ДОСААФ, 1954. — 88 с.
30. Башаров Е.А. Конструирование агрегатов вертолетов Методическое пособие для курсового проектирования. — Москва: МАИ, 2016. — 240 с.
31. Братухин И.П. Автожиры. Теория и расчёт Ленинград: Госмаштехиздат, 1934. — 102 с.
32. Братухин И.П. Проектирование и конструкции вертолетов Москва: Государственное издательство оборонной промышленности, 1955. — 358 с.

33. Гребеников А.Г., Лосев Л.И., Урбанович В.А., Чумак А.С. Методика, алгоритм и программа определения параметров общего вида одновинтового вертолета Учеб. пособие. — Х.: Нац. аэрокосм. ун-т "Харьк. авиац. ин-т", АНТООО "КНК", 2009. — 105 с. ISBN 978-966-662-183-5
34. Далин В.Н. Конструирование деталей и узлов вертолета М.: МАИ, 1977г. - 53 с.
35. Джонсон У. Теория вертолетов. Книга 1 В 2-х книгах. Перевод с английского. — М.: Мир, 1983. — 502 с.
36. Джонсон У. Теория вертолетов. Книга 2 В 2-х книгах. Перевод с английского. М.: Мир, 1983. - 529 с.
37. Eren, Z., Tataroglu, S., Balkan, D., Mecitoglu, Z. (2017). Modeling of Bird Strike on a Composite Helicopter Rotor Blade. 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (USA, Texas, Grapevine, 9-13 Jan., 2017).
38. Iwase, T., Kishitani, T., Furukawa, M. (2017). Study on Influence of Blade Number on Aerodynamic Noise of Half-ducted Propeller Fans for Packaged Airconditioners. International Journal of Fluid Machinery and Systems, 10(4), 318–327.
39. Yang, Y., Veldhuis, L.L.M., Eitelberg, G. (2017). Aerodynamic impact of a stream wise vortex on a propeller. Aerospace Science and Technology, 70, 108–120.
- 40.
41. Дмитриев И.С., Есаулов С.Ю. Системы управления одновинтовых вертолётов М.: Машиностроение, 1969. — 220 с.:
42. Дьяченко Ю.В. Технология изготовления лопастей вертолётов Издательство: Харьковский Авиационный Институт, 1992. - 54 с.
43. Жустрин Г.К., Кронштадтов В.В. Весовые характеристики вертолета и их предварительный расчет М.: Машиностроение, 1978. — 112 с.
44. Завалов О.А. Конструкция вертолета Учебник для студентов вузов, обучающихся по специальности "Самолето- и вертолетостроение" направления подгот. дипломированных специалистов "Авиастроение". — Моск. авиац. ин-т (гос. техн. ун-т). — М. : МАИ, 2004. — 314 с. : ил. — ISBN 5-7035-1397-9.
45. Завалов О.А. Конструкция несущих и рулевых винтов вертолетов Учебное пособие к курсовому и дипломному проектированию. — Москва: МАИ, 2001. — 72 с.
46. Изаксон А.М. Советское вертолетостроение 2-е изд., перераб. и доп. Учебник. — М.: Машиностроение, 1981. — 295 с.
47. Кирпикин А.А. Приближённый расчёт резонансных диаграмм несущих винтов вертолётов Учеб. пособие — Харьков: ХАИ, 1999. — 62 с.
48. Кирпикин А.А. Расчет нагрузок на фюзеляжи самолетов и вертолетов Учеб. пособие. — Харьков: Харьк. авиац. ин-т, 1992. — 96 с.
49. Кривцов В.С., Карпов Я.С., Лосев Л.И. Проектирование вертолетов Харьков: Национальный аэрокосмический университет "Харьковский авиационный институт", 2003. — 344 с. — ISBN 966-662-046-4.

49. Кривцов В.С., Карпов Я.С., Лосев Л.И. Проектирование вертолетов Харьков: Национальный аэрокосмический университет "Харьковский авиационный институт", 2003. — 344 с.
50. Курочкин Ф.П. Конструирование винтов, силовых установок и приводов вертолета Учебное пособие. — М.: МАИ, 1980. — 140 с.
51. Лебедев И.М. Расчет торсиона несущего винта в системе MSC.Nastran КГТУ (КАИ) им. А. Н. Туполева, Казань. - 10 с.
52. Лосев Л.И. Приближенное определение основных параметров вертолета Учебное пособие. — Харьков: ХАИ, 1988. — 54 с.
53. Лосев Л.И., Рябков В.И. Выбор параметров соосного вертолета по критерию минимума взлетной массы Учебное пособие по курсу "Проектирование вертолетов". — Харьков: ХАИ, 1999. — 101 с.
54. Лосев Л.И., Урбанович В.А. Алгоритм и программа определения взлетной массы вертолета одновинтовой схемы Учебное пособие по дипломному проектированию. — Харьков: ХАИ, 1992. — 55 с.
55. Миль М.Л., Некрасов А.В., Браверман А.С. и др. Вертолеты, расчет и проектирование. Том 2. Колебания и динамическая прочность Москва: Машиностроение, 1967. — 424 с.
56. Михеев Р.А. Прочность вертолетов Учебник для авиационных специальностей втузов. — М.: Машиностроение, 1984. — 280 с.
57. Михеев Р.А., Скулков Д.Д. Вибрации вертолета и средства их гашения Учебное пособие. — М.: МАИ, 1993. — 40 с.
58. Тищенко М.Н. и др. Вертолеты. Выбор параметров при проектировании М.: Машиностроение, 1976. — 368 с.
59. Тищенко М.Н., Некрасов А.В., Радин А.С. Вертолеты. Выбор параметров при проектировании Москва: Машиностроение, 1976. - 368 с.
60. Туркина А.И. Расчет на прочность винтов современных вертолетов Учебное пособие. — М.: МАИ, 1990. — 32 с. 69. Хафер К., Закс Г. Техника вертикального взлета и посадки Пер. с нем. — М.: Мир, 1985. – 376 С.
61. Ефременков И.В., Горбунов И.В., Леонтьев В.Л., Гисметулин А.Р. Особенности моделирования процессов механической обработки в cae-системах. Известия Самарского научного центра Российской академии наук. 2013. Т. 15. № 4-4. С. 846-853.
62. Ефременков И.В. Гисметулин А.Р. Горбунов И.В. Разработка препроцессора для моделирования операций механообработки в cae системе ls-dyna. Известия Самарского научного центра Российской академии наук. 2014. Т. 16. № 1-5. С. 1338-1342.
63. Ефременков И.В., Леонтьев В.Л. О взаимодействии упругих лопастей вращающегося винта с газовой средой. Известия Самарского научного центра Российской академии наук. 2012. Т. 14. № 4-2. С. 644-646.

64. Ефременков И.В., Гисментулин А.Р., Горбунов И.В. Моделирование процесса сверления с помощью sph и конечно-элементного методов. Известия Самарского научного центра Российской академии наук. 2014. Т. 16. №1-5. С. 1346-1351.
65. Леонтьев В.Л., Ефременков И.В. О моделировании резонанса вращающегося вертолетного винта. Известия Самарского научного центра Российской академии наук. 2013. Т. 15. № 4-3. С. 698-701.
66. Леонтьев В.Л., Ефременков И.В. Конечный элемент на основе ортогональных финитных функций в плоских задачах теории упругости Ученые записки Ульяновского государственного университета Сер. "Математика и информационные технологии" Ульяновск, 2016. С. 56-60.
67. Леонтьев В. Л. Методы конечных элементов, основанные на использовании обобщенных функций Куранта в теории упругих колебаний / В. Л. Леонтьев // Проблемы динамики и прочности электро- и энергомашин. Тезисы докладов Всероссийского научного семинара (18-20 мая 1993 г.). — СанктПетербург : Изд-во Института проблем машиноведения РАН, 1993. — С. 21-22.
68. Леонтьев В. Л. Об одном обобщении функций Куранта / В. Л. Леонтьев // Теория функций и приближений. Труды 7-й Саратовской зимней школы (30 января – 4 февраля 1994 г.). Часть 3. — Саратов : Изд-во СГУ, 1994. — С. 36-40.
69. Леонтьев В. Л. Об ортогональных финитных функциях второй степени, связанных с треугольными сетками, и их применении в математическом моделировании / В. Л. Леонтьев, А. В. Кочулимов // Вестник 150 Самарского государственного технического университета. — 2011. — Т. 22. — №1. — С. 241243. — ISSN 1991-8615.
70. Леонтьев, В. Л. Ортогональные финитные функции и численные методы / В. Л. Леонтьев. — Ульяновск : Изд-во УлГУ, 2003. — 177 с. — ISBN 588866-144-9.
71. Леонтьев В.Л. Ортогональные финитные функции, смешанные вариационные принципы в численных методах: Диссертация доктора физико-математических наук : 05.13.18 / Леонтьев Виктор Леонтьевич. — Ульяновск, 2002. — 390 с.
72. Ширяльщикова В.И., Резниченко В.И. Общая сборка вертолетов. Учебное пособие. — М.: МАИ, 1979. — 61 с.
73. Cheng, A. and D. T. Cheng. Heritage and early history of the boundary element method, *Engineering Analysis with Boundary Elements*, 29, 268—302. 2005
74. Abramowitz M., Stegun I. A. *Handbook of Mathematical Functions*. — 10th printing with corrections. — Dover, 1972. — ISBN 978-0-486-61272-0. Equation 25.4.45
75. *Applications of the Jacobian to Analysis and Geometry*, 2011, 94
76. Любимова А.А. Численное интегрирование с помощью квадратурных формул Гаусса. Южно-Уральский государственный университет филиал в г. Златоусте
77. Kruger, W., Dillinger, J., De Breuker, R., Reyes, M., Haydn, K. (2016). Adaptive Wing: Investigations of Passive Wing Technologies for Loads Reduction in the Clean Sky Smart

Fixed Wing Aircraft (SFWA). Greener aviation (Belgium, Brussels, 1113 Oct., 2016), Paper ID №122, 12 p.

78. Patel, Y., Gaurav, A., Srinivas, K., Singh, Y. (2017). A review on design and analysis of the propeller used in UAV. *Int. J. Adv. Prod. Ind. Eng.*, 605, 20-23.

79. Leontiev V.L., Efremenkov I.V. Finite element modeling and investigation of elastic homogeneous and heterogeneous materials. *Materials Physics and Mechanics*. 2019. Т. 42. № 3. С. 340-350.

80. Morgado, J., Abdollahzadeh, M., Silvestreetal, M.A.R. (2015). High altitude propeller design and analysis. *Aerospace Science and Technology*, 45, 398–407.

81. Ефременков И.В. Средства для проведения расчета напряженно-деформированного состояния сложных геометрических конструкций. *Актуальные проблемы современной науки в 21 веке*. 2015.С. 27-28.

82. Леонтьев В.Л., Ефременков И.В. Методические рекомендации по выполнению лабораторных работ, связанных с решением задач аэрогидродинамики и с использованием комплекса программ ansys workbench. Ульяновский государственный университет. 2017.

83. Ефременков И.В. Расчет аэродинамических показателей взаимодействия лопастей винта с окружающей средой. *Наука, теория, практика авиационно-промышленного кластера современной России*. 2017.С.23-26.

84. Ефременков И.В. математическое моделирование динамического взаимодействия твердого тела с окружающим его пространство. *Universum: технические науки*. 2019. №6 (63). С.6-8.

ПРИЛОЖЕНИЕ А

Код подпрограммы на языке Fortran, для внедрения ОФФ в 3D КЭ

UserElem.f – файл-библиотека, описывающая вычислительную работу конечного элемента.

```

subroutine UserElem (elId, matId, keyMtx, lumpm, nDim, nNodes,
&
Nodes, nIntPnts, nUsrDof, kESTress,
&
keyAnsMat, keySym, nKeyOpt, KeyOpt,
&
temper, temperB, tRef, kTherm,
&
nPress, Press, kPress, nReal, RealConst,
&
nSaveVars, saveVars, xRef, xCur,
&
TotValDofs, IncValDofs, ItrValDofs,
&
VelValDofs, AccValDofs,
&
kfstps, nlgeom, nrkey, outkey, elPrint, iott,
&
keyHisUpd, ldstep, isubst, ieqitr, timval,
&
keyEleErr, keyEleCnv,
&
eStiff, eMass, eDamp, eSStiff,
&
fExt, fInt, elVol, elMass, elCG,
&
nRsltBsc, RsltBsc, nRsltVar, RsltVar,
&
nElEng, elEnergy)
&

#include "impcom.inc"
c
EXTERNAL      ElemGetMat

INTEGER      elId, matId, keyMtx(10), lumpm, nDim, nNodes,
&
Nodes(nNodes), nIntPnts, nUsrDof, kESTress,
&
keyAnsMat, keySym, nKeyOpt, KeyOpt(nKeyOpt),
*
kTherm, nPress, kPress, nReal, nSaveVars,
&
kfstps, nlgeom, nrkey, outkey,
&
elPrint, iott, keyHisUpd,
&
ldstep, isubst, ieqitr, keyEleErr, keyEleCnv,
&
nRsltBsc, nRsltVar, nElEng

DOUBLE PRECISION temper(nNodes), temperB(nNodes), tRef,
&
Press(nPress), RealConst(nReal),
&
saveVars(nSaveVars),
&
xRef(nDim,nNodes), xCur(nDim,nNodes),
&
TotValDofs(nUsrDof), IncValDofs(nUsrDof),
&
ItrValDofs(nUsrDof), VelValDofs(nUsrDof),
&
AccValDofs(nUsrDof), timval,
&
eStiff(nUsrDof,nUsrDof), eMass(nUsrDof,nUsrDof),
&
eDamp(nUsrDof,nUsrDof), eSStiff(nUsrDof,nUsrDof),
&
fExt(nUsrDof), fInt(nUsrDof),
&
elVol, elMass, elCG(3),
&
RsltBsc(nRsltBsc), RsltVar(nRsltVar),
&
elEnergy(nElEng)

#include "locknm.inc"

EXTERNAL      vzero, vmove, vmult, vdot, vidot,
&
maxv, matxb, matba, maat, matsym, getMatProp,
&
erhandler, equivStrain, ElemJac, ElemMass,
&
ElemRsltNode, ElemShpFn, pplock, ppunlock

DOUBLE PRECISION vdot, vidot, TBMat(8,8), Stif(8,8)

INTEGER      nUsrDof2, intPnt, iNode, nTens, flgSingular,
&
k1, k2, k3, nComp, iDim, iDim1, iComp,
&
nNodesCorner, nDirect, kThermIP, n, ic, ir
DOUBLE PRECISION BMat(nDim*2,nUsrDof), Ex, nu, density, G, workDb,
&
con1, con2, cMat(nDim*2,nDim*2), shIsoC(nNodes),
&
shIso(nNodes), shDerIso(nDim,nNodes), wtIP(1),
&
workArr(360), elJac(nDim*nDim), detJac, dperr(2),
&
shDerEl(nDim,nNodes), dVol, Strain(nDim*2),
&
Stress(nDim*2), wStrain(48), wStress(48),

```

```

&          nStrain(48), nStress(48), sigm, tem, prop(3),
&          IncStrain(nDim*2), defG(3,3), Pnt(3,20),Nn(20),
&          defG0(3,3), xCurIP(nDim), TemperIP, dN(3,20),
&          TemperIPB, StressTh(nDim*2), MatProp(5),
&          StrainPl(nDim*2), StrainCr(nDim*2),
&          StrainTh(nDim*2), StrainSw, StressBk(nDim*2),
&          MatRotGlb(3,3), wStrainTh(48), wStrainPl(48),
&          wStrainCr(48), eMassb(nNodes,nNodes), EnergyD(3),
&          c0,c1,c2,c3,u1,u2,u3,u4,u,uu,d,c(8,8),xc
CHARACTER*4          label(3)

double precision s,t,ft(20),fs(20), fk(20), ft2,fs2,ftX(4),fsX(4)

INTEGER debug, ix,i,j,k,r,iPnt,jPnt

DO i = 1, 3
  DO j = 1, 20
    Pnt(i,j)=0.877465543
  END DO
END DO

nTens = nDim*2
nComp = nDim*nDim
nDirect = 3
nUsrDof2 = nUsrDof*nUsrDof
CALL vzero (BMat(1,1),nUsrDof*nTens)
IF (keyMtx(1).EQ.1) CALL vzero (eStiff(1,1),nUsrDof2)
IF (keyMtx(2).EQ.1) CALL vzero (eMass(1,1) ,nUsrDof2)
IF (keyMtx(5).EQ.1) CALL vzero (fExt(1) ,nUsrDof)
IF (keyMtx(6).EQ.1) CALL vzero (fInt(1) ,nUsrDof)
IF (nlgeom.EQ.0) THEN
  DO iDim = 1, 3
    DO iDim1 = 1, 3
      defG0(iDim, iDim1) = 0.0D0
    END DO
    defG0(iDim, iDim) = 1.0D0
  END DO
  CALL vmove (defG0(1,1),defG(1,1),9)
ELSE
END IF
elVol = 0.d0
elMass= 0.d0
IF (keyMtx(2).EQ.1) CALL vzero (eMassb(1,1),nNodes*nNodes)
CALL vzero (elEnergy(1), nElEng)

IF (debug.EQ.1) THEN
  write (*,*)
  write (*,*)'elId =',elId,
& ' keyAnsMat=',keyAnsMat,' nDim=',nDim, ' kEStress=',
& kEStress, ' keySym=',keySym, ' nIntPnts=',nIntPnts
END IF

IF (KeyOpt(1).EQ.0) THEN
  CALL ElemShpFn (1, 0, 1, shIsoC(1), nNodes)
  ELSE IF (KeyOpt(1).EQ.1) THEN
  CALL ElemShpFn (2, 0, 1, shIsoC(1), nNodes)
ELSE
  dperr(1) = KeyOpt(1)
  dperr(2) = elId
  CALL erhandler ('UserElem', 1000, 3, 'Invalid input
& KEYOPT(1) = %I at element %I.',dperr(1),
& ' ')
END IF
elCG(3) = 0.0d0
DO iDim = 1, nDim
  elCG(iDim) = vidot (shIsoC(1), 1, xCur(iDim,1), nDim,
& nNodes)
&
END DO

IF (keyAnsMat.EQ.1) THEN

ELSE

  label(1) = 'ex '

```

```

label(2) = 'prxy'
label(3) = 'dens'
tem = vdot (shIsoC(1), temper(1), nNodes)

CALL getMatProp (elId,matId,tem,3,label(1),prop(1))
Ex = prop(1)
nu = prop(2)
density = prop(3)

IF (nReal.GE.1 .AND. Ex .EQ.0.0d0) Ex = RealConst(1)
IF (nReal.GE.2 .AND. nu .EQ.0.0d0) nu = RealConst(2)
IF (nReal.GE.3 .AND. density.EQ.0.0d0) density = RealConst(3)

G = 0.5d0*Ex/(1.0d0+nu)
workDb = (1.0d0+nu)*(1.0d0-2.0d0 * nu)
con1 = (1.0d0-nu)*Ex/workDb
con2 = nu*Ex/workDb
CALL vzero (cMat(1,1), nTens*nTens)
cMat(1,1) = con1
cMat(2,2) = con1
cMat(3,3) = con1
cMat(4,4) = G
cMat(1,2) = con2
cMat(1,3) = con2
cMat(2,3) = con2
cMat(2,1) = con2
cMat(3,1) = con2
cMat(3,2) = con2
IF (KeyOpt(1).EQ.1) THEN
    cMat(5,5) = G
    cMat(6,6) = G
END IF
END IF
kTherm = 0

DO 100 intPnt = 1, nIntPnts

    IF (KeyOpt(1).EQ.0) THEN

c-----

        !добавленная процедура вычисления ортогональных финитных функций
        include "OFF3D.for"

        !вычисление вспомогательных функций формы на основе ОФФ
        include "OFFX3D.for"

        do i=1,nNodes
            shIso(i)=Nn(i)
        end do

    END IF

    DO iDim = 1, nDim
        xCurIP(iDim)=vidot(shIso(1),1,xCur(iDim,1),nDim,nNodes)
    END DO

c-----

        !вычисление производных функций формы
        include "dN3D.for"

        do i=1,nDim
            do j=1,nNodes
                shDerIso(i,j)=dN(i,j)
            end do
        end do

c-----

    CALL vzero (workArr(1), nComp)
    iComp = 1
    DO iDim = 1, nDim
        DO iDim1 = 1, nDim
            DO iNode = 1, nNodes

```

```

        workArr(iComp) = workArr(iComp)
&          + shDerIso(iDim1,iNode)*xCur(iDim,iNode)
        END DO
        iComp = iComp + 1
    END DO
END DO
CALL ElemJac (workArr(1), elJac(1), nDim, detJac,
&             flgSingular)
& IF (flgSingular.LE.0) THEN
    dperr(1) = detJac
    dperr(2) = elId
    CALL erhandler ('UserElem', 1100, 3, 'Negative element
&             Jacobian value %I at element %I. This
&             is due to wrong element order or bad
&             mesh.',dperr(1),' ')
    GOTO 990
END IF
DO iNode = 1, nNodes
    IF (KeyOpt(1).EQ.0) THEN
&         shDerEl(1,iNode) = elJac(1)*shDerIso(1,iNode)
&             + elJac(3)*shDerIso(2,iNode)
&         shDerEl(2,iNode) = elJac(2)*shDerIso(1,iNode)
&             + elJac(4)*shDerIso(2,iNode)
        ELSE
&         shDerEl(1,iNode) = elJac(1)*shDerIso(1,iNode)
&             + elJac(4)*shDerIso(2,iNode)
&             + elJac(7)*shDerIso(3,iNode)
&         shDerEl(2,iNode) = elJac(2)*shDerIso(1,iNode)
&             + elJac(5)*shDerIso(2,iNode)
&             + elJac(8)*shDerIso(3,iNode)
&         shDerEl(3,iNode) = elJac(3)*shDerIso(1,iNode)
&             + elJac(6)*shDerIso(2,iNode)
&             + elJac(9)*shDerIso(3,iNode)
        END IF
    END DO
    dVol = detJac*wtIP(1)

    k1 = 1
    DO iNode = 1,nNodes
        k2 = k1 + 1
        BMat(1,k1) = shDerEl(1,iNode)
        BMat(2,k2) = shDerEl(2,iNode)
        BMat(4,k1) = shDerEl(2,iNode)
        BMat(4,k2) = shDerEl(1,iNode)
        IF (KeyOpt(1).EQ.1) THEN
            k3 = k2 + 1
            BMat(3,k3) = shDerEl(3,iNode)
            BMat(5,k2) = shDerEl(3,iNode)
            BMat(5,k3) = shDerEl(2,iNode)
            BMat(6,k3) = shDerEl(1,iNode)
            BMat(6,k1) = shDerEl(3,iNode)
        END IF
        k1 = k1 + nDim
    END DO

    CALL maxv (BMat(1,1), IncValDofs(1), IncStrain(1), nTens,
&             nUsrDof)

    IF (keyAnsMat.EQ.1) THEN

        CALL ElemGetMat (elId, matId, nDim, nTens, nDirect,
&             intPnt, xCurIP(1), TemperIP,
&             TemperIPB, kThermIP, IncStrain(1),
&             defG0(1,1), defG(1,1),
&             cMat(1,1), MatProp(1), Stress(1),
&             Strain(1), StressTh(1), StrainTh(1),
&             StrainPl(1), StrainCr(1),
&             StressBk(1), StrainSw, EnergyD(1),
&             MatRotGlb(1,1))
        if (kThermIP .eq. 1) kTherm = 1
        density = MatProp(4)
        nu = MatProp(5)
    ELSE

        IF (nlgeom.EQ.0) CALL vmove (IncStrain(1), Strain(1),
&             nTens)
&         Stress(1) = con1*Strain(1) + con2*(Strain(2)+Strain(3))
&         Stress(2) = con1*Strain(2) + con2*(Strain(3)+Strain(1))
&         Stress(3) = con1*Strain(3) + con2*(Strain(1)+Strain(2))

```

```

Stress(4) = G*Strain(4)
IF (KeyOpt(1).EQ.1) THEN
  Stress(5) = G*Strain(5)
  Stress(6) = G*Strain(6)
END IF

kThermIP = 0
call vzero (StressTh(1),nTens)
END IF

IF (keyMtx(1).EQ.1) CALL matba (BMat(1,1), cMat(1,1),
&                               eStiff(1,1), nTens, nTens, nUsrDof,
&                               nTens, nUsrDof, workArr(1), dVol)

  IF (keyMtx(2).EQ.1) THEN
    IF (density.NE.0.0d0) THEN
      workDb = density*dVol
      CALL maat (shIso(1),eMassb(1,1),nNodes,nNodes,workDb)
    ENDIF
  ENDIF

IF (keyMtx(5).EQ.1 .AND. kThermIP.EQ.1 .AND. outkey.EQ.0) THEN
  CALL vmult (StressTh(1), workArr(1), nTens, dVol)
  CALL matxb (BMat(1,1), workArr(1), fExt(1), nTens, nTens,
&             nUsrDof, nUsrDof, 1, -nTens)
END IF

IF (keyMtx(6).EQ.1) THEN
  CALL vmult (Stress(1), workArr(1), nTens, dVol)
  CALL matxb (BMat(1,1), workArr(1), fInt(1), nTens, nTens,
&             nUsrDof, nUsrDof, 1, -nTens)
END IF

elVol = elVol+dVol
elMass = elMass+dVol*density
IF (keyAnsMat.EQ.0) elEnergy(1) = elEnergy(1)
&                               + 0.5d0*dVol*vdot (Strain(1), Stress(1), nTens)
k1 = (intPnt-1)*nTens+1
CALL vmove (Stress(1), saveVars(k1), nTens)
IF (outkey.EQ.1) THEN
  CALL vmove (Strain(1), wStrain(k1), nTens)
  CALL vmove (Stress(1), wStress(k1), nTens)
  IF (keyAnsMat.EQ.1) THEN
    CALL vmove (StrainTh(1), wStrainTh(k1), nTens)
    CALL vmove (StrainPl(1), wStrainPl(k1), nTens)
    CALL vmove (StrainCr(1), wStrainCr(k1), nTens)
  END IF

  IF (debug.EQ.1) THEN
    write (*,3010) intPnt, (Strain(ix),ix=1,nTens)
    write (*,3020) (Stress(ix),ix=1,nTens)
    write (*,3030) (StrainPl(ix),ix=1,nTens)
3010   FORMAT (/1x, 'intPnt=',i2, 'Strain=',6(e15.8,2x))
3020   FORMAT (1x, 8x, 'Stress=',6(e15.8,2x))
3030   FORMAT (1x, 8x, 'StrainPl=',6(e15.8,2x))
  END IF
END IF
100 CONTINUE

IF (keyMtx(1).EQ.1) CALL matsym (eStiff(1,1), nUsrDof, nUsrDof)

IF (keyMtx(2).EQ.1 .AND. density.NE.0.0d0) THEN
  CALL ElemMass (eMassb(1,1), nNodes, nDim, nUsrDof, eMass(1,1))
ENDIF

IF (outkey.EQ.1) THEN
  IF (KeyOpt(1).EQ.0) THEN
    nNodesCorner = nNodes
  ELSE
    nNodesCorner = 8
  END IF
  CALL ElemRsltNode (KeyOpt(1), nrkey, nTens, wStress(1),
&                   wStrain(1), nIntPnts, nStress(1),
&                   nStrain(1), nNodesCorner)

```

```

IF (nRsltBsc.GT.0) THEN
DO iNode = 1, nNodesCorner
  k1 = (iNode-1)*nTens + 1
  k2 = (iNode-1)*7 + 1
  CALL vmove (nStress(k1), RsltBsc(k2), nTens)
  sigm = (nStress(k1)+nStress(k1+1)+nStress(k1+2))/3.0d0
  IF (KeyOpt(1).EQ.0) THEN
    RsltBsc(k2+4) = 0.0d0
    RsltBsc(k2+5) = 0.0d0
    RsltBsc(k2+6) = SQRT(1.5d0*
&      ( nStress(k1)-sigm)*(nStress(k1)-sigm)
&      + (nStress(k1+1)-sigm)*(nStress(k1+1)-sigm)
&      + (nStress(k1+2)-sigm)*(nStress(k1+2)-sigm)
&      + 2.0d0*nStress(k1+3)*nStress(k1+3)))
  ELSE
    RsltBsc(k2+6) = SQRT(1.5d0*
&      ( nStress(k1)-sigm)*(nStress(k1)-sigm)
&      + (nStress(k1+1)-sigm)*(nStress(k1+1)-sigm)
&      + (nStress(k1+2)-sigm)*(nStress(k1+2)-sigm)
&      + 2.0d0*(nStress(k1+3)*nStress(k1+3)
&      + nStress(k1+4)*nStress(k1+4)
&      + nStress(k1+5)*nStress(k1+5)))
  END IF

  k2 = (nNodesCorner+iNode-1)*7 + 1
  CALL vmove (nStrain(k1), RsltBsc(k2), nTens)
  IF (KeyOpt(1).EQ.0) THEN
    RsltBsc(k2+4) = 0.0d0
    RsltBsc(k2+5) = 0.0d0
  END IF
  CALL equivStrain (nu, nStrain(k1), nTens,
&      RsltBsc(k2+6))
&
  END DO
END IF
k1 = nNodesCorner*nTens
CALL vmove (nStrain(1), RsltVar(1), k1)
CALL vmove (nStress(1), RsltVar(k1+1), k1)

IF (elPrint .EQ. 1) THEN
CALL pplock (LOCKOT)
WRITE (iott,2000) elId
2000  FORMAT (/1x, 'Material Point output for element',I8)
WRITE (iott,2100)
2100  FORMAT(/4x, 'Intg.Pt. "S"      Stresses')
DO intPnt = 1, nIntPnts
  k1 = (intPnt-1)*nTens
  WRITE (iott, 2110) intPnt, (wStress(k1+k2),k2=1,nTens)
2110  FORMAT (4x,I4, 4x, 6(E12.5,1x))
END DO
WRITE (iott,2200)
2200  FORMAT(/4x, 'Intg.Pt. "EPTO"     Strains')
DO intPnt = 1, nIntPnts
  k1 = (intPnt-1)*nTens
  WRITE (iott, 2110) intPnt, (wStrain(k1+k2),k2=1,nTens)
END DO
IF (keyAnsMat.EQ.1) THEN
  workDb = 0.0d0
  DO k1 = 1, nIntPnts*nTens
    workDb = workDb + ABS(wStrainPl(k1))
  END DO
  IF (workDb.GT.0.0d0) THEN
    WRITE (iott,2300)
2300  FORMAT(/4x, 'Intg.Pt. "EPPL"     Strains')
    DO intPnt = 1, nIntPnts
      k1 = (intPnt-1)*nTens
      WRITE (iott, 2110) intPnt, (wStrainPl(k1+k2),
&      k2=1,nTens)
&
    END DO
  END IF
  workDb = 0.0d0
  DO k1 = 1, nIntPnts*nTens
    workDb = workDb + ABS(wStrainCr(k1))
  END DO
  IF (workDb.GT.0.0d0) THEN
    WRITE (iott,2400)
2400  FORMAT(/4x, 'Intg.Pt. "EPCR"     Strains')
    DO intPnt = 1, nIntPnts
      k1 = (intPnt-1)*nTens
      WRITE (iott, 2110) intPnt, (wStrainCr(k1+k2),

```



```

&                                k2=1,nTens)
                                END DO
                                END IF
                                workDb = 0.0d0
                                DO k1 = 1, nIntPnts*nTens
                                  workDb = workDb + ABS(wStrainTh(k1))
                                END DO
                                IF (workDb.GT.1.0d-12) THEN
2500                                WRITE (iott,2500)
                                  FORMAT(/4x, 'Intg.Pt. "EPH"      Strains')
                                  DO intPnt = 1, nIntPnts
                                    k1 = (intPnt-1)*nTens
&                                WRITE (iott, 2110) intPnt, (wStrainTh(k1+k2),
&                                k2=1,nTens)
                                  END DO
                                END IF
                                write (iott,3000)
3000                                format(2/)
                                END IF
                                CALL ppunlock (LOCKOT)
                                END IF
                                END IF
990 CONTINUE

RETURN

close(19)
END

```

OFF3D.for – Файл, содержащий процедуры вычисления ортогональных финитных функций

```

if (intPnt.EQ.1) then

ft(1)=(((sqrt(2.)-1.)*(Pnt(1,1)+1.))/2.)+1.
ft(2)=(((sqrt(2.)-1.)*(Pnt(1,2)+1.))/2.)+1.
ft(3)=(((sqrt(2.)-1.)*(Pnt(1,3)+1.))/2.)+1.
ft(4)=(((sqrt(2.)-1.)*(Pnt(1,4)+1.))/2.)+1.
ft(5)=(((sqrt(2.)-1.)*(-1.-Pnt(1,5)))/2.)
ft(6)=(((sqrt(2.)-1.)*(-1.-Pnt(1,6)))/2.)
ft(7)=(((sqrt(2.)-1.)*(-1.-Pnt(1,7)))/2.)
ft(8)=(((sqrt(2.)-1.)*(-1.-Pnt(1,8)))/2.)

fs(1)=(((sqrt(2.)-1.)*(Pnt(2,1)-1.))/2.)+1.
fs(2)=(((sqrt(2.)-1.)*(-1.-Pnt(2,2)))/2.)
fs(3)=(((sqrt(2.)-1.)*(Pnt(2,3)-1.))/2.)+1.
fs(4)=(((sqrt(2.)-1.)*(-1.-Pnt(2,4)))/2.)
fs(5)=(((sqrt(2.)-1.)*(-1.-Pnt(2,5)))/2.)
fs(6)=(((sqrt(2.)-1.)*(Pnt(2,6)-1.))/2.)+1.
fs(7)=(((sqrt(2.)-1.)*(-1.-Pnt(2,7)))/2.)
fs(8)=(((sqrt(2.)-1.)*(Pnt(2,8)-1.))/2.)+1.

fk(1)=(((sqrt(2.)+1.)*(Pnt(3,1)-1.))/2.)+1.
fk(2)=(((sqrt(2.)+1.)*(Pnt(3,2)-1.))/2.)+1.
fk(3)=(((sqrt(2.)+1.)*(1.-Pnt(3,3)))/2.)
fk(4)=(((sqrt(2.)+1.)*(1.-Pnt(3,4)))/2.)
fk(5)=(((sqrt(2.)+1.)*(1.-Pnt(3,5)))/2.)
fk(6)=(((sqrt(2.)+1.)*(1.-Pnt(3,6)))/2.)
fk(7)=(((sqrt(2.)+1.)*(Pnt(3,7)-1.))/2.)+1.
fk(8)=(((sqrt(2.)+1.)*(Pnt(3,8)-1.))/2.)+1.

end if
if (intPnt.EQ.2) then

ft(1)=(((sqrt(2.)-1.)*(Pnt(1,1)+1.))/2.)+1.
ft(2)=(((sqrt(2.)-1.)*(Pnt(1,2)+1.))/2.)+1.
ft(3)=(((sqrt(2.)-1.)*(Pnt(1,3)+1.))/2.)+1.
ft(4)=(((sqrt(2.)-1.)*(Pnt(1,4)+1.))/2.)+1.
ft(5)=(((sqrt(2.)-1.)*(-1.-Pnt(1,5)))/2.)
ft(6)=(((sqrt(2.)-1.)*(-1.-Pnt(1,6)))/2.)
ft(7)=(((sqrt(2.)-1.)*(-1.-Pnt(1,7)))/2.)
ft(8)=(((sqrt(2.)-1.)*(-1.-Pnt(1,8)))/2.)

```

```

fs(1)=(((sqrt(2.)+1.))* (1.-Pnt(2,1)))/2.
fs(2)=(((sqrt(2.)+1.))* (Pnt(2,2)-1.))/2.+1.
fs(3)=(((sqrt(2.)+1.))* (1.-Pnt(2,3)))/2.
fs(4)=(((sqrt(2.)+1.))* (Pnt(2,4)-1.))/2.+1.
fs(5)=(((sqrt(2.)+1.))* (Pnt(2,5)-1.))/2.+1.
fs(6)=(((sqrt(2.)+1.))* (1.-Pnt(2,6)))/2.
fs(7)=(((sqrt(2.)+1.))* (Pnt(2,7)-1.))/2.+1.
fs(8)=(((sqrt(2.)+1.))* (1.-Pnt(2,8)))/2.

fk(1)=(((sqrt(2.)+1.))* (Pnt(3,1)-1.))/2.+1.
fk(2)=(((sqrt(2.)+1.))* (Pnt(3,2)-1.))/2.+1.
fk(3)=(((sqrt(2.)+1.))* (1.-Pnt(3,3)))/2.
fk(4)=(((sqrt(2.)+1.))* (1.-Pnt(3,4)))/2.
fk(5)=(((sqrt(2.)+1.))* (1.-Pnt(3,5)))/2.
fk(6)=(((sqrt(2.)+1.))* (1.-Pnt(3,6)))/2.
fk(7)=(((sqrt(2.)+1.))* (Pnt(3,7)-1.))/2.+1.
fk(8)=(((sqrt(2.)+1.))* (Pnt(3,8)-1.))/2.+1.

end if
if (intPnt.EQ.3) then

ft(1)=(((sqrt(2.)-1.))* (Pnt(1,1)+1.))/2.+1.
ft(2)=(((sqrt(2.)-1.))* (Pnt(1,2)+1.))/2.+1.
ft(3)=(((sqrt(2.)-1.))* (Pnt(1,3)+1.))/2.+1.
ft(4)=(((sqrt(2.)-1.))* (Pnt(1,4)+1.))/2.+1.
ft(5)=(((sqrt(2.)-1.))* (-1.-Pnt(1,5)))/2.
ft(6)=(((sqrt(2.)-1.))* (-1.-Pnt(1,6)))/2.
ft(7)=(((sqrt(2.)-1.))* (-1.-Pnt(1,7)))/2.
ft(8)=(((sqrt(2.)-1.))* (-1.-Pnt(1,8)))/2.

fs(1)=(((sqrt(2.)-1.))* (Pnt(2,1)-1.))/2.+1.
fs(2)=(((sqrt(2.)-1.))* (-1.-Pnt(2,2)))/2.
fs(3)=(((sqrt(2.)-1.))* (Pnt(2,3)-1.))/2.+1.
fs(4)=(((sqrt(2.)-1.))* (-1.-Pnt(2,4)))/2.
fs(5)=(((sqrt(2.)-1.))* (-1.-Pnt(2,5)))/2.
fs(6)=(((sqrt(2.)-1.))* (Pnt(2,6)-1.))/2.+1.
fs(7)=(((sqrt(2.)-1.))* (-1.-Pnt(2,7)))/2.
fs(8)=(((sqrt(2.)-1.))* (Pnt(2,8)-1.))/2.+1.

fk(1)=(((sqrt(2.)-1.))* (-1.-Pnt(3,1)))/2.
fk(2)=(((sqrt(2.)-1.))* (-1.-Pnt(3,2)))/2.
fk(3)=(((sqrt(2.)-1.))* (Pnt(3,3)+1.))/2.+1.
fk(4)=(((sqrt(2.)-1.))* (Pnt(3,4)+1.))/2.+1.
fk(5)=(((sqrt(2.)-1.))* (Pnt(3,5)+1.))/2.+1.
fk(6)=(((sqrt(2.)-1.))* (Pnt(3,6)+1.))/2.+1.
fk(7)=(((sqrt(2.)-1.))* (-1.-Pnt(3,7)))/2.
fk(8)=(((sqrt(2.)-1.))* (-1.-Pnt(3,8)))/2.

end if
if (intPnt.EQ.4) then

ft(1)=(((sqrt(2.)-1.))* (Pnt(1,1)+1.))/2.+1.
ft(2)=(((sqrt(2.)-1.))* (Pnt(1,2)+1.))/2.+1.
ft(3)=(((sqrt(2.)-1.))* (Pnt(1,3)+1.))/2.+1.
ft(4)=(((sqrt(2.)-1.))* (Pnt(1,4)+1.))/2.+1.
ft(5)=(((sqrt(2.)-1.))* (-1.-Pnt(1,5)))/2.
ft(6)=(((sqrt(2.)-1.))* (-1.-Pnt(1,6)))/2.
ft(7)=(((sqrt(2.)-1.))* (-1.-Pnt(1,7)))/2.
ft(8)=(((sqrt(2.)-1.))* (-1.-Pnt(1,8)))/2.

fs(1)=(((sqrt(2.)+1.))* (1.-Pnt(2,1)))/2.
fs(2)=(((sqrt(2.)+1.))* (Pnt(2,2)-1.))/2.+1.
fs(3)=(((sqrt(2.)+1.))* (1.-Pnt(2,3)))/2.
fs(4)=(((sqrt(2.)+1.))* (Pnt(2,4)-1.))/2.+1.
fs(5)=(((sqrt(2.)+1.))* (Pnt(2,5)-1.))/2.+1.
fs(6)=(((sqrt(2.)+1.))* (1.-Pnt(2,6)))/2.
fs(7)=(((sqrt(2.)+1.))* (Pnt(2,7)-1.))/2.+1.
fs(8)=(((sqrt(2.)+1.))* (1.-Pnt(2,8)))/2.

fk(1)=(((sqrt(2.)-1.))* (-1.-Pnt(3,1)))/2.
fk(2)=(((sqrt(2.)-1.))* (-1.-Pnt(3,2)))/2.
fk(3)=(((sqrt(2.)-1.))* (Pnt(3,3)+1.))/2.+1.
fk(4)=(((sqrt(2.)-1.))* (Pnt(3,4)+1.))/2.+1.
fk(5)=(((sqrt(2.)-1.))* (Pnt(3,5)+1.))/2.+1.
fk(6)=(((sqrt(2.)-1.))* (Pnt(3,6)+1.))/2.+1.
fk(7)=(((sqrt(2.)-1.))* (-1.-Pnt(3,7)))/2.
fk(8)=(((sqrt(2.)-1.))* (-1.-Pnt(3,8)))/2.

end if

```

```

if (intPnt.EQ.5) then

    ft(1)=(((sqrt(2.)+1.))* (1-Pnt(1,1)))/2.
    ft(2)=(((sqrt(2.)+1.))* (1-Pnt(1,2)))/2.
    ft(3)=(((sqrt(2.)+1.))* (1-Pnt(1,3)))/2.
    ft(4)=(((sqrt(2.)+1.))* (1-Pnt(1,4)))/2.
    ft(5)=(((sqrt(2.)+1.))* (Pnt(1,5)-1.))/2.+1.
    ft(6)=(((sqrt(2.)+1.))* (Pnt(1,6)-1.))/2.+1.
    ft(7)=(((sqrt(2.)+1.))* (Pnt(1,7)-1.))/2.+1.
    ft(8)=(((sqrt(2.)+1.))* (Pnt(1,8)-1.))/2.+1.

    fs(1)=(((sqrt(2.)+1.))* (1.-Pnt(2,1)))/2.
    fs(2)=(((sqrt(2.)+1.))* (Pnt(2,2)-1.))/2.+1.
    fs(3)=(((sqrt(2.)+1.))* (1.-Pnt(2,3)))/2.
    fs(4)=(((sqrt(2.)+1.))* (Pnt(2,4)-1.))/2.+1.
    fs(5)=(((sqrt(2.)+1.))* (Pnt(2,5)-1.))/2.+1.
    fs(6)=(((sqrt(2.)+1.))* (1.-Pnt(2,6)))/2.
    fs(7)=(((sqrt(2.)+1.))* (Pnt(2,7)-1.))/2.+1.
    fs(8)=(((sqrt(2.)+1.))* (1.-Pnt(2,8)))/2.

    fk(1)=(((sqrt(2.)-1.))* (-1.-Pnt(3,1)))/2.
    fk(2)=(((sqrt(2.)-1.))* (-1.-Pnt(3,2)))/2.
    fk(3)=(((sqrt(2.)-1.))* (Pnt(3,3)+1.))/2.+1.
    fk(4)=(((sqrt(2.)-1.))* (Pnt(3,4)+1.))/2.+1.
    fk(5)=(((sqrt(2.)-1.))* (Pnt(3,5)+1.))/2.+1.
    fk(6)=(((sqrt(2.)-1.))* (Pnt(3,6)+1.))/2.+1.
    fk(7)=(((sqrt(2.)-1.))* (-1.-Pnt(3,7)))/2.
    fk(8)=(((sqrt(2.)-1.))* (-1.-Pnt(3,8)))/2.

end if
if (intPnt.EQ.6) then

    ft(1)=(((sqrt(2.)+1.))* (1-Pnt(1,1)))/2.
    ft(2)=(((sqrt(2.)+1.))* (1-Pnt(1,2)))/2.
    ft(3)=(((sqrt(2.)+1.))* (1-Pnt(1,3)))/2.
    ft(4)=(((sqrt(2.)+1.))* (1-Pnt(1,4)))/2.
    ft(5)=(((sqrt(2.)+1.))* (Pnt(1,5)-1.))/2.+1.
    ft(6)=(((sqrt(2.)+1.))* (Pnt(1,6)-1.))/2.+1.
    ft(7)=(((sqrt(2.)+1.))* (Pnt(1,7)-1.))/2.+1.
    ft(8)=(((sqrt(2.)+1.))* (Pnt(1,8)-1.))/2.+1.

    fs(1)=(((sqrt(2.)-1.))* (Pnt(2,1)-1.))/2.+1.
    fs(2)=(((sqrt(2.)-1.))* (-1.-Pnt(2,2)))/2.
    fs(3)=(((sqrt(2.)-1.))* (Pnt(2,3)-1.))/2.+1.
    fs(4)=(((sqrt(2.)-1.))* (-1.-Pnt(2,4)))/2.
    fs(5)=(((sqrt(2.)-1.))* (-1.-Pnt(2,5)))/2.
    fs(6)=(((sqrt(2.)-1.))* (Pnt(2,6)-1.))/2.+1.
    fs(7)=(((sqrt(2.)-1.))* (-1.-Pnt(2,7)))/2.
    fs(8)=(((sqrt(2.)-1.))* (Pnt(2,8)-1.))/2.+1.

    fk(1)=(((sqrt(2.)-1.))* (-1.-Pnt(3,1)))/2.
    fk(2)=(((sqrt(2.)-1.))* (-1.-Pnt(3,2)))/2.
    fk(3)=(((sqrt(2.)-1.))* (Pnt(3,3)+1.))/2.+1.
    fk(4)=(((sqrt(2.)-1.))* (Pnt(3,4)+1.))/2.+1.
    fk(5)=(((sqrt(2.)-1.))* (Pnt(3,5)+1.))/2.+1.
    fk(6)=(((sqrt(2.)-1.))* (Pnt(3,6)+1.))/2.+1.
    fk(7)=(((sqrt(2.)-1.))* (-1.-Pnt(3,7)))/2.
    fk(8)=(((sqrt(2.)-1.))* (-1.-Pnt(3,8)))/2.

end if
if (intPnt.EQ.7) then

    ft(1)=(((sqrt(2.)+1.))* (1-Pnt(1,1)))/2.
    ft(2)=(((sqrt(2.)+1.))* (1-Pnt(1,2)))/2.
    ft(3)=(((sqrt(2.)+1.))* (1-Pnt(1,3)))/2.
    ft(4)=(((sqrt(2.)+1.))* (1-Pnt(1,4)))/2.
    ft(5)=(((sqrt(2.)+1.))* (Pnt(1,5)-1.))/2.+1.
    ft(6)=(((sqrt(2.)+1.))* (Pnt(1,6)-1.))/2.+1.
    ft(7)=(((sqrt(2.)+1.))* (Pnt(1,7)-1.))/2.+1.
    ft(8)=(((sqrt(2.)+1.))* (Pnt(1,8)-1.))/2.+1.

    fs(1)=(((sqrt(2.)+1.))* (1.-Pnt(2,1)))/2.
    fs(2)=(((sqrt(2.)+1.))* (Pnt(2,2)-1.))/2.+1.
    fs(3)=(((sqrt(2.)+1.))* (1.-Pnt(2,3)))/2.
    fs(4)=(((sqrt(2.)+1.))* (Pnt(2,4)-1.))/2.+1.
    fs(5)=(((sqrt(2.)+1.))* (Pnt(2,5)-1.))/2.+1.
    fs(6)=(((sqrt(2.)+1.))* (1.-Pnt(2,6)))/2.

```

```

fs(7)=(((sqrt(2.)+1.)*(Pnt(2,7)-1.))/2.)+1.
fs(8)=(((sqrt(2.)+1.)*(1.-Pnt(2,8)))/2.

fk(1)=(((sqrt(2.)+1.)*(Pnt(3,1)-1.))/2.)+1.
fk(2)=(((sqrt(2.)+1.)*(Pnt(3,2)-1.))/2.)+1.
fk(3)=(((sqrt(2.)+1.)*(1.-Pnt(3,3)))/2.
fk(4)=(((sqrt(2.)+1.)*(1.-Pnt(3,4)))/2.
fk(5)=(((sqrt(2.)+1.)*(1.-Pnt(3,5)))/2.
fk(6)=(((sqrt(2.)+1.)*(1.-Pnt(3,6)))/2.
fk(7)=(((sqrt(2.)+1.)*(Pnt(3,7)-1.))/2.)+1.
fk(8)=(((sqrt(2.)+1.)*(Pnt(3,8)-1.))/2.)+1.

end if
if (intPnt.EQ.8) then

ft(1)=(((sqrt(2.)+1.)*(1-Pnt(1,1)))/2.
ft(2)=(((sqrt(2.)+1.)*(1-Pnt(1,2)))/2.
ft(3)=(((sqrt(2.)+1.)*(1-Pnt(1,3)))/2.
ft(4)=(((sqrt(2.)+1.)*(1-Pnt(1,4)))/2.
ft(5)=(((sqrt(2.)+1.)*(Pnt(1,5)-1.))/2.)+1.
ft(6)=(((sqrt(2.)+1.)*(Pnt(1,6)-1.))/2.)+1.
ft(7)=(((sqrt(2.)+1.)*(Pnt(1,7)-1.))/2.)+1.
ft(8)=(((sqrt(2.)+1.)*(Pnt(1,8)-1.))/2.)+1.

fs(1)=(((sqrt(2.)-1.)*(Pnt(2,1)-1.))/2.)+1.
fs(2)=(((sqrt(2.)-1.)*(-1.-Pnt(2,2)))/2.
fs(3)=(((sqrt(2.)-1.)*(Pnt(2,3)-1.))/2.)+1.
fs(4)=(((sqrt(2.)-1.)*(-1.-Pnt(2,4)))/2.
fs(5)=(((sqrt(2.)-1.)*(-1.-Pnt(2,5)))/2.
fs(6)=(((sqrt(2.)-1.)*(Pnt(2,6)-1.))/2.)+1.
fs(7)=(((sqrt(2.)-1.)*(-1.-Pnt(2,7)))/2.
fs(8)=(((sqrt(2.)-1.)*(Pnt(2,8)-1.))/2.)+1.

fk(1)=(((sqrt(2.)+1.)*(Pnt(3,1)-1.))/2.)+1.
fk(2)=(((sqrt(2.)+1.)*(Pnt(3,2)-1.))/2.)+1.
fk(3)=(((sqrt(2.)+1.)*(1.-Pnt(3,3)))/2.
fk(4)=(((sqrt(2.)+1.)*(1.-Pnt(3,4)))/2.
fk(5)=(((sqrt(2.)+1.)*(1.-Pnt(3,5)))/2.
fk(6)=(((sqrt(2.)+1.)*(1.-Pnt(3,6)))/2.
fk(7)=(((sqrt(2.)+1.)*(Pnt(3,7)-1.))/2.)+1.
fk(8)=(((sqrt(2.)+1.)*(Pnt(3,8)-1.))/2.)+1.

end if

```

**OFFX3D.for – Файл, содержащий процедуры вычисления
вспомогательных функций формы с ортогональными финитными функциями**

```

if (intPnt.EQ.1) then

Nn(1)=ft(1)*fs(1)*fk(1)
Nn(2)=ft(2)*fs(2)*fk(2)
Nn(3)=ft(3)*fs(3)*fk(3)
Nn(4)=ft(4)*fs(4)*fk(4)
Nn(5)=ft(5)*fs(5)*fk(5)
Nn(6)=ft(6)*fs(6)*fk(6)
Nn(7)=ft(7)*fs(7)*fk(7)
Nn(8)=ft(8)*fs(8)*fk(8)

end if
if (intPnt.EQ.2) then

Nn(1)=ft(1)*fs(1)*fk(1)
Nn(2)=ft(2)*fs(2)*fk(2)
Nn(3)=ft(3)*fs(3)*fk(3)
Nn(4)=ft(4)*fs(4)*fk(4)
Nn(5)=ft(5)*fs(5)*fk(5)
Nn(6)=ft(6)*fs(6)*fk(6)
Nn(7)=ft(7)*fs(7)*fk(7)
Nn(8)=ft(8)*fs(8)*fk(8)

end if
if (intPnt.EQ.4) then

Nn(1)=ft(1)*fs(1)*fk(1)
Nn(2)=ft(2)*fs(2)*fk(2)
Nn(3)=ft(3)*fs(3)*fk(3)
Nn(4)=ft(4)*fs(4)*fk(4)

```

```

Nn(5)=ft(5)*fs(5)*fk(5)
Nn(6)=ft(6)*fs(6)*fk(6)
Nn(7)=ft(7)*fs(7)*fk(7)
Nn(8)=ft(8)*fs(8)*fk(8)

end if
if (intPnt.EQ.6) then

    Nn(1)=ft(1)*fs(1)*fk(1)
    Nn(2)=ft(2)*fs(2)*fk(2)
    Nn(3)=ft(3)*fs(3)*fk(3)
    Nn(4)=ft(4)*fs(4)*fk(4)
    Nn(5)=ft(5)*fs(5)*fk(5)
    Nn(6)=ft(6)*fs(6)*fk(6)
    Nn(7)=ft(7)*fs(7)*fk(7)
    Nn(8)=ft(8)*fs(8)*fk(8)

end if
if (intPnt.EQ.9) then

    Nn(1)=ft(1)*fs(1)*fk(1)
    Nn(2)=ft(2)*fs(2)*fk(2)
    Nn(3)=ft(3)*fs(3)*fk(3)
    Nn(4)=ft(4)*fs(4)*fk(4)
    Nn(5)=ft(5)*fs(5)*fk(5)
    Nn(6)=ft(6)*fs(6)*fk(6)
    Nn(7)=ft(7)*fs(7)*fk(7)
    Nn(8)=ft(8)*fs(8)*fk(8)

end if
if (intPnt.EQ.10) then

    Nn(1)=ft(1)*fs(1)*fk(1)
    Nn(2)=ft(2)*fs(2)*fk(2)
    Nn(3)=ft(3)*fs(3)*fk(3)
    Nn(4)=ft(4)*fs(4)*fk(4)
    Nn(5)=ft(5)*fs(5)*fk(5)
    Nn(6)=ft(6)*fs(6)*fk(6)
    Nn(7)=ft(7)*fs(7)*fk(7)
    Nn(8)=ft(8)*fs(8)*fk(8)

end if
if (intPnt.EQ.12) then

    Nn(1)=ft(1)*fs(1)*fk(1)
    Nn(2)=ft(2)*fs(2)*fk(2)
    Nn(3)=ft(3)*fs(3)*fk(3)
    Nn(4)=ft(4)*fs(4)*fk(4)
    Nn(5)=ft(5)*fs(5)*fk(5)
    Nn(6)=ft(6)*fs(6)*fk(6)
    Nn(7)=ft(7)*fs(7)*fk(7)
    Nn(8)=ft(8)*fs(8)*fk(8)

end if
if (intPnt.EQ.14) then

    Nn(1)=ft(1)*fs(1)*fk(1)
    Nn(2)=ft(2)*fs(2)*fk(2)
    Nn(3)=ft(3)*fs(3)*fk(3)
    Nn(4)=ft(4)*fs(4)*fk(4)
    Nn(5)=ft(5)*fs(5)*fk(5)
    Nn(6)=ft(6)*fs(6)*fk(6)
    Nn(7)=ft(7)*fs(7)*fk(7)
    Nn(8)=ft(8)*fs(8)*fk(8)

end if

```

**dN3D.for – Файл, содержащий процедуры вычисления производных
вспомогательных функций формы**

```

if (intPnt.EQ.1) then

dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.

```

```

dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k)/8.
dN(1,6)=(1.-s)*(1.-k)/8.
dN(1,7)=(1.+s)*(1.+k)/8.
dN(1,8)=(1.-s)*(1.+k)/8.

dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k)/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k)/8.
dN(2,5)=(1.+t)*(1.-k)/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k)/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.

dN(3,1)=(1.-t)*(1.-s)/8.
dN(3,2)=(1.-t)*(1.+s)/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.
dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k)/8.
dN(3,8)=(1.+t)*(1.-s)/8.

end if
if (intPnt.EQ.2) then

dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.
dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k)/8.
dN(1,6)=(1.-s)*(1.-k)/8.
dN(1,7)=(1.+s)*(1.+k)/8.
dN(1,8)=(1.-s)*(1.+k)/8.

dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k)/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k)/8.
dN(2,5)=(1.+t)*(1.-k)/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k)/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.

dN(3,1)=(1.-t)*(1.-s)/8.
dN(3,2)=(1.-t)*(1.+s)/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.
dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k)/8.
dN(3,8)=(1.+t)*(1.-s)/8.

end if
if (intPnt.EQ.3) then

dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.
dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k)/8.
dN(1,6)=(1.-s)*(1.-k)/8.
dN(1,7)=(1.+s)*(1.+k)/8.
dN(1,8)=(1.-s)*(1.+k)/8.

dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k)/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k)/8.
dN(2,5)=(1.+t)*(1.-k)/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k)/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.

dN(3,1)=(1.-t)*(1.-s)/8.
dN(3,2)=(1.-t)*(1.+s)/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.

```

```

dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k)/8.
dN(3,8)=(1.+t)*(1.-s))/8.

end if
if (intPnt.EQ.4) then

dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.
dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k))/8.
dN(1,6)=(1.-s)*(1.-k))/8.
dN(1,7)=(1.+s)*(1.+k))/8.
dN(1,8)=(1.-s)*(1.+k))/8.

dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k))/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k))/8.
dN(2,5)=(1.+t)*(1.-k))/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k))/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.

dN(3,1)=(1.-t)*(1.-s))/8.
dN(3,2)=(1.-t)*(1.+s))/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.
dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k))/8.
dN(3,8)=(1.+t)*(1.-s))/8.

end if

if (intPnt.EQ.5) then

dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.
dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k))/8.
dN(1,6)=(1.-s)*(1.-k))/8.
dN(1,7)=(1.+s)*(1.+k))/8.
dN(1,8)=(1.-s)*(1.+k))/8.

dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k))/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k))/8.
dN(2,5)=(1.+t)*(1.-k))/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k))/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.

dN(3,1)=(1.-t)*(1.-s))/8.
dN(3,2)=(1.-t)*(1.+s))/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.
dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k))/8.
dN(3,8)=(1.+t)*(1.-s))/8.

end if

if (intPnt.EQ.6) then

dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.
dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k))/8.
dN(1,6)=(1.-s)*(1.-k))/8.
dN(1,7)=(1.+s)*(1.+k))/8.

```

```

dN(1,8)=(1.-s)*(1.+k)/8.

dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k)/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k)/8.
dN(2,5)=(1.+t)*(1.-k)/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k)/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.

dN(3,1)=(1.-t)*(1.-s)/8.
dN(3,2)=(1.-t)*(1.+s)/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.
dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k)/8.
dN(3,8)=(1.+t)*(1.-s)/8.

end if

if (intPnt.EQ.7) then

dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.
dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k)/8.
dN(1,6)=(1.-s)*(1.-k)/8.
dN(1,7)=(1.+s)*(1.+k)/8.
dN(1,8)=(1.-s)*(1.+k)/8.

dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k)/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k)/8.
dN(2,5)=(1.+t)*(1.-k)/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k)/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.

dN(3,1)=(1.-t)*(1.-s)/8.
dN(3,2)=(1.-t)*(1.+s)/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.
dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k)/8.
dN(3,8)=(1.+t)*(1.-s)/8.

end if

if (intPnt.EQ.8) then
dN(1,1)=(-1.*(1.-s)*(1.+k))/8.
dN(1,2)=(-1.*(1.+s)*(1.+k))/8.
dN(1,3)=(-1.*(1.-s)*(1.-k))/8.
dN(1,4)=(-1.*(1.+s)*(1.-k))/8.
dN(1,5)=(1.+s)*(1.-k)/8.
dN(1,6)=(1.-s)*(1.-k)/8.
dN(1,7)=(1.+s)*(1.+k)/8.
dN(1,8)=(1.-s)*(1.+k)/8.
dN(2,1)=(-1.*(1.-t)*(1.+k))/8.
dN(2,2)=(1.-t)*(1.+k)/8.
dN(2,3)=(-1.*(1.-t)*(1.-k))/8.
dN(2,4)=(1.-t)*(1.-k)/8.
dN(2,5)=(1.+t)*(1.-k)/8.
dN(2,6)=(-1.*(1.+t)*(1.-k))/8.
dN(2,7)=(1.+t)*(1.+k)/8.
dN(2,8)=(-1.*(1.+t)*(1.+k))/8.
dN(3,1)=(1.-t)*(1.-s)/8.
dN(3,2)=(1.-t)*(1.+s)/8.
dN(3,3)=(-1.*(1.-t)*(1.-s))/8.
dN(3,4)=(-1.*(1.-t)*(1.+s))/8.
dN(3,5)=(-1.*(1.+t)*(1.+s))/8.
dN(3,6)=(-1.*(1.+t)*(1.-s))/8.
dN(3,7)=(1.+t)*(1.+k)/8.
dN(3,8)=(1.+t)*(1.-s)/8.

end if

```



```

&          con1, con2, cMat(nDim*2,nDim*2), shIsoC(nNodes),
&          shIso(nNodes), shDerIso(nDim,nNodes), wtIP(1),
&          workArr(360), elJac(nDim*nDim), detJac, dperr(2),
&          shDerEl(nDim,nNodes), dVol, Strain(nDim*2),
&          Stress(nDim*2), wStrain(48), wStress(48),
&          nStrain(48), nStress(48), sigm, tem, prop(3),
&          IncStrain(nDim*2), defG(3,3), Pnt(2,4),Nn(4),
&          defG0(3,3), xCurIP(nDim), TemperIP, dN(2,4),
&          TemperIPB, StressTh(nDim*2), MatProp(5),
&          StrainPl(nDim*2), StrainCr(nDim*2),
&          StrainTh(nDim*2), StrainSw, StressBk(nDim*2),
&          MatRotGlb(3,3), wStrainTh(48), wStrainPl(48),
&          wStrainCr(48), eMassb(nNodes,nNodes), EnergyD(3),
&          c0,c1,c2,c3,u1,u2,u3,u4,u,uu,d,c(8,8),xc
CHARACTER*4      label(3)

double precision s,t,ft(4),fs(4),ft2,fs2,ftX(4),fsX(4)

INTEGER debug, ix,i,j,k,r

nTens = nDim*2
nComp = nDim*nDim
nDirect = 3
nUsrDof2 = nUsrDof*nUsrDof
CALL vzero (BMat(1,1),nUsrDof*nTens)
IF (keyMtx(1).EQ.1) CALL vzero (eStiff(1,1),nUsrDof2)
IF (keyMtx(2).EQ.1) CALL vzero (eMass(1,1) ,nUsrDof2)
IF (keyMtx(5).EQ.1) CALL vzero (fExt(1) ,nUsrDof)
IF (keyMtx(6).EQ.1) CALL vzero (fInt(1) ,nUsrDof)
IF (nlgeom.EQ.0) THEN
  DO iDim = 1, 3
    DO iDim1 = 1, 3
      defG0(iDim, iDim1) = 0.0D0
    END DO
    defG0(iDim, iDim) = 1.0D0
  END DO
  CALL vmove (defG0(1,1),defG(1,1),9)
ELSE
END IF
elVol = 0.d0
elMass= 0.d0
IF (keyMtx(2).EQ.1) CALL vzero (eMassb(1,1),nNodes*nNodes)
CALL vzero (elEnergy(1), nElEng)

IF (debug.EQ.1) THEN
  write (*,*)
  write (*,*)'elId =',elId,
& ' keyAnsMat=',keyAnsMat,' nDim=',nDim, ' kEStress=',
& kEStress, ' keySym=',keySym, ' nIntPnts=',nIntPnts
END IF

IF (KeyOpt(1).EQ.0) THEN
  CALL ElemShpFn (1, 0, 1, shIsoC(1), nNodes)
  ELSE IF (KeyOpt(1).EQ.1) THEN
  CALL ElemShpFn (2, 0, 1, shIsoC(1), nNodes)
ELSE
  dperr(1) = KeyOpt(1)
  dperr(2) = elId
  CALL erhandler ('UserElem', 1000, 3, 'Invalid input
& KEYOPT(1) = %I at element %I.',dperr(1),
& ' ')
END IF
elCG(3) = 0.0d0
DO iDim = 1, nDim

```

```

        elCG(iDim) = vidot (shIsoC(1), 1, xCur(iDim,1), nDim,
&                nNodes)
    END DO

    IF (keyAnsMat.EQ.1) THEN

    ELSE

        label(1) = 'ex '
        label(2) = 'prxy'
        label(3) = 'dens'
        tem = vdot (shIsoC(1), temper(1), nNodes)

        CALL getMatProp (elId,matId,tem,3,label(1),prop(1))
        Ex = prop(1)
        nu = prop(2)
        density = prop(3)

        IF (nReal.GE.1 .AND. Ex .EQ.0.0d0) Ex = RealConst(1)
        IF (nReal.GE.2 .AND. nu .EQ.0.0d0) nu = RealConst(2)
        IF (nReal.GE.3 .AND. density.EQ.0.0d0) density = RealConst(3)

        G = 0.5d0*Ex/(1.0d0+nu)
        workDb = (1.0d0+nu)*(1.0d0-2.0d0 * nu)
        con1 = (1.0d0-nu)*Ex/workDb
        con2 = nu*Ex/workDb
        CALL vzero (cMat(1,1), nTens*nTens)
        cMat(1,1) = con1
        cMat(2,2) = con1
        cMat(3,3) = con1
        cMat(4,4) = G
        cMat(1,2) = con2
        cMat(1,3) = con2
        cMat(2,3) = con2
        cMat(2,1) = con2
        cMat(3,1) = con2
        cMat(3,2) = con2
        IF (KeyOpt(1).EQ.1) THEN
            cMat(5,5) = G
            cMat(6,6) = G
        END IF
    END IF
    kTherm = 0

    do i=1,nDim
        do j=1,nNodes
            Pnt(i,j) = 0.577
        end do
    end do

    DO 100 intPnt = 1, nIntPnts

    IF (KeyOpt(1).EQ.0) THEN

    -----

        include "OFF.for" !добавленная процедура ортогональные финитные функции
        include "N42LE2.for" !добавленная процедура расчета функций формы на основе

    ОФФ

        do i=1,nNodes
            shIso(i)=Nn(i)
        end do

```

формы

```

-----
      CALL ElemShpFn (1, intPnt, 3, wtIP(1), 1)
    ELSE
      CALL ElemShpFn (2, intPnt, 1, shIso(1), nNodes)
      CALL ElemShpFn (2, intPnt, 2, shDerIso(1,1), nUsrDof)
      CALL ElemShpFn (2, intPnt, 3, wtIP(1), 1)
    END IF

    DO iDim = 1, nDim
      xCurIP(iDim)=vidot(shIso(1),1,xCur(iDim,1),nDim,nNodes)
    END DO
-----

```

```

include "dn42LE2.for" !добавленная процедура расчета производных функции

```

```

do i=1,nDim
  do j=1,nNodes
    shDerIso(i,j)=dN(i,j)
  end do
end do

```

```

-----
CALL vzero (workArr(1), nComp)
iComp = 1
DO iDim = 1, nDim
  DO iDim1 = 1, nDim
    DO iNode = 1, nNodes
      workArr(iComp) = workArr(iComp)
&      + shDerIso(iDim1,iNode)*xCur(iDim,iNode)
    END DO
    iComp = iComp + 1
  END DO
END DO
CALL ElemJac (workArr(1), elJac(1), nDim, detJac,
&      flgSingular)
IF (flgSingular.LE.0) THEN
  dperr(1) = detJac
  dperr(2) = elId
  CALL erhandler ('UserElem', 1100, 3, 'Negative element
&      Jacobian value %I at element %I. This
&      is due to wrong element order or bad
&      mesh.', dperr(1), ' ')
  GOTO 990
END IF
DO iNode = 1, nNodes
  IF (KeyOpt(1).EQ.0) THEN
    shDerEl(1,iNode) = elJac(1)*shDerIso(1,iNode)
&      + elJac(3)*shDerIso(2,iNode)
    shDerEl(2,iNode) = elJac(2)*shDerIso(1,iNode)
&      + elJac(4)*shDerIso(2,iNode)
  ELSE
    shDerEl(1,iNode) = elJac(1)*shDerIso(1,iNode)
&      + elJac(4)*shDerIso(2,iNode)
&      + elJac(7)*shDerIso(3,iNode)
    shDerEl(2,iNode) = elJac(2)*shDerIso(1,iNode)
&      + elJac(5)*shDerIso(2,iNode)
&      + elJac(8)*shDerIso(3,iNode)
    shDerEl(3,iNode) = elJac(3)*shDerIso(1,iNode)
&      + elJac(6)*shDerIso(2,iNode)
&      + elJac(9)*shDerIso(3,iNode)
  END IF
END DO
dVol = detJac*wtIP(1)

```

```

k1 = 1
DO iNode = 1, nNodes
  k2 = k1 + 1
  BMat(1, k1) = shDerEl(1, iNode)
  BMat(2, k2) = shDerEl(2, iNode)
  BMat(4, k1) = shDerEl(2, iNode)
  BMat(4, k2) = shDerEl(1, iNode)
  IF (KeyOpt(1).EQ.1) THEN
    k3 = k2 + 1
    BMat(3, k3) = shDerEl(3, iNode)
    BMat(5, k2) = shDerEl(3, iNode)
    BMat(5, k3) = shDerEl(2, iNode)
    BMat(6, k3) = shDerEl(1, iNode)
    BMat(6, k1) = shDerEl(3, iNode)
  END IF
  k1 = k1 + nDim
END DO

CALL maxv (BMat(1,1), IncValDofs(1), IncStrain(1), nTens,
&          nUsrDof)

IF (keyAnsMat.EQ.1) THEN

  CALL ElemGetMat (eId, matId, nDim, nTens, nDirect,
&                 intPnt, xCurIP(1), TemperIP,
&                 TemperIPB, kThermIP, IncStrain(1),
&                 defG0(1,1), defG(1,1),
&                 cMat(1,1), MatProp(1), Stress(1),
&                 Strain(1), StressTh(1), StrainTh(1),
&                 StrainPl(1), StrainCr(1),
&                 StressBk(1), StrainSw, EnergyD(1),
&                 MatRotGlb(1,1))
  if (kThermIP .eq. 1) kTherm = 1
  density = MatProp(4)
  nu = MatProp(5)
ELSE

  IF (nlgeom.EQ.0) CALL vmove (IncStrain(1), Strain(1),
&                             nTens)
  Stress(1) = con1*Strain(1) + con2*(Strain(2)+Strain(3))
  Stress(2) = con1*Strain(2) + con2*(Strain(3)+Strain(1))
  Stress(3) = con1*Strain(3) + con2*(Strain(1)+Strain(2))
  Stress(4) = G*Strain(4)
  IF (KeyOpt(1).EQ.1) THEN
    Stress(5) = G*Strain(5)
    Stress(6) = G*Strain(6)
  END IF

  kThermIP = 0
  call vzero (StressTh(1), nTens)
END IF

IF (keyMtx(1).EQ.1) CALL matba (BMat(1,1), cMat(1,1),
&                             eStiff(1,1), nTens, nTens, nUsrDof,
&                             nTens, nUsrDof, workArr(1), dVol)

IF (keyMtx(2).EQ.1) THEN
  IF (density.NE.0.0d0) THEN
    workDb = density*dVol
    CALL maat (shIso(1), eMassb(1,1), nNodes, nNodes, workDb)
  ENDIF
ENDIF

IF (keyMtx(5).EQ.1 .AND. kThermIP.EQ.1 .AND. outkey.EQ.0) THEN
  CALL vmult (StressTh(1), workArr(1), nTens, dVol)
  CALL matxb (BMat(1,1), workArr(1), fExt(1), nTens, nTens,
&            nUsrDof, nUsrDof, 1, -nTens)

```

```

END IF

IF (keyMtx(6).EQ.1) THEN
  CALL vmult (Stress(1), workArr(1), nTens, dVol)
  CALL matxb (BMat(1,1), workArr(1), fInt(1), nTens, nTens,
&              nUsrDof, nUsrDof, 1, -nTens)

END IF

elVol = elVol+dVol
elMass = elMass+dVol*density
IF (keyAnsMat.EQ.0) elEnergy(1) = elEnergy(1)
&      + 0.5d0*dVol*vdot(Strain(1), Stress(1), nTens)
k1 = (intPnt-1)*nTens+1
CALL vmove (Stress(1), saveVars(k1), nTens)
IF (outkey.EQ.1) THEN
  CALL vmove (Strain(1), wStrain(k1), nTens)
  CALL vmove (Stress(1), wStress(k1), nTens)
  IF (keyAnsMat.EQ.1) THEN
    CALL vmove (StrainTh(1), wStrainTh(k1), nTens)
    CALL vmove (StrainPl(1), wStrainPl(k1), nTens)
    CALL vmove (StrainCr(1), wStrainCr(k1), nTens)
  END IF

  IF (debug.EQ.1) THEN
    write (*,3010) intPnt, (Strain(ix),ix=1,nTens)
    write (*,3020) (Stress(ix),ix=1,nTens)
    write (*,3030) (StrainPl(ix),ix=1,nTens)
3010    FORMAT (/1x, 'intPnt=',i2, 'Strain=',6(e15.8,2x))
3020    FORMAT (1x, 8x, 'Stress=',6(e15.8,2x))
3030    FORMAT (1x, 8x, 'StrainPl=',6(e15.8,2x))
  END IF
END IF
100 CONTINUE

IF (keyMtx(1).EQ.1) CALL matsym (eStiff(1,1), nUsrDof, nUsrDof)

IF (keyMtx(2).EQ.1 .AND. density.NE.0.0d0) THEN
  CALL ElemMass (eMassb(1,1), nNodes, nDim, nUsrDof, eMass(1,1))
ENDIF

IF (outkey.EQ.1) THEN
  IF (KeyOpt(1).EQ.0) THEN
    nNodesCorner = nNodes
  ELSE
    nNodesCorner = 8
  END IF
  CALL ElemRsltNode (KeyOpt(1), nrkey, nTens, wStress(1),
&                    wStrain(1), nIntPnts, nStress(1),
&                    nStrain(1), nNodesCorner)

  IF (nRsltBsc.GT.0) THEN
    DO iNode = 1, nNodesCorner
      k1 = (iNode-1)*nTens + 1
      k2 = (iNode-1)*7 + 1
      CALL vmove (nStress(k1), RsltBsc(k2), nTens)
      sigm = (nStress(k1)+nStress(k1+1)+nStress(k1+2))/3.0d0
      IF (KeyOpt(1).EQ.0) THEN
        RsltBsc(k2+4) = 0.0d0
        RsltBsc(k2+5) = 0.0d0
        RsltBsc(k2+6) = SQRT(1.5d0*
&          ( nStress(k1)-sigm)*(nStress(k1)-sigm)
&          + (nStress(k1+1)-sigm)*(nStress(k1+1)-sigm)
&          + (nStress(k1+2)-sigm)*(nStress(k1+2)-sigm)
&          + 2.0d0*nStress(k1+3)*nStress(k1+3)))
      ELSE
        RsltBsc(k2+6) = SQRT(1.5d0*

```

```

&          ( nStress(k1)-sigm)*(nStress(k1)-sigm)
&          + (nStress(k1+1)-sigm)*(nStress(k1+1)-sigm)
&          + (nStress(k1+2)-sigm)*(nStress(k1+2)-sigm)
&          + 2.0d0*(nStress(k1+3)*nStress(k1+3)
&          +          nStress(k1+4)*nStress(k1+4)
&          +          nStress(k1+5)*nStress(k1+5)))
      END IF

      k2 = (nNodesCorner+iNode-1)*7 + 1
      CALL vmove (nStrain(k1), RsltBsc(k2), nTens)
      IF (KeyOpt(1).EQ.0) THEN
        RsltBsc(k2+4) = 0.0d0
        RsltBsc(k2+5) = 0.0d0
      END IF
      CALL equivStrain (nu, nStrain(k1), nTens,
&          RsltBsc(k2+6))
&
      END DO
    END IF
    k1 = nNodesCorner*nTens
    CALL vmove (nStrain(1), RsltVar(1), k1)
    CALL vmove (nStress(1), RsltVar(k1+1), k1)

    IF (elPrint .EQ. 1) THEN
      CALL plock (LOCKOT)
      WRITE (iott,2000) elId
2000     FORMAT (/1x, 'Material Point output for element',I8)
      WRITE (iott,2100)
2100     FORMAT(/4x, 'Intg.Pt. "S"      Stresses')
      DO intPnt = 1, nIntPnts
        k1 = (intPnt-1)*nTens
        WRITE (iott, 2110) intPnt, (wStress(k1+k2),k2=1,nTens)
2110     FORMAT (4x,I4, 4x, 6(E12.5,1x))
      END DO
      WRITE (iott,2200)
2200     FORMAT(/4x, 'Intg.Pt. "EPTO"     Strains')
      DO intPnt = 1, nIntPnts
        k1 = (intPnt-1)*nTens
        WRITE (iott, 2110) intPnt, (wStrain(k1+k2),k2=1,nTens)
      END DO
      IF (keyAnsMat.EQ.1) THEN
        workDb = 0.0d0
        DO k1 = 1, nIntPnts*nTens
          workDb = workDb + ABS(wStrainPl(k1))
        END DO
        IF (workDb.GT.0.0d0) THEN
          WRITE (iott,2300)
2300         FORMAT(/4x, 'Intg.Pt. "EPPL"     Strains')
          DO intPnt = 1, nIntPnts
            k1 = (intPnt-1)*nTens
&          WRITE (iott, 2110) intPnt, (wStrainPl(k1+k2),
&          k2=1,nTens)
&
          END DO
        END IF
        workDb = 0.0d0
        DO k1 = 1, nIntPnts*nTens
          workDb = workDb + ABS(wStrainCr(k1))
        END DO
        IF (workDb.GT.0.0d0) THEN
          WRITE (iott,2400)
2400         FORMAT(/4x, 'Intg.Pt. "EPCR"     Strains')
          DO intPnt = 1, nIntPnts
            k1 = (intPnt-1)*nTens
&          WRITE (iott, 2110) intPnt, (wStrainCr(k1+k2),
&          k2=1,nTens)
&
          END DO
        END IF
        workDb = 0.0d0

```

```

DO k1 = 1, nIntPnts*nTens
  workDb = workDb + ABS(wStrainTh(k1))
END DO
IF (workDb.GT.1.0d-12) THEN
  WRITE (iott,2500)
  FORMAT (/4x, 'Intg.Pt. "EPTH"      Strains')
  DO intPnt = 1, nIntPnts
    k1 = (intPnt-1)*nTens
    WRITE (iott, 2110) intPnt, (wStrainTh(k1+k2),
&                                k2=1,nTens)
    END DO
  END IF
  write (iott,3000)
  format(2/)
  END IF
  CALL ppunlock (LOCKOT)
END IF
END IF
990 CONTINUE

RETURN

close(19)
END

```

OFF.for – Вспомогательный файл, содержащий процедуры вычисления ортогональных финитных функций.

```

if (intPnt.EQ.1) then

  ft(1)=(((sqrt(2.)-1.) /2.) * ((-Pnt(1,1))+1.)+1.)
  ft(2)=(((sqrt(2.)+1.) /2.) * (Pnt(1,2)-1.)+1.)
  ft(3)=(((sqrt(2.)+1.) /2.) * (Pnt(1,3)-1.)+1.)
  ft(4)=(((sqrt(2.)-1.) /2.) * (-Pnt(1,4)+1.)+1.)

  fs(1)=(((sqrt(2.)-1.) /2.) * (-Pnt(2,1)+1.)+1.)
  fs(2)=(((sqrt(2.)-1.) /2.) * (-Pnt(2,2)+1.)+1.)
  fs(3)=(((sqrt(2.)+1.) /2.) * ((Pnt(2,3))-1.)+1.)
  fs(4)=(((sqrt(2.)+1.) /2.) * ((Pnt(2,4))-1.)+1.)

end if
if (intPnt.EQ.2) then

  ft(1)=(((sqrt(2.)+1.) /2.) * (Pnt(1,1)-1.)+1.)
  ft(2)=(((sqrt(2.)-1.) /2.) * (-Pnt(1,1)+1.)+1.)
  ft(3)=(((sqrt(2.)-1.) /2.) * (-Pnt(1,1)+1.)+1.)
  ft(4)=(((sqrt(2.)+1.) /2.) * (Pnt(1,1)-1.)+1.)

  fs(1)=(((sqrt(2.)-1.) /2.) * (-Pnt(1,1)+1.)+1.)
  fs(2)=(((sqrt(2.)-1.) /2.) * (-Pnt(1,1)+1.)+1.)
  fs(3)=(((sqrt(2.)+1.) /2.) * ((Pnt(1,1))-1.)+1.)
  fs(4)=(((sqrt(2.)+1.) /2.) * ((Pnt(1,1))-1.)+1.)

end if
if (intPnt.EQ.3) then

  ft(1)=(((sqrt(2.)+1.) /2.) * (Pnt(1,1)-1.)+1.)
  ft(2)=(((sqrt(2.)-1.) /2.) * (-Pnt(1,2)+1.)+1.)
  ft(3)=(((sqrt(2.)-1.) /2.) * (-Pnt(1,3)+1.)+1.)
  ft(4)=(((sqrt(2.)+1.) /2.) * (Pnt(1,4)-1.)+1.)

  fs(1)=(((sqrt(2.)+1.) /2.) * ((Pnt(2,1))-1.)+1.)
  fs(2)=(((sqrt(2.)+1.) /2.) * ((Pnt(2,2))-1.)+1.)

```



```

fs(3)=(((sqrt(2.)-1.)/2.)*(-Pnt(2,3)+1.))+1.)
fs(4)=(((sqrt(2.)-1.)/2.)*(-Pnt(2,4)+1.))+1.)

end if
if (intPnt.EQ.4) then
ft(1)=(((sqrt(2.)-1.)/2.)*(-Pnt(1,1)+1.))+1.)
ft(2)=(((sqrt(2.)+1.)/2.)*(Pnt(1,2)-1.))+1.)
ft(3)=(((sqrt(2.)+1.)/2.)*(Pnt(1,3)-1.))+1.)
ft(4)=(((sqrt(2.)-1.)/2.)*(-Pnt(1,4)+1.))+1.)

fs(1)=(((sqrt(2.)+1.)/2.)*((Pnt(2,1))-1.))+1.)
fs(2)=(((sqrt(2.)+1.)/2.)*((Pnt(2,2))-1.))+1.)
fs(3)=(((sqrt(2.)-1.)/2.)*(-Pnt(2,3)+1.))+1.)
fs(4)=(((sqrt(2.)-1.)/2.)*(-Pnt(2,4)+1.))+1.)

end if

```

N42LE2.for – Файл для вычисления вспомогательных функций формы на основе ОФФ.

```

if (intPnt.EQ.1) then
Nn(1)=ft(1)*fs(1)
Nn(2)=(ft(2)*fs(2))
Nn(3)=ft(3)*fs(3)
Nn(4)=(fs(4)*ft(4))
end if
if (intPnt.EQ.2) then
Nn(1)=ft(1)*fs(1)
Nn(2)=ft(2)*fs(2)
Nn(3)=ft(3)*fs(3)
Nn(4)=ft(4)*fs(4)
end if
if (intPnt.EQ.3) then
Nn(1)=ft(1)*fs(1)
Nn(2)=ft(2)*fs(2)
Nn(3)=ft(3)*fs(3)
Nn(4)=ft(4)*fs(4)
end if
if (intPnt.EQ.4) then
Nn(1)=ft(1)*fs(1)
Nn(2)=ft(2)*fs(2)
Nn(3)=ft(3)*fs(3)
Nn(4)=ft(4)*fs(4)
end if

```

ПРИЛОЖЕНИЕ В

Код программной среды для расчета взаимодействия вращающихся упругих лопастей винта с газами

Class MainWindow

```
'ОСНОВНЫЕ ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ ПРОГРАММНОЙ СРЕДЫ
Public F1 As Integer
Public NameP, IssGeomDown, OkrGeomDown As String
Public NameF2 As String
Public NameFold2 As String
Public cStep, mStep As Integer
Public Fla As Integer
Dim IssCr, OkrCr As Integer

'ПЕРЕМЕННЫЕ ДЛЯ ВНЕДРЕНИЯ В АНСИС
Public WorkingDir, PythonFileName, WBCCommand, PythonWorkingDir As String
Dim PathJob, PathPro As String
Dim PathExe As String
Dim NameJob As String

'ПЕРЕМЕННЫЕ ДЛЯ СОЗДАНИЯ СПЛАЙНА
Private MouseDownPointCollection As New PointCollection
Private ListOfBezierWeights As New ObjectModel.ObservableCollection(Of RationalBezierPoint)

'ПЕРЕМЕННЫЕ ДЛЯ СОЗДАНИЯ МАТЕРИАЛОВ (разбить на подгруппы, чтобы не запутаться)
Public i, j, k1, A, AA, q, qq, B, NumUserLib, NumUserLib2, num As Integer
Public Text, NumR As String
Dim Arr() As Integer
Public NameMat As String
Dim S() As String
Public BoxSel As Integer

Public Y, V, K, G As Double

Public HaveLib = {"General Materials", "General Non-linear Materials", "Explicit Materials", "Hyperelastic
Materials", "Magnetic B-H Curves", "Thermal Materials", "Fluid Materials", "Composite Materials", "-1", "-1", "-1", "-
1"}

Public HaveMat_1 = {"Air", "Aluminum Alloy", "Concrete", "Copper Alloy", "Gray Cast Iron", "Magnesium Alloy",
"Polyethylene", "Silicon Anisotropic", "Stainless Steel", "Structural Steel", "Titanium Alloy"}
Public HaveMat_2 = {"Aluminum Alloy NL", "Concrete NL", "Copper Alloy NL", "Gasket Linear Unloading", "Gasket Non
Linear Unloading", "Magnesium Alloy NL", "Stainless Steel NL", "Structural Steel NL", "Titanium Alloy NL"}
Public HaveMat_3 = {"ADIPRENE", "Air(Atmospheric)", "AL 1100-O", "AL 2024", "AL 2024-T4", "AL 6061-T6", "AL 7039",
"AL 7075-T6", "AL 921-T", "AL/AP HE", "AL2024T351", "AL203-99.5", "AL203-99.7", "AL203 CERA", "AL5083H116",
"ALUMINUM", "ANFO", "ANTIMONY", "BARIUM", "BERYLLIUM", "BERYLLIUM2", "BISMUTH", "BORONCARBI", "BRASS", "BTF", "C4",
"CADMIUM", "CALCIUM", "CART BRASS", "CHROMIUM", "COBALT", "COMP A-3", "COMP B", "CONC140MPA", "CONC-35MPA", "CONCRETE-
L", "COPPER", "COPPER2", "CU OFHC", "CU-OFHC", "CU-OFHC2", "CU-OFHC-F", "CYCLOTOL", "DIPAM", "DU-.75TI", "EL-506A",
"EL-506C",
"EPOXY RES", "EPOXY RES2", "EXPLOS. D", "FEFO", "FLOATGLASB", "FLOATGLASS", "GERMANIUM", "GOLD", "GOLD 5% CU",
"GOLD2", "H-6", "HAFNIUM", "HAFNIUM-2", "HMX", "HMX-INERT", "HMX-TNT", "HNS 1.00", "HNS 1.40", "HNS 1.65",
"INCENDPOWD", "INDIUM", "IRIDIUM", "IRON", "IRON-ARMCO", "IRON-ARMCO2", "IRON-C.E.", "LEAD", "LEAD2", "LEAD3",
"LITHIUM", "LITHIUM F", "LITH-MAGN", "LUCITE", "LX-01", "LX-04-1", "LX-07", "LX-09-1", "LX-10-1", "LX-11", "LX-14-0",
"LX-17-0", "MAG AZ-31B", "MAGNESIUM", "MAGNESIUM2", "MERCURY", "MOLYBDENUM", "MOLYBDENUM2", "NEOPRENE", "NICKEL",
"NICKEL ALL", "NICKEL2", "NICKEL-200", "NICKEL3", "NIOBIUM", "NIOBIUM AL", "NIOBIUM2", "NM",
"NYLON", "OCTOL", "PALLADIUM", "PARAFFIN", "PBX-9010", "PBX-9011", "PBX-9404-3", "PBX-9407", "PBX-9501", "PBX-
9502", "PENTOLITE", "PERICLASE", "PETN 0.88", "PETN 1.26", "PETN 1.50", "PETN 1.77", "PHENOXY", "PLAT 20%IR",
"PLATINUM", "PLATINUM2", "PLEXIGLAS", "POLYCARB", "POLYETHYL.", "POLYRUBBER", "POLYRUBBERH", "POLYSTYREN",
```

```
"POLYURETH", "POLYURETH.", "POTASSIUM", "QUARTZ", "RHA", "RHENIUM", "RHODIUM", "Rubber1", "Rubber2", "RUBIDIUM", "RX-01-AD", "RX-01-AE", "RX-03-BB", "RX-04-DS", "RX-06-AF", "RX-08-AC", "RX-08-BV", "RX-08-DR", "RX-08-DW", "RX-23-AA", "RX-23-AB", "RX-23-AC", "SAND", "SEISMOPLAS", "SiC", "SILVER", "SILVER2", "SIS2541-3", "SOD.CHLOR.", "SODIUM", "SS 21-6-9", "SS 304", "SS-304", "STEEL 1006", "STEEL 4340", "STEEL S-7", "STEEL V250", "STNL.STEEL", "STRONTIUM", "SULFUR", "TANT 10% W", "TANTALUM", "TANTALUM2", "TANTALUM3", "TEFLON", "TEFLONh", "TETRYL", "THALLIUM", "THORIUM", "THORIUM2", "TI 6%AL4%V", "TIN", "TIN2", "TITANIUM", "TITANIUM2", "TITANIUM-2", "TNT", "TNT-2", "TUNG.ALLOY", "TUNGSTEN", "TUNGSTEN2", "TUNGSTEN3", "U 0.75%TI", "U 5%MO", "U 8%NB3%ZR", "U-0.75%TI", "U3 WT % MO", "URANIUM", "URANIUM2", "URANIUM3", "VANADIUM", "VANADIUM2", "W 4%NI2%FE", "WATER", "WATER2", "WATER3", "X-0219", "XTX-8003", "ZINC", "ZIRCONIUM", "ZIRCONIUM2"}
```

```
Public HaveMat_4 = {"Elastomer Sample (Mooney-Rivlin)", "Elastomer Sample (Neo-Hookean)", "Elastomer Sample (Ogden)", "Elastomer Sample (Yeoh)", "Neoprene Rubber"}
```

```
Public HaveMat_5 = {"1018 Steel 90.5 HRB", "Co25ni45", "Cold Rolled low carbon strip steel", "Ferro Cobalt: 34.5% Co", "Gray Cast Iron", "hymu49", "Ingot iron, annealed", "M14 Steel", "M19 Steel", "M2 Steel", "M22 Steel", "M27 Steel", "M3 Steel", "M4 Steel", "M43 Steel", "M47 Steel", "M50 Steel", "M54 Steel", "M6 Steel", "Moly permalloy", "Monel Annealed", "Mu metal", "Nodular Cast Iron", "Powdered iron Sintered plus annealed", "Pure Iron Annealed", "Pure Nickel Annealed", "SA1008", "SA1010", "SA1020", "Silicon Core Iron", "Sintered material 80HP", "SS416", "Supermendure"}
```

```
Public HaveMat_6 = {"Air", "Alnico5", "Alnico9", "Alumina 92%", "Alumina 96%", "Aluminum", "Aluminum Nitride", "Aluminum Oxide", "Bakelite", "Benzocyclobuten", "Beryllium", "Beryllium Oxide", "Brass", "Bronze", "Cast Iron", "Ceramic5", "Ceramic8D", "Chromium", "Cobalt", "Copper", "Diamond", "Diamond High Pressure", "Diamond PL CVD", "Ferrite", "FR-4 Epoxy", "Gallium Arsenide", "Glass", "Gold", "Graphite", "Iron", "Lead", "Magnesium", "Marble", "Mica", "Molybdenum", "Nickel", "Palladium", "Platinum", "Polyethylene", "Polyimide", "Porcelain", "Quartz Glass", "Rubber Hard", "Silicon", "Silicon Dioxide", "Silicon Nitrate", "Silver", "SmCo 24", "SmCo 28", "Solder", "Steel 1008", "Steel 1010", "Steel Stainless", "Tantalum Nitride", "Teflon", "Teflon Based", "Tin", "Titanium", "Tungsten", "Water Fresh", "Zinc"}
```

```
Public HaveMat_7 = {"Acetone", "Acetylene", "Air", "Aluminum", "Aluminum Solid", "Ammonia Liquid", "Ammonia Vapor", "Anthracite", "Argon", "Argon Liquid", "Ash", "Atomic Hydrogen", "Atomic Oxygen", "Benzene Liquid", "Benzene Vapor", "Calcium Carbonate", "Calcium Oxide", "Calcium Sulfate", "Calcium Sulfide", "Carbon", "Carbon Dioxide", "Carbon Disulfide Vapor", "Carbon Monoxide", "Carbon Solid", "Carbonyl Sulfide", "Chlorine", "Chloroform", "Coal Hv", "Coal Lv", "Coal Mv", "Copper", "Diesel Liquid", "Disilane", "Disilicon Dihydride", "Disilicon Trihydride", "Dolomite", "Engine Oil", "Ethane", "Ethyl Acetate", "Ethyl Alcohol Liquid", "Ethyl Alcohol Vapor", "Ethylene", "Ethylene Glycol", "Fluorine", "Freon 12", "Fuel Oil Liquid", "Fuel Oil Vapor", "Gasoil Liquid", "Gasoil Vapor", "Glycerin", "Gold", "Gypsum", "Helium", "Helium Liquid",
```

```
"Hydrazine Liquid", "Hydrogen", "Hydrogen Liquid", "Hydrogen Peroxide", "Hydrogen Sulfide", "Hydroperoxyl", "Hydroxyl", "Ice", "Isobutyl Alcohol", "Kerosene Liquid", "Kerosene Vapor", "Lignite", "Liquid Solid", "Mercury", "Methane", "Methyl Alcohol Liquid", "Methyl Alcohol Vapor", "N Butane", "N Heptane Liquid", "N Heptane Vapor", "N Hexane Liquid", "N Hexane Vapor", "N Octane Liquid", "N Octane Vapor", "N Pentane Liquid", "N Pentane Vapor", "Nickel", "Nitric Acid", "Nitrogen", "Nitrogen Liquid", "Oxygen", "Oxygen Liquid", "Particle Mixture Template", "Peat", "Propane", "Propylene", "Silane", "Silicon Carbide", "Silicon Liquid", "Silicon Solid", "Silylene", "Steel", "Sulfur Dioxide", "Sulfur Liquid", "Sulfur Solid", "Sulfur Vapor", "Titanium", "Toluene Liquid", "Toluene Vapor", "Turpentine", "Urea Solid", "Urea Vapor", "Water Liquid", "Water Vapor",
```

```
"Wood"}
```

```
Public HaveMat_8 = {"Epoxy Carbon UD (230 GPa) Prepreg", "Epoxy Carbon UD (230 GPa) Wet", "Epoxy Carbon UD (395 GPa) Prepreg", "Epoxy Carbon Woven (230 GPa) Prepreg", "Epoxy Carbon Woven (230 GPa) Wet", "Epoxy Carbon Woven (395 GPa) Prepreg", "Epoxy E-Glass UD", "Epoxy E-Glass Wet", "Epoxy S-Glass UD", "Honeycomb", "PVC Foam(60 kg m^-3)", "PVC Foam(80 kg m^-3)", "Resin Epoxy", "Resin Polyester", "SAN Foam(103 kg m^-3)", "SAN Foam(81 kg m^-3)"}
```

```
Public HaveMat_9 = {"-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1"}
```

```
Public HaveMat_10 = {"-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1"}
```

```
Public HaveMat_11 = {"-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1"}
```

```
Public HaveMat_12 = {"-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1", "-1"}
```

```
Public FullMat = {HaveMat_1, HaveMat_2, HaveMat_3, HaveMat_4, HaveMat_5, HaveMat_6, HaveMat_7, HaveMat_8, HaveMat_9, HaveMat_10, HaveMat_11, HaveMat_12}
```

```

Public Tollbox = {"Physical Properties", "Linear Elastic", "Hyperelastic Experimental Data", "Hyperelastic",
"Chaboche Test Data", "Plasticity", "Creep", "Life", "Strength", "Gasket", "Viscoelastic Test Data", "Viscoelastic",
"Shape memory Alloy", "Damage", "Cohesive Zone", "Fracture Criteria", "Thermal", "Thermopower", "Linear 'Soft'
Magnetic Material", "Linear 'Hard' Magnetic Material", "Nonlinear 'Soft' Magnetic Material", "Nonlinear 'Soft'
Magnetic Material", "Electric", "Brittle/Granular", "Equations of State", "Porosity", "Failure", "Nonlinear", "Elasto-
Plastic Behavior"}

```

```

Public Prop_1 = {"Density", "Isotropic Secant Coefficient of Thermal Expansion", "Orthotropic Secant Coefficient
of Thermal Expansion", "Isotropic Instantaneous Coefficient of Thermal Expansion", "Orthotropic Instantaneous
Coefficient of Thermal Expansion", "Constant Damping Coefficient", "Damping Factor (a)", "Damping Factor (b)", "Ply
Type"}

```

```

Public Prop_2 = {"Isotropic Elasticity", "Orthotropic Elasticity", "Anisotropic Elasticity", "Viscoelastic"}

```

```

Public Prop_3 = {"Uniaxial Test Data", "Biaxial Test Data", "Shear Test Data", "Volumetric Test Data", "Simple
Shear Test Data", "Uniaxial Tension Test Data", "Uniaxial Compression Test Data"}

```

```

Public Prop_4 = {"Neo-Hookean", "Arruda-Boyce", "Gent", "Blatz-Ko", "Mooney-Rivlin 2 Parameter", "Mooney-Rivlin 3
Parameter", "Mooney-Rivlin 5 Parameter", "Mooney-Rivlin 9 Parameter", "Polynomial 1st Order", "Polynomial 2nd Order",
"Polynomial 3rd Order", "Yeoh 1st Order", "Yeoh 2nd Order", "Yeoh 3rd Order", "Ogden 1st Order", "Ogden 2nd Order",
"Ogden 3rd Order", "Response Function", "Ogden Foam 1st Order", "Ogden Foam 2nd Order", "Ogden Foam 3rd Order",
"Extended Tube", "Mullins Effect"}

```

```

Public Prop_5 = {"Uniaxial Plastic Strain Test Data"}

```

```

Public Prop_6 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_7 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_8 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_9 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_10 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_11 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_12 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_13 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_14 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_15 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_16 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_17 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_18 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_19 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_20 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_21 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_22 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_23 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_24 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_25 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_26 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_27 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_28 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

Public Prop_29 = {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""}

```

```

'ПЕРЕМЕННЫЕ КЛАССА ДЛЯ СОЗДАНИЯ ТАБЛИЦЫ МАТЕРИАЛОВ

```

```

Dim itemList As New List(Of ListViewItemTemplate)

```

```

Dim itemList2 As New List(Of ListViewItemTemplate2)

```

```

Dim itemList3 As New List(Of ListViewItemTemplate3)

```

```

Dim item As New ListViewItemTemplate

```

```

Dim item2 As New ListViewItemTemplate2

```

```

Dim item3 As New ListViewItemTemplate3

```

```

'КЛАССЫ ДЛЯ РАБОТЫ ПРОГРАММНОЙ СРЕДЫ

```

```

'КЛАСС ДЛЯ МАТЕРИАЛОВ

```

```

Public Class ListViewItemTemplate

```

```

Public Property Library As String
Public Property Material As String
Public Property Number As String
Public Property Safe As Boolean
End Class
Public Class ListViewItemTemplate2
Public Property Library As String
Public Property Material As String
Public Property Safe2 As Boolean
End Class
Public Class ListViewItemTemplate3
Public Property Material As String
Public Property Safe As Boolean
End Class

'КЛАСС ДЛЯ СОЗДАНИЯ СПЛАЙНА
Public Class RationalBezierPoint
Implements ComponentModel.INotifyPropertyChanged

Public Shared ID As Integer = 1

Public Sub reset()
ID = 1
End Sub

Sub New()
PointNumber = ID
ID += 1
End Sub

Private pPointNumber As Integer
Public Property PointNumber() As Integer
Get
Return pPointNumber
End Get
Set(ByVal value As Integer)
pPointNumber = value
PropertyChange("PointNumber")
End Set
End Property

Private pWeight As Double
Public Property Weight() As Double
Get
Return pWeight
End Get
Set(ByVal value As Double)
pWeight = value
PropertyChange("Weight")
End Set
End Property

Private pOverhauserWeight As Double
Public Property OverhauserWeight() As Double
Get
Return pOverhauserWeight
End Get

```

```

    Set (ByVal value As Double)
        pOverhauserWeight = value
        PropertyChanged ("OverhauserWeight")
    End Set
End Property

Public Event PropertyChanged(sender As Object, e As System.ComponentModel.PropertyChangedEventArgs) Implements
System.ComponentModel.INotifyPropertyChanged.PropertyChanged

Private Sub PropertyChanged(ByVal info As String)
    RaiseEvent PropertyChanged(Me, New ComponentModel.PropertyChangedEventArgs (info))
End Sub
End Class

Private Class PointSort
    Implements IComparer
    Public Enum Mode
        X
        Y
    End Enum

    Private currentMode As Mode = Mode.X

    Public Sub New(ByVal mode As Mode)
        currentMode = mode
    End Sub

    Private Function IComparer_Compare(ByVal a As Object, ByVal b As Object) As Integer Implements
IComparer.Compare
        Dim point1 As Point = CType(a, Point)
        Dim point2 As Point = CType(b, Point)

        If currentMode = Mode.X Then

            If point1.X > point2.X Then
                Return 1
            ElseIf point1.X < point2.X Then
                Return -1
            Else
                If point1.Y > point2.Y Then
                    Return 1
                ElseIf point1.Y < point2.Y Then
                    Return -1
                Else
                    Return 0
                End If
            End If
        Else
            If point1.Y > point2.Y Then

                Return 1
            ElseIf point1.Y < point2.Y Then
                Return -1
            Else
                If point1.X > point2.X Then

                    Return 1
                ElseIf point1.X < point2.X Then

```

```

        Return -1
    Else
        Return 0
    End If
End If
End If
End Function
End Class

```

'ЗАДАНИЕ ПЕРЕМЕННЫХ И НАСТРОЕК ВО ВРЕМЯ ЗАГРУЗКИ

```
Private Sub MainWindow_Loaded(sender As Object, e As RoutedEventArgs) Handles Me.Loaded
```

```

    StartBut.Visibility = False
    button.Visibility = True
    button3.Visibility = True
    button4.Visibility = True
    s1.Visibility = True
    s2.Visibility = True
    s3.Visibility = True
    CreatMat.Visibility = True

```

```
FormCreate.Visibility = True
```

```

IssDown.Visibility = True
OkrDown.Visibility = True

```

```
'button.Visibility = False
```

```
'Label2.Text = "Version: 1.45"
```

```
Copyright: efremenkovic@11.06.2016"
```

```

    button.Content = "Выход"
    'button4.Visibility = True
    'button3.Visibility = False
    IssCreate.Visibility = True
    cStep = 1
    mStep = 1

```

```

    button16.Visibility = True
    button17.Visibility = True
    button18.Visibility = True

```

```

CreateLib.Visibility = Visibility.Hidden
NumUserLib = 8
button2001.Visibility = Visibility.Hidden
button3001.Visibility = Visibility.Hidden
CreateMat.Visibility = Visibility.Hidden
CreateMat2.Visibility = Visibility.Hidden
num = 0
IzSv.Visibility = Visibility.Hidden

```

```

t1.Visibility = Visibility.Hidden
t2.Visibility = Visibility.Hidden
t3.Visibility = Visibility.Hidden
t4.Visibility = Visibility.Hidden
t5.Visibility = Visibility.Hidden
t6.Visibility = Visibility.Hidden

```

```
t7.Visibility = Visibility.Hidden  
t8.Visibility = Visibility.Hidden  
t9.Visibility = Visibility.Hidden  
t10.Visibility = Visibility.Hidden  
t11.Visibility = Visibility.Hidden  
t12.Visibility = Visibility.Hidden  
t13.Visibility = Visibility.Hidden  
t14.Visibility = Visibility.Hidden  
t15.Visibility = Visibility.Hidden  
t16.Visibility = Visibility.Hidden  
t17.Visibility = Visibility.Hidden  
t18.Visibility = Visibility.Hidden  
t19.Visibility = Visibility.Hidden  
t20.Visibility = Visibility.Hidden  
t21.Visibility = Visibility.Hidden  
t22.Visibility = Visibility.Hidden  
t23.Visibility = Visibility.Hidden  
t24.Visibility = Visibility.Hidden  
t25.Visibility = Visibility.Hidden
```

```
tt1.Visibility = Visibility.Hidden  
tt2.Visibility = Visibility.Hidden  
tt3.Visibility = Visibility.Hidden  
tt4.Visibility = Visibility.Hidden  
tt5.Visibility = Visibility.Hidden  
tt6.Visibility = Visibility.Hidden  
tt7.Visibility = Visibility.Hidden  
tt8.Visibility = Visibility.Hidden  
tt9.Visibility = Visibility.Hidden  
tt10.Visibility = Visibility.Hidden  
tt11.Visibility = Visibility.Hidden  
tt12.Visibility = Visibility.Hidden  
tt13.Visibility = Visibility.Hidden  
tt14.Visibility = Visibility.Hidden  
tt15.Visibility = Visibility.Hidden  
tt16.Visibility = Visibility.Hidden  
tt17.Visibility = Visibility.Hidden  
tt18.Visibility = Visibility.Hidden  
tt19.Visibility = Visibility.Hidden  
tt20.Visibility = Visibility.Hidden  
tt21.Visibility = Visibility.Hidden  
tt22.Visibility = Visibility.Hidden  
tt23.Visibility = Visibility.Hidden  
tt24.Visibility = Visibility.Hidden
```

```
comboBox.Visibility = Visibility.Hidden
```

'Vvod.Text = "Данный этап необходим для подключения программной среды к вычислительному комплексу Ansys Workbench. Автоматически будет запущен поиск исполнительного файла-привязки, если данный файл был перемещен или отсутствует по стандартному пути, то пользователю необходимо в ручном режиме указать местонахождение этого файла."

'Для создания единого программного комплекса с целью проведения интерактивного моделирования взаимодействия лопастей винта с окружающей средой с выявлением резонанса на их поверхности, для достижения поставленной цели требуется подключение исполняемого файла-привязки: runWB32.exe"

StartText.Text = "Программная среда, ориентированная на проведение расчета динамики взаимодействия вращающихся лопастей винта с окружающей средой, выявление резонанса на лопастях винта (и на методы его устранения)." .

Весь процесс работы программной среды сводится к выполнению последовательных этапов (с возможностью возвращению на предыдущие шаги), которые формируют в программном комплексе Ansys рабочий проект с требуемыми параметрами и модулями на решение определенного типа задач.

Этапы программной среды включают в себя:

1. Внедрение (подключение) программной среды к исполняемому приложению Ansys для проведения расчета. Создание рабочей области, включающей в себя рабочую папку, название проекта и тип задачи.
2. Проверка параметров первого этапа на основе создания тестового проекта с модулями решаемого типа задачи.
3. Создание, или импорт, цифровой модели исследуемого объекта и окружающей области. Этап создания лопастей винта разбит на соответствующие подэтапы проектирования с предложенными вариантами вида лопастей.
4. Создание, или импорт, материалов в библиотеку Ansys требуемых для задачи материалов. Предварительный расчет веса лопасти с указанными материалами.
5. Дискретизация исследуемой модели и окружающей области методом конечных элементов.
6. Назначение граничных и других условий для решаемого типа задачи.
7. Проведение расчета.
8. Анализ полученных результатов."

```
label11.Content = "Этап создания и импорта исследуемой модели и окружающей среды."
```

```
End Sub
```

```
Private Sub button_Click(sender As Object, e As RoutedEventArgs) Handles button.Click
    Close()
End Sub
```

```
End Sub
```

```
Private Sub button1_Click(sender As Object, e As RoutedEventArgs) Handles button1.Click
```

```
    Dim OFD1 As New Microsoft.Win32.OpenFileDialog()
    OFD1.Filter = "Exe files (*.Exe)|*.exe|All Files (*.*)|*.*"
    OFD1.FilterIndex = 1
    OFD1.ShowDialog()
    textBox.Text = OFD1.FileName
    PathExe = OFD1.FileName
End Sub
```

```
End Sub
```

```
Private Sub button2_Click(sender As Object, e As RoutedEventArgs)
```

```
    Dim FBD1 As New System.Windows.Forms.FolderBrowserDialog()
    FBD1.ShowDialog()
    textBox1.Text = FBD1.SelectedPath
    PathJob = textBox1.Text
End Sub
```

```
End Sub
```

```
Private Sub button3_Click(sender As Object, e As RoutedEventArgs) Handles button3.Click
```

```
    cStep += 1
    If (cStep = 2) Then
        button4.Visibility = False
        s1.Visibility = True
        s2.Visibility = False
    ElseIf (cStep = 3) Then
        button3.Visibility = True
        s3.Visibility = False
        s2.Visibility = True
    End If
End Sub
```

```
End Sub
```

```

Private Sub radioButton_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton.Checked
    Fla = 1
    TBL1.Text = "Описание: Расчет эмулирует вращение лопастей винта в открытом пространстве, окружающая среда
представляет собой куб, нижняя грань которого - земля, боковые - открытые грани, через верхнюю - подается воздушный
поток со скоростью выбранную пользователем"
    image.Source = New BitmapImage(New Uri("pack://application:,,,/1.JPG"))
End Sub

Private Sub radioButton1_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton1.Checked
    Fla = 2
    image.Source = New BitmapImage(New Uri("pack://application:,,,/2.JPG"))
    TBL1.Text = "Описание: Вращение винта задается за счет нагоняемого сверху потока"
End Sub

Private Sub radioButton2_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton2.Checked
    Fla = 3
    image.Source = New BitmapImage(New Uri("pack://application:,,,/3.JPG"))
    TBL1.Text = "Описание: Эмулируется расчет в аэродинамической трубе"
End Sub

Private Sub radioButton3_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton3.Checked
    Fla = 4
    image.Source = New BitmapImage(New Uri("pack://application:,,,/4.JPG"))
    TBL1.Text = "Описание: Воздушный поток набегае на вращающийся винт (симуляция полета самолета) (в
разработке)"
End Sub

Private Sub button4_Click(sender As Object, e As RoutedEventArgs) Handles button4.Click
    cStep -= 1
    If (cStep = 2) Then
        button3.Visibility = False
        s2.Visibility = False
        s3.Visibility = True
    ElseIf (cStep = 1) Then
        button4.Visibility = True
        s2.Visibility = True
        s1.Visibility = False
    End If
End Sub

Private Sub button5_Click(sender As Object, e As RoutedEventArgs) Handles button5.Click

    WorkingDir = PathJob
    WorkingDir = WorkingDir + "\"
    PythonFileName = WorkingDir + "Core1.py"
    PythonWorkingDir = Replace(WorkingDir, "\", "/")
    WBCommand = PathExe + " -B -R " + PythonFileName
    NameJob = textBox2.Text

    label12.Content = PathJob + " / " + PathExe + " / " + NameJob

    Dim FilePyt As New IO.StreamWriter(PythonFileName, False)

    FilePyt.WriteLine("# encoding: utf-8")
    FilePyt.WriteLine("SetScriptVersion(Version=""15.0"")")
    FilePyt.WriteLine("")
    FilePyt.WriteLine("workingDir = AbsUserPathName(""" + PythonWorkingDir + """)")
    FilePyt.WriteLine("logfilename = workingDir + ""WB.log"" ")

```

```

FilePvt.WriteLine("paramfilename = workingDir + ""Param.out""")
FilePvt.WriteLine("wbproj = workingDir + "" + NameJob + ".wbpj""")

FilePvt.WriteLine("logFile = open(logfilename, ""w"")")
FilePvt.WriteLine("logFile.write(""Proj Dir: %s\n"" % workingDir)")
FilePvt.WriteLine("Save(")
FilePvt.WriteLine("FilePath= wbproj,")
FilePvt.WriteLine("Overwrite=True)")
FilePvt.WriteLine("logFile.write(""Save: %s\n"" % wbproj)")
FilePvt.WriteLine("logFile.close()")
FilePvt.Close()

Interaction.Shell(WBCommand)
' System.Diagnostics.Process.Start(WBCommand)
End Sub

Private Sub button9_Click(sender As Object, e As RoutedEventArgs) Handles button9.Click
Dim OFD3 As New Microsoft.Win32.OpenFileDialog()
'OFD1.Filter = "Exe files (*.Exe)|*.exe|All Files (*.*)|*.*"
'OFD1.FilterIndex = 1
OFD3.ShowDialog()
textBox4.Text = OFD3.FileName
OkrGeomDown = OFD3.FileName
End Sub

Private Sub button8_Click(sender As Object, e As RoutedEventArgs) Handles button8.Click
If (IssCr = 1) Then
PythonFileName = WorkingDir + "Core4.py"
PythonWorkingDir = Replace(WorkingDir, "\", "/")
WBCommand = PathExe + " -B -R " + PythonFileName
PathPro = PythonWorkingDir + NameJob
IssGeomDown = Replace(IssGeomDown, "\", "/")

Dim FilePvt As New IO.StreamWriter(PythonFileName, False)

FilePvt.WriteLine("# encoding: utf-8")
FilePvt.WriteLine("SetScriptVersion(Version=""15.0"")")
FilePvt.WriteLine("")
FilePvt.WriteLine("Open(FilePath="" + PathPro + ".wbpj"")")
FilePvt.WriteLine("")

FilePvt.WriteLine("template3 = GetTemplate(TemplateName = ""Geometry"")")
FilePvt.WriteLine("system3 = template3.CreateSystem()")
FilePvt.WriteLine("template4 = GetTemplate(TemplateName = ""Mesh"")")
FilePvt.WriteLine("system4 = template4.CreateSystem()")
FilePvt.WriteLine("geometry2 = system3.GetContainer(ComponentName = ""Geometry"")")
FilePvt.WriteLine("geometry2.SetFile(FilePath = "" + IssGeomDown + """)")
FilePvt.WriteLine("geometry2.Edit()")
FilePvt.WriteLine("geometry2.SendCommand(Command=""agb.Regen();"")")
FilePvt.WriteLine("geometry2.Exit()")
FilePvt.WriteLine("component4 = system3.GetComponent(Name = ""Geometry"")")
FilePvt.WriteLine("component5 = system4.GetComponent(Name = ""Geometry"")")
FilePvt.WriteLine("component5.ReplaceWithShare(TargetSystem = system4, ComponentToShare = component4,
SourceSystem = system3)")
FilePvt.WriteLine("")
FilePvt.WriteLine("Save(Overwrite = True)")
FilePvt.Close()

```

```

Interaction.Shell(WBCommand)

ElseIf (IssCr = 2) Then
    FormCreate.Visibility = True
    IssCreate.Visibility = False
Else
End If
End Sub

Private Sub button6_Click(sender As Object, e As RoutedEventArgs) Handles button6.Click
    'Visibility = True
    s1.Visibility = True
    s2.Visibility = True
    s3.Visibility = True
    button3.Visibility = True
    button4.Visibility = True
    button.Visibility = True
    Width = 1200
    Height = 700
    FormCreate.Visibility = False
    button16.Visibility = False
    button17.Visibility = False
    button18.Visibility = True
End Sub

Private Sub button7_Click(sender As Object, e As RoutedEventArgs) Handles button7.Click
    Dim OFD2 As New Microsoft.Win32.OpenFileDialog()
    'OFD1.Filter = "Exe files (*.Exe)|*.exe|All Files (*.*)|*.*"
    'OFD1.FilterIndex = 1
    OFD2.ShowDialog()
    textBox3.Text = OFD2.FileName
    IssGeomDown = OFD2.FileName
    If (IssGeomDown <> " " Or IssGeomDown <> Nothing) Then
        IssCr = 1
    End If
End Sub

Private Sub radioButton4_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton4.Checked
    IssDown.Visibility = False
    textBlockIss.Text = "Загрузить модель"
End Sub

Private Sub radioButton5_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton5.Checked
    IssDown.Visibility = True
    IssCr = 2
    textBlockIss.Text = "Создать модель"
End Sub

Private Sub radioButton6_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton6.Checked
    OkrDown.Visibility = False
    textBlock2.Text = "Загрузить модель"
End Sub

Private Sub radioButton7_Checked(sender As Object, e As RoutedEventArgs) Handles radioButton7.Checked
    OkrDown.Visibility = True
    textBlock2.Text = "Создать модель"
    OkrCr = 2

```

End Sub

'Разобрать!!'

Private Sub cnvMain_MouseDown(sender As System.Object, e As System.Windows.Input.MouseButtonEventArgs)

Dim p As New Point

p = e.MouseDevice.GetPosition(cnvMain)

Dim r, r2 As New RationalBezierPoint

r.Weight = 1

r.OverhauserWeight = ListOfBezierWeights.Count + 1

ListOfBezierWeights.Add(r)

MouseDownPointCollection.Add(p)

Dim radius As Double

radius = 2.5

Dim c As New Ellipse

c.Fill = Brushes.Red

c.Width = radius * 2

c.Height = radius * 2

Canvas.SetLeft(c, p.X - c.Height / 2)

Canvas.SetTop(c, p.Y - c.Width / 2)

cnvMain.Children.Add(c)

End Sub

Private Sub btnClearAll_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)

cnvMain.Children.Clear()

MouseDownPointCollection.Clear()

If ListOfBezierWeights.Count <> 0 Then ListOfBezierWeights(0).reset()

ListOfBezierWeights.Clear()

End Sub

Private Sub tbnDrawDerivateOfBezierSegment_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)

GenerateSlopes(BezierFunction(MouseDownPointCollection, 0.3), DerivateBezierFunction(MouseDownPointCollection, 0.3))

End Sub

Private Sub button12_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)

ClearAllLines()

Dim W(ListOfBezierWeights.Count - 1) As Double

'For i As Integer = 0 To ListOfBezierWeights.Count - 1

'W(i) = ListOfBezierWeights(i).OverhauserWeight

'Next

PlotPoints(CatmullRomSpline(MouseDownPointCollection, 0.01, False))

End Sub

Private Sub btnDrawBesselOverhauserSpline_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)

ClearAllLines()

```

Dim W(ListOfBezierWeights.Count - 1) As Double
For i As Integer = 0 To ListOfBezierWeights.Count - 1
    W(i) = ListOfBezierWeights(i).OverhauserWeight
Next
PlotPoints(BesselOverhauserSpline(MouseDownPointCollection, W, 0.01, False))

End Sub

Private Sub btnDrawLagrangeInterpolation_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)
    ClearAllLines()
    PlotPoints(LagrangianInterpolation(MouseDownPointCollection))
End Sub

Private Sub btnCalculateConvexHull_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)
    ClearAllLines()
    PlotPoints(ConvexHull2D(MouseDownPointCollection))
End Sub

Private Sub PlotPoints(ByVal p As PointCollection)
    For i As Integer = 0 To p.Count - 2
        Dim ll As New Line
        ll.Stroke = Brushes.Black
        ll.StrokeThickness = 2
        ll.X1 = p(i).X
        ll.Y1 = p(i).Y
        ll.X2 = p(i + 1).X
        ll.Y2 = p(i + 1).Y
        cnvMain.Children.Add(ll)
    Next
End Sub

Private Sub ClearAllLines()

    For i As Integer = cnvMain.Children.Count - 1 To 0 Step -1
        If TypeOf (cnvMain.Children(i)) Is Line Then
            cnvMain.Children.RemoveAt(i)
        End If
    Next
End Sub

Private Sub RemoveImage()

    For i As Integer = cnvMain.Children.Count - 1 To 0 Step -1
        If TypeOf (cnvMain.Children(i)) Is Image Then
            cnvMain.Children.RemoveAt(i)
        End If
    Next
End Sub

Private Sub GenerateSlopes(ByVal p As PointCollection, ByVal pp As PointCollection)
    For i As Integer = 0 To pp.Count - 1

        Dim d As Double
        d = pp(i).Y / pp(i).X

        Dim ll As New Line
        ll.Stroke = Brushes.Black
        ll.StrokeThickness = 2

```

```

    ll.X1 = p(i).X
    ll.Y1 = p(i).Y
    ll.X2 = p(i).X + 10
    ll.Y2 = p(i).Y + 10 * d
    cnvMain.Children.Add(ll)
Next

End Sub

Private Function AngleInDegree(ByVal start As Point, ByVal endd As Point) As Double
    Return Math.Atan2(start.Y - endd.Y, endd.X - start.X) * 180 / Math.PI
End Function

Private Function AngleInRadianer(ByVal start As Point, ByVal endd As Point) As Double
    Return Math.Atan2(start.Y - endd.Y, endd.X - start.X) * 180 / Math.PI
End Function

Private Function MovePoint(ByVal StartPoint As Point, ByVal Length As Double, Optional ByVal theta As Double = 90)
As Point
    Dim EndPoint As New Point

    theta *= Math.PI / 180

    EndPoint.X = Math.Cos(theta) * (Length) - Math.Sin(theta) * (Length) + StartPoint.X
    EndPoint.Y = Math.Sin(theta) * (Length) + Math.Cos(theta) * (Length) + StartPoint.Y
    Return EndPoint
End Function

Private Function GetEucledianDistance(ByVal p1 As Point, ByVal p2 As Point) As Double
    Return Math.Sqrt((p1.X - p2.X) ^ 2 + (p1.Y - p2.Y) ^ 2)
End Function

Private Function BezierFunction(ByVal p As PointCollection, ByVal StepSize As Double) As PointCollection
    Dim result As New PointCollection
    Dim B As Double()

    For k As Double = 0 To 1 Step StepSize

        B = AllBernstein(p.Count, k)

        Dim CX, CY As Double
        CX = 0
        CY = 0
        For j As Integer = 0 To p.Count - 1
            CX = CX + B(j) * p(j).X
            CY = CY + B(j) * p(j).Y
        Next
        result.Add(New Point(CX, CY))
    Next

    result.Add(p(p.Count - 1))

    Return result
End Function

Private Function AllBernstein(ByVal n As Integer, ByVal u As Double) As Double()

```

```

Dim B(n - 1) As Double
B(0) = 1
Dim u1 As Double
u1 = 1 - u
Dim saved, temp As Double
For j As Integer = 1 To n - 1

    saved = 0
    For k As Integer = 0 To j - 1
        temp = B(k)
        B(k) = saved + u1 * temp
        saved = u * temp
    Next
    B(j) = saved
Next

Return B
End Function

Private Function DerivateBezierFunction(ByVal p As PointCollection, ByVal StepSize As Double) As PointCollection
    Dim result As New PointCollection
    Dim B As Double()
    Dim CX, CY As Double
    For k As Double = 0 To 1 Step StepSize

        B = AllDerivateBernstein(p.Count, k)

        CX = 0
        CY = 0
        For j As Integer = 0 To p.Count - 1
            CX = CX + B(j) * p(j).X
            CY = CY + B(j) * p(j).Y
        Next
        result.Add(New Point(CX, CY))
    Next

    B = AllDerivateBernstein(p.Count, 1)

    CX = 0
    CY = 0
    For j As Integer = 0 To p.Count - 1
        CX = CX + B(j) * p(j).X
        CY = CY + B(j) * p(j).Y
    Next
    result.Add(New Point(CX, CY))

    Return result
End Function

Private Function PointBezierFunction(ByVal p As PointCollection, ByVal StepSize As Double) As PointCollection
    Dim result As New PointCollection
    Dim B As Double()
    Dim CX, CY As Double
    Dim k As Double = StepSize

    B = AllBernstein(p.Count, k)

    CX = 0

```



```

CY = 0
For j As Integer = 0 To p.Count - 1
    CX = CX + B(j) * p(j).X
    CY = CY + B(j) * p(j).Y
Next
Return New Point(CX, CY)

End Function

Private Function GetDerivativeAtPoint(ByVal k As Double, ByVal p As PointCollection) As Line
    Dim result As New PointCollection
    Dim B As Double()
    Dim CX, CY As Double

    B = AllBernstein(p.Count, k)

    CX = 0
    CY = 0
    For j As Integer = 0 To p.Count - 1
        CX = CX + B(j) * p(j).X
        CY = CY + B(j) * p(j).Y
    Next
    result.Add(New Point(CX, CY))

    B = AllDerivateBernstein(p.Count, k)

    CX = 0
    CY = 0
    For j As Integer = 0 To p.Count - 1
        CX = CX + B(j) * p(j).X
        CY = CY + B(j) * p(j).Y
    Next

    Dim d As Double
    d = CY / CX

    Dim l1 As New Line
    l1.Stroke = Brushes.DarkGreen
    l1.StrokeThickness = 2
    l1.X1 = result(0).X
    l1.Y1 = result(0).Y
    l1.X2 = result(0).X + 10
    l1.Y2 = result(0).Y + 10 * d

    Dim p1, p2 As New Point
    p1.X = result(0).X
    p1.Y = result(0).Y
    p2.X = result(0).X + 10
    p2.Y = result(0).Y + 10 * d

    Return l1
End Function

Private Function GetDerivativeAngleAtPoint(ByVal k As Double, ByVal p As PointCollection) As Double
    Dim result As New PointCollection
    Dim B As Double()

```

```

Dim CX, CY As Double

B = AllBernstein(p.Count, k)

CX = 0
CY = 0
For j As Integer = 0 To p.Count - 1
    CX = CX + B(j) * p(j).X
    CY = CY + B(j) * p(j).Y
Next
result.Add(New Point(CX, CY))

B = AllDerivateBernstein(p.Count, k)

CX = 0
CY = 0
For j As Integer = 0 To p.Count - 1
    CX = CX + B(j) * p(j).X
    CY = CY + B(j) * p(j).Y
Next

Dim d As Double
d = CY / CX

Dim p1, p2 As New Point
p1.X = result(0).X
p1.Y = result(0).Y
p2.X = result(0).X + 10
p2.Y = result(0).Y + 10 * d

Return AngleInDegree(p1, p2)
End Function

Private Function RationalBezierFunction(ByVal p As PointCollection, ByVal Weight() As Double, ByVal StepSize As
Double) As PointCollection
    Dim result As New PointCollection
    Dim B As Double()

    For k As Double = 0 To 1 Step StepSize

        B = RationalBasisFunction(p.Count, k, Weight)

        Dim CX, CY As Double
        CX = 0
        CY = 0
        For j As Integer = 0 To p.Count - 1
            CX = CX + B(j) * p(j).X
            CY = CY + B(j) * p(j).Y
        Next
        result.Add(New Point(CX, CY))
    Next

    result.Add(p(p.Count - 1))

    Return result
End Function

Private Function AllDerivateBernstein(ByVal n As Integer, ByVal u As Double) As Double()

```

```

Dim B As Double()
B = AllBernstein(n - 1, u)

Dim result(n - 1) As Double

For i As Integer = 0 To n - 1
    If i = 0 Then
        result(i) = n * (-B(i))
    ElseIf i = n - 1 Then
        result(i) = n * (B(i - 1))
    Else
        result(i) = n * (B(i - 1) - B(i))
    End If
Next

Return result
End Function

Private Function RationalBasisFunction(ByVal n As Integer, ByVal u As Double, ByVal weight() As Double) As Double()
    If weight.Length <> n Then Return Nothing

    Dim B() As Double
    B = AllBernstein(n, u)

    Dim result(n - 1) As Double

    Dim test As Double
    test = 0
    For j As Integer = 0 To n - 1
        test += B(j) * weight(j)
    Next

    For i As Integer = 0 To n - 1
        result(i) = B(i) * weight(i) / test
    Next

    Return result
End Function

Public Function PointOnCatmullRomCurve(ByVal p0 As System.Windows.Point, ByVal p1 As System.Windows.Point, ByVal p2 As System.Windows.Point, ByVal p3 As System.Windows.Point, ByVal t As Double) As System.Windows.Point
    Dim result As New System.Windows.Point()

    Dim lix1, liy1, lix2, liy2 As Double
    lix1 = 0.5 * (p2.X - p0.X)
    lix2 = 0.5 * (p3.X - p1.X)

    liy1 = 0.5 * (p2.Y - p0.Y)
    liy2 = 0.5 * (p3.Y - p1.Y)

    Dim PointList As New PointCollection
    PointList.Add(p1)
    PointList.Add(New Point(p1.X + (1 / 3) * lix1, p1.Y + (1 / 3) * liy1))
    PointList.Add(New Point(p2.X - (1 / 3) * lix2, p2.Y - (1 / 3) * liy2))

```

```

PointList.Add(p2)

Return PointBezierFunction(PointList, t)
End Function

Public Function CatmullRomSpline(ByVal Points As PointCollection, Optional ByVal InterpolationStep As Double =
0.1, Optional ByVal IsPolygon As Boolean = False) As PointCollection
    Dim result As New PointCollection

    If Points.Count <= 2 Then
        Return Points
    End If

    If IsPolygon Then
        For i As Integer = 0 To Points.Count - 1
            If i = 0 Then
                For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
                    result.Add(PointOnCatmullRomCurve(Points(Points.Count - 1), Points(i), Points(i + 1), Points(i
+ 2), k))

                Next
            ElseIf i = Points.Count - 1 Then
                For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
                    result.Add(PointOnCatmullRomCurve(Points(i - 1), Points(i), Points(0), Points(1), k))
                Next
            ElseIf i = Points.Count - 2 Then
                For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
                    result.Add(PointOnCatmullRomCurve(Points(i - 1), Points(i), Points(i + 1), Points(0), k))
                Next
            Else
                For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
                    result.Add(PointOnCatmullRomCurve(Points(i - 1), Points(i), Points(i + 1), Points(i + 2), k))
                Next
            End If
        Next
    Else
        Dim yarray, xarray As New List(Of Double)
        xarray.Add(Points(0).X - (Points(1).X - Points(0).X) / 2)
        yarray.Add(Points(0).Y - (Points(1).Y - Points(0).Y) / 2)

        For Each ps As System.Windows.Point In Points
            xarray.Add(ps.X)
            yarray.Add(ps.Y)
        Next

        xarray.Add((Points(Points.Count - 1).X - (Points(Points.Count - 2).X) / 2 + Points(Points.Count - 1).X))
        yarray.Add((Points(Points.Count - 1).Y - (Points(Points.Count - 2).Y) / 2 + Points(Points.Count - 1).Y))

        Dim r As New PointCollection
        For i As Integer = 0 To yarray.Count - 1
            r.Add(New System.Windows.Point(xarray(i), yarray(i)))
        Next

        For i As Integer = 3 To r.Count - 1
            For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
                result.Add(PointOnCatmullRomCurve(r(i - 3), r(i - 2), r(i - 1), r(i), k))
            Next
        Next
    End If
End Function

```

```

        result.Add(Points(Points.Count - 1))
    End If

    Return result
End Function

Private Sub button17_Click(sender As Object, e As RoutedEventArgs) Handles button17.Click
    mStep += 1
    If (mStep = 2) Then
        button18.Visibility = False
        FormCreate.Visibility = True
        CreatMat.Visibility = False
    ElseIf (mStep = 3) Then
        button17.Visibility = True
        'MeshCreate.Visibility = False
        CreatMat.Visibility = True
    End If
End Sub

Private Sub button18_Click(sender As Object, e As RoutedEventArgs) Handles button18.Click
    mStep -= 1
    If (mStep = 2) Then
        button16.Visibility = False
        CreatMat.Visibility = False
        'MeshCreate.Visibility = True
    ElseIf (mStep = 1) Then
        button17.Visibility = True
        CreatMat.Visibility = True
        FormCreate.Visibility = False
    End If
End Sub

Private Sub button16_Click(sender As Object, e As RoutedEventArgs) Handles button16.Click
    Close()
End Sub

Public Function BesselOverhauserSpline(ByVal Points As PointCollection, ByVal t() As Double, Optional ByVal
InterpolationStep As Double = 0.1, Optional ByVal IsPolygon As Boolean = False) As PointCollection
    Dim result As New PointCollection

    If Points.Count <= 2 Then
        Return Points
    End If

    If IsPolygon Then
        For i As Integer = 0 To Points.Count - 1
            If i = 0 Then
                For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep

                    result.Add(PointOnBesselOverhauserCurve(Points(Points.Count - 1), Points(i), Points(i + 1),
Points(i + 2), k))

                Next
            ElseIf i = Points.Count - 1 Then
                For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
                    result.Add(PointOnBesselOverhauserCurve(Points(i - 1), Points(i), Points(0), Points(1), k))
                Next
            ElseIf i = Points.Count - 2 Then

```

```

        For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
            result.Add(PointOnBesselOverhauserCurve(Points(i - 1), Points(i), Points(i + 1), Points(0),
k))

        Next
    Else
        For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
            result.Add(PointOnBesselOverhauserCurve(Points(i - 1), Points(i), Points(i + 1), Points(i +
2), k))

        Next
    End If
Next
Else
    Dim yarray, xarray As New List(Of Double)
    xarray.Add(Points(0).X - (Points(1).X - Points(0).X) / 2)
    yarray.Add(Points(0).Y - (Points(1).Y - Points(0).Y) / 2)

    Dim rt As New List(Of Double)
    rt.Add(-(t(1) - t(0)) / 2)
    Dim kk As Integer = 0
    For Each ps As System.Windows.Point In Points
        xarray.Add(ps.X)
        yarray.Add(ps.Y)
        rt.Add(t(kk))
        kk += 1
    Next

    xarray.Add((Points(Points.Count - 1).X - (Points(Points.Count - 2).X) / 2 + Points(Points.Count - 1).X))
    yarray.Add((Points(Points.Count - 1).Y - (Points(Points.Count - 2).Y) / 2 + Points(Points.Count - 1).Y))
    rt.Add(t(Points.Count - 1) + (t(Points.Count - 1) - t(Points.Count - 2)) / 2)

    Dim r As New PointCollection
    For i As Integer = 0 To yarray.Count - 1
        r.Add(New System.Windows.Point(xarray(i), yarray(i)))
    Next

    For i As Integer = 3 To r.Count - 1
        For k As Double = 0 To (1 - InterpolationStep) Step InterpolationStep
            Dim tt As New List(Of Double)
            tt.Add(rt(i - 3))
            tt.Add(rt(i - 2))
            tt.Add(rt(i - 1))
            tt.Add(rt(i))

            result.Add(PointOnBesselOverhauserCurve(r(i - 3), r(i - 2), r(i - 1), r(i), k, tt.ToArray))
        Next
    Next
    result.Add(Points(Points.Count - 1))
End If

Return result
End Function

Public Function PointOnBesselOverhauserCurve(ByVal p0 As System.Windows.Point, ByVal p1 As System.Windows.Point,
ByVal p2 As System.Windows.Point, ByVal p3 As System.Windows.Point, ByVal u As Double, Optional ByVal t() As Double =
Nothing) As System.Windows.Point
    Dim result As New System.Windows.Point()

```

```

If t Is Nothing Then

    t = {1, 2, 3, 4}
End If

Dim ViXPlusHalf, ViXMinusHalf, ViYPlusHalf, ViYMinusHalf, ViX, ViY As Double
ViXPlusHalf = (p2.X - p1.X) / (t(2) - t(1))
ViXMinusHalf = (p1.X - p0.X) / (t(1) - t(0))
ViYPlusHalf = (p2.Y - p1.Y) / (t(2) - t(1))
ViYMinusHalf = (p1.Y - p0.Y) / (t(1) - t(0))

ViX = ((t(2) - t(1)) * ViXMinusHalf + (t(1) - t(0)) * ViXPlusHalf) / (t(2) - t(0))
ViY = ((t(2) - t(1)) * ViYMinusHalf + (t(1) - t(0)) * ViYPlusHalf) / (t(2) - t(0))

Dim PointList As New PointCollection
PointList.Add(p1)
PointList.Add(New Point(p1.X + (1 / 3) * (t(2) - t(1)) * ViX, p1.Y + (1 / 3) * (t(2) - t(1)) * ViY))

ViXPlusHalf = (p3.X - p2.X) / (t(3) - t(2))
ViXMinusHalf = (p2.X - p1.X) / (t(2) - t(1))
ViYPlusHalf = (p3.Y - p2.Y) / (t(3) - t(2))
ViYMinusHalf = (p2.Y - p1.Y) / (t(2) - t(1))

ViX = ((t(3) - t(2)) * ViXMinusHalf + (t(2) - t(1)) * ViXPlusHalf) / (t(3) - t(1))
ViY = ((t(3) - t(2)) * ViYMinusHalf + (t(2) - t(1)) * ViYPlusHalf) / (t(3) - t(1))

PointList.Add(New Point(p2.X - (1 / 3) * (t(3) - t(2)) * ViX, p2.Y - (1 / 3) * (t(3) - t(2)) * ViY))
PointList.Add(p2)

Return PointBezierFunction(PointList, u)
End Function

Private Sub btnDrawBSplineCurve_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)
    Dim KN() As Double = {0, 0, 0, 0, 1, 1, 1, 1}
    PlotPoints(BSpline(MouseDownPointCollection, 3, KN, 0.01))
End Sub

Private Function FindSpan(ByVal n As Integer, ByVal p As Integer, ByVal t As Double, ByVal U() As Double) As
Integer
    If U(n + 1) = t Then Return n

    Dim low, mid, high As Double
    low = p
    high = n + 1

    mid = (low + high) / 2

    While (t < U(mid) Or t >= U(mid + 1))
        If t < U(mid) Then
            high = mid
        Else
            low = mid
        End If

        mid = (low + high) / 2

```

```

    End While

    Return mid
End Function

Private Function BasisFuns(ByVal i As Integer, ByVal t As Double, ByVal p As Integer, ByVal U() As Double) As
Double()
    Dim N(p), left(p), right(p) As Double
    Dim saved As Double
    N(0) = 1

    For j As Integer = 1 To p
        left(j) = t - U(i + 1 - j)
        right(j) = U(i + j) - t
        saved = 0

        For r As Integer = 0 To j - 1
            Dim temp As Double
            temp = N(r) / (right(r + 1) + left(j - r))
            N(r) = saved + right(r + 1) * temp
            saved = left(j - r) * temp
        Next
        N(j) = saved
    Next
    Return N
End Function

Private Function BSplineAtPointT(ByVal n As Integer, ByVal p As Integer, ByVal U() As Double, ByVal PP As
PointCollection, ByVal t As Double) As Point
    Dim span As Integer
    span = FindSpan(n, p, t, U)

    Dim NN() As Double
    NN = BasisFuns(span, t, p, U)

    Dim CX, CY As Double
    CX = 0
    CY = 0
    For i As Integer = 0 To p - 1
        CX = CX + NN(i) * PP(span - p + i).X
        CY = CY + NN(i) * PP(span - p + i).Y
    Next
    Return New Point(CX, CY)
End Function

Private Function BSpline(ByVal p As PointCollection, ByVal Degree As Integer, ByVal KnotVector() As Double, ByVal
StepSize As Double) As PointCollection
    Dim result As New PointCollection

    For k As Double = 0 To 1 Step StepSize
        result.Add(BSplineAtPointT(p.Count, Degree, KnotVector, p, k))
    Next

    Return result
End Function

```



```
Private Sub button21_Click(sender As Object, e As RoutedEventArgs) Handles button21.Click
    CreatMat.Visibility = False
    CreateMat2.Visibility = True
End Sub

Private Sub button22_Click(sender As Object, e As RoutedEventArgs) Handles button22.Click
    Try
        textBox.Text = IO.Directory.GetFiles("C:\Program Files\", "RunWB2.exe", IO.SearchOption.AllDirectories)(0)
    Catch ex As Exception

    End Try
End Sub

Private Sub button5_Copy001_Click(sender As Object, e As RoutedEventArgs) Handles button5_Copy001.Click

End Sub

Private Sub button19_Click(sender As Object, e As RoutedEventArgs) Handles button19.Click

    IzSv.Visibility = Visibility.Hidden

    t1.Visibility = Visibility.Hidden
    t2.Visibility = Visibility.Hidden
    t3.Visibility = Visibility.Hidden
    t4.Visibility = Visibility.Hidden
    t5.Visibility = Visibility.Hidden
    t6.Visibility = Visibility.Hidden
    t7.Visibility = Visibility.Hidden
    t8.Visibility = Visibility.Hidden
    t9.Visibility = Visibility.Hidden
    t10.Visibility = Visibility.Hidden
    t11.Visibility = Visibility.Hidden
    t12.Visibility = Visibility.Hidden
    t13.Visibility = Visibility.Hidden
    t14.Visibility = Visibility.Hidden
    t15.Visibility = Visibility.Hidden
    t16.Visibility = Visibility.Hidden
    t17.Visibility = Visibility.Hidden
    t18.Visibility = Visibility.Hidden
    t19.Visibility = Visibility.Hidden
    t20.Visibility = Visibility.Hidden
    t21.Visibility = Visibility.Hidden
    t22.Visibility = Visibility.Hidden
    t23.Visibility = Visibility.Hidden
    t24.Visibility = Visibility.Hidden
    t25.Visibility = Visibility.Hidden

    tt1.Visibility = Visibility.Hidden
    tt2.Visibility = Visibility.Hidden
    tt3.Visibility = Visibility.Hidden
    tt4.Visibility = Visibility.Hidden
    tt5.Visibility = Visibility.Hidden
    tt6.Visibility = Visibility.Hidden
    tt7.Visibility = Visibility.Hidden
    tt8.Visibility = Visibility.Hidden
    tt9.Visibility = Visibility.Hidden
    tt10.Visibility = Visibility.Hidden
```

```

tt11.Visibility = Visibility.Hidden
tt12.Visibility = Visibility.Hidden
tt13.Visibility = Visibility.Hidden
tt14.Visibility = Visibility.Hidden
tt15.Visibility = Visibility.Hidden
tt16.Visibility = Visibility.Hidden
tt17.Visibility = Visibility.Hidden
tt18.Visibility = Visibility.Hidden
tt19.Visibility = Visibility.Hidden
tt20.Visibility = Visibility.Hidden
tt21.Visibility = Visibility.Hidden
tt22.Visibility = Visibility.Hidden
tt23.Visibility = Visibility.Hidden
tt24.Visibility = Visibility.Hidden

comboBox.Visibility = Visibility.Hidden

For i = 0 To itemsList3.Count - 1
    If (itemsList3(i).Safe = True) Then NameMat = itemsList3(i).Material
Next

Select Case (NameMat)
    Case HaveMat_1(0)
        t1.Text = "Isotropic Relative Permeability"
        tt1.Text = "1"
        tt1.Visibility = False
        t1.Visibility = False
        IzSv.Visibility = False

    Case HaveMat_1(1)
        t1.Text = "Density"
        t2.Text = "Young's Modulus"
        t3.Text = "Poisson's Ratio"
        t4.Text = "Bulk Modulus"
        t5.Text = "Shear Modulus"
        t6.Text = "Coefficient thermal Expansion"
        t7.Text = "Reference Temperature"
        t8.Text = "Tensile Yield Strength"
        t9.Text = "Compressive Yield Strength"
        t10.Text = "Tensile Ultimate Strength"
        t11.Text = "Compressive Ultimate Strength"
        t12.Text = "Specific Heat"
        t13.Text = "Isotropic Relative Permeability"

        tt1.Text = "2770"
        tt2.Text = "71000000000"
        tt3.Text = "0,33"
        tt4.Text = "69607843137,2549"
        tt5.Text = "26691729323,3083"
        tt6.Text = "0,000025"
        tt7.Text = "22"
        tt8.Text = "280000000"
        tt9.Text = "280000000"
        tt10.Text = "310000000"
        tt11.Text = "0"
        tt12.Text = "875"
        tt13.Text = "1"

```

```
t25.Text = "Isotropic Elastic Derive from"
```

```
tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
```

```
t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
```

```
t25.Visibility = False
```

```
IzSv.Visibility = False
comboBox.Visibility = False
```

Case HaveMat_1(2)

```
t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"
t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
```

```
tt1.Text = "2300"
tt2.Text = "30000000000"
tt3.Text = "0,18"
tt4.Text = "15625000000"
tt5.Text = "12711864406,7797"
tt6.Text = "0,000014"
tt7.Text = "22"
```

```

tt8.Text = "0"
tt9.Text = "0"
tt10.Text = "5000000"
tt11.Text = "41000000"
tt12.Text = "780"
tt13.Text = "0.72"

```

```
t25.Text = "Isotropic Elastic Derive from"
```

```

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False

```

```

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False

```

```
t25.Visibility = False
```

```

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_1 (3)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"
t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
t14.Text = "Isotropic Relative Permeability"

```

```

tt1.Text = "8300"
tt2.Text = "110000000000"
tt3.Text = "0,34"
tt4.Text = "114583333333,333"
tt5.Text = "41044776119,403"
tt6.Text = "0,000018"
tt7.Text = "22"
tt8.Text = "280000000"
tt9.Text = "280000000"
tt10.Text = "430000000"
tt11.Text = "0"
tt12.Text = "385"
tt13.Text = "401"
tt14.Text = "1"

t25.Text = "Isotropic Elastic Derive from"

```

```

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
tt14.Visibility = False

```

```

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
t14.Visibility = False

```

```
t25.Visibility = False
```

```
IzSv.Visibility = False
comboBox.Visibility = False
```

Case HaveMat_1(4)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"

```

```

t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"
t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
t14.Text = "Isotropic Relative Permeability"
t15.Text = "Isotropic Resistivity"

```

```

tt1.Text = "7200"
tt2.Text = "110000000000"
tt3.Text = "0,28"
tt4.Text = "83333333333,3333"
tt5.Text = "42968750000"
tt6.Text = "0,000011"
tt7.Text = "22"
tt8.Text = "0"
tt9.Text = "0"
tt10.Text = "240000000"
tt11.Text = "820000000"
tt12.Text = "447"
tt13.Text = "52"
tt14.Text = "10000"
tt15.Text = "0,00000096"

```

```

t25.Text = "Isotropic Elastic Derive from"

```

```

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
tt14.Visibility = False
tt15.Visibility = False

```

```

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False

```

```

t13.Visibility = False
t14.Visibility = False
t15.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_1(5)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"
t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
t14.Text = "Isotropic Relative Permeability"
t15.Text = "Isotropic Resistivity"

tt1.Text = "1800"
tt2.Text = "45000000000"
tt3.Text = "0,35"
tt4.Text = "50000000000"
tt5.Text = "16666666666,6667"
tt6.Text = "0,000026"
tt7.Text = "22"
tt8.Text = "193000000"
tt9.Text = "193000000"
tt10.Text = "255000000"
tt11.Text = "0"
tt12.Text = "1024"
tt13.Text = "156"
tt14.Text = "10000"
tt15.Text = "0,00000077"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False

```

```

tt14.Visibility = False
tt15.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
t14.Visibility = False
t15.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_1(6)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"
t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
't14.Text = "Isotropic Relative Permeability"
't15.Text = "Isotropic Resistivity"

tt1.Text = "950"
tt2.Text = "1100000000"
tt3.Text = "0,42"
tt4.Text = "2291666666,66667"
tt5.Text = "387323943,661972"
tt6.Text = "0,00023"
tt7.Text = "22"
tt8.Text = "25000000"
tt9.Text = "0"
tt10.Text = "33000000"
tt11.Text = "0"
tt12.Text = "296"
tt13.Text = "0,28"
'tt14.Text = "10000"
'tt15.Text = "0,00000077"

t25.Text = "Isotropic Elastic Derive from"

```



```

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
'tt14.Visibility = False
'tt15.Visibility = False

```

```

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
't14.Visibility = False
't15.Visibility = False

```

```
t25.Visibility = False
```

```

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_1 (7)

```

t1.Text = "Density"
t2.Text = "Anisotropic Elasticity M(1,1) "
t3.Text = "M(2,1) "
t4.Text = "M(2,2) "
t5.Text = "M(3,1) "
t6.Text = "M(3,2) "
t7.Text = "M(3,3) "
t8.Text = "M(4,1) "
t9.Text = "M(4,2) "
t10.Text = "M(4,3) "
t11.Text = "M(4,4) "
t12.Text = "M(5,1) "
t13.Text = "M(5,2) "
t14.Text = "M(5,3) "
t15.Text = "M(5,4) "
t16.Text = "M(5,5) "
t17.Text = "M(6,1) "
t18.Text = "M(6,2) "

```

```

t19.Text = "M(6,3) "
t20.Text = "M(6,4) "
t21.Text = "M(6,5) "
t22.Text = "M(6,6) "
t23.Text = "Specific Heat"
t24.Text = "Isotropic Thermal Conductivity"

```

```

tt1.Text = "2330"
tt2.Text = "166000"
tt3.Text = "64000"
tt4.Text = "166000"
tt5.Text = "64000"
tt6.Text = "64000"
tt7.Text = "166000"
tt8.Text = "0"
tt9.Text = "0"
tt10.Text = "0"
tt11.Text = "80000"
tt12.Text = "0"
tt13.Text = "0"
tt14.Text = "0"
tt15.Text = "0"
tt16.Text = "80000"
tt17.Text = "0"
tt18.Text = "0"
tt19.Text = "0"
tt20.Text = "0"
tt21.Text = "0"
tt22.Text = "80000"
tt23.Text = "702"
tt24.Text = "124"

```

```
't25.Text = "Isotropic Elastic Derive from"
```

```

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
tt14.Visibility = False
tt15.Visibility = False
tt16.Visibility = False
tt17.Visibility = False
tt18.Visibility = False
tt19.Visibility = False
tt20.Visibility = False
tt21.Visibility = False
tt22.Visibility = False
tt23.Visibility = False
tt24.Visibility = False

```

```

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
t14.Visibility = False
t15.Visibility = False
t16.Visibility = False
t17.Visibility = False
t18.Visibility = False
t19.Visibility = False
t20.Visibility = False
t21.Visibility = False
t22.Visibility = False
t23.Visibility = False
t24.Visibility = False

't25.Visibility = False

IzSv.Visibility = False
'comboBox.Visibility = False

```

Case HaveMat_1(8)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"
t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
t14.Text = "Isotropic Relative Permeability"
t15.Text = "Isotropic Resistivity"

tt1.Text = "7750"
tt2.Text = "193000000000"
tt3.Text = "0,31"
tt4.Text = "169298245614,035"
tt5.Text = "73664122137,4046"
tt6.Text = "0,000017"
tt7.Text = "22"
tt8.Text = "207000000"
tt9.Text = "207000000"

```

```

tt10.Text = "586000000"
tt11.Text = "0"
tt12.Text = "480"
tt13.Text = "15,1"
tt14.Text = "1"
tt15.Text = "0,00000077"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
tt14.Visibility = False
tt15.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
t14.Visibility = False
t15.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_1(9)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"

```

```

t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
t14.Text = "Isotropic Relative Permeability"
t15.Text = "Isotropic Resistivity"

tt1.Text = "7850"
tt2.Text = "200000000000"
tt3.Text = "0,3"
tt4.Text = "1666666666666,667"
tt5.Text = "76923076923,0769"
tt6.Text = "0,000012"
tt7.Text = "22"
tt8.Text = "250000000"
tt9.Text = "250000000"
tt10.Text = "460000000"
tt11.Text = "0"
tt12.Text = "434"
tt13.Text = "60,5"
tt14.Text = "10000"
tt15.Text = "0,00000017"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
tt14.Visibility = False
tt15.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
t14.Visibility = False
t15.Visibility = False

t25.Visibility = False

```

```
IzSv.Visibility = False
comboBox.Visibility = False
```

```
Case HaveMat_1(10)
```

```
t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Coefficient Thermal Expansion"
t7.Text = "Reference Temperature"
t8.Text = "Tensile Yield Strength"
t9.Text = "Compressive Yield Strength"
t10.Text = "Tensile Ultimate Strength"
t11.Text = "Compressive Ultimate Strength"
t12.Text = "Specific Heat"
t13.Text = "Isotropic Thermal Conductivity"
t14.Text = "Isotropic Relative Permeability"
t15.Text = "Isotropic Resistivity"
```

```
tt1.Text = "4620"
tt2.Text = "96000000000"
tt3.Text = "0,36"
tt4.Text = "114285714285,714"
tt5.Text = "35294117647,0588"
tt6.Text = "0,0000094"
tt7.Text = "22"
tt8.Text = "930000000"
tt9.Text = "930000000"
tt10.Text = "1070000000"
tt11.Text = "0"
tt12.Text = "522"
tt13.Text = "21,9"
tt14.Text = "1"
tt15.Text = "0,0000017"
```

```
t25.Text = "Isotropic Elastic Derive from"
```

```
tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
tt9.Visibility = False
tt10.Visibility = False
tt11.Visibility = False
tt12.Visibility = False
tt13.Visibility = False
tt14.Visibility = False
tt15.Visibility = False
```

```
t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
```

```

t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
t9.Visibility = False
t10.Visibility = False
t11.Visibility = False
t12.Visibility = False
t13.Visibility = False
t14.Visibility = False
t15.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_2 (0)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Yield Strength"
t7.Text = "Tangent Modulus"
t8.Text = "Specific Heat"

tt1.Text = "2770"
tt2.Text = "71000000000"
tt3.Text = "0,33"
tt4.Text = "69607843137,2549"
tt5.Text = "26691729323,3083"
tt6.Text = "280000000"
tt7.Text = "500000000"
tt8.Text = "875"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False

t25.Visibility = False

```

```
IzSv.Visibility = False
comboBox.Visibility = False
```

Case HaveMat_2 (1)

```
t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Maximum Tensile Pressure"
t7.Text = "Fracture Energy Gf"

tt1.Text = "2300"
tt2.Text = "30000000000"
tt3.Text = "0,18"
tt4.Text = "15625000000"
tt5.Text = "12711864406,7797"
tt6.Text = "-4000000"
tt7.Text = "100"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False
```

Case HaveMat_2 (2)

```
t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Yield Strength"
t7.Text = "Tangent Modulus"
t8.Text = "Specific Heat"

tt1.Text = "8300"
tt2.Text = "110000000000"
tt3.Text = "0,34"
tt4.Text = "114583333333,333"
```



```

tt5.Text = "41044776119,403"
tt6.Text = "280000000"
tt7.Text = "1150000000"
tt8.Text = "385"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_2(3)

```

t1.Text = "Temperature"

tt1.Text = "100"

IzSv.Visibility = False

```

Case HaveMat_2(4)

```

t1.Text = "Temperature"

tt1.Text = "100"

IzSv.Visibility = False

```

Case HaveMat_2(5)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Yield Strength"
t7.Text = "Tangent Modulus"
t8.Text = "Specific Heat"

tt1.Text = "1800"
tt2.Text = "45000000000"
tt3.Text = "0,35"

```

```

tt4.Text = "50000000000"
tt5.Text = "16666666666,6667"
tt6.Text = "193000000"
tt7.Text = "920000000"
tt8.Text = "1024"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False

```

Case HaveMat_2 (6)

```

t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Yield Strength"
t7.Text = "Tangent Modulus"
t8.Text = "Specific Heat"

tt1.Text = "7750"
tt2.Text = "193000000000"
tt3.Text = "0,31"
tt4.Text = "169298245614,035"
tt5.Text = "73664122137,4046"
tt6.Text = "210000000"
tt7.Text = "1800000000"
tt8.Text = "480"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False

```

```
tt7.Visibility = False
tt8.Visibility = False
```

```
t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
```

```
t25.Visibility = False
```

```
IzSv.Visibility = False
comboBox.Visibility = False
```

Case HaveMat_2 (7)

```
t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Yield Strength"
t7.Text = "Tangent Modulus"
t8.Text = "Specific Heat"
```

```
tt1.Text = "7850"
tt2.Text = "2000000000000"
tt3.Text = "0,3"
tt4.Text = "166666666666,667"
tt5.Text = "76923076923,0769"
tt6.Text = "250000000"
tt7.Text = "1450000000"
tt8.Text = "434"
```

```
t25.Text = "Isotropic Elastic Derive from"
```

```
tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False
```

```
t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False
```

```
t25.Visibility = False
```

```
IzSv.Visibility = False
comboBox.Visibility = False
```

Case HaveMat_2 (8)

```
t1.Text = "Density"
t2.Text = "Young's Modulus"
t3.Text = "Poisson's Ratio"
t4.Text = "Bulk Modulus"
t5.Text = "Shear Modulus"
t6.Text = "Yield Strength"
t7.Text = "Tangent Modulus"
t8.Text = "Specific Heat"

tt1.Text = "4620"
tt2.Text = "96000000000"
tt3.Text = "0,36"
tt4.Text = "114285714285,714"
tt5.Text = "35294117647,0588"
tt6.Text = "930000000"
tt7.Text = "2150000000"
tt8.Text = "522"

t25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False
tt8.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False
t8.Visibility = False

t25.Visibility = False

IzSv.Visibility = False
comboBox.Visibility = False
```

Case HaveMat_3 (0)

```
t1.Text = "Density"
t2.Text = "Specific Heat"
t3.Text = "Adiabatic Exponent  $\gamma$ "
t4.Text = "Adiabatic Constant"
t5.Text = "Pressure Shift"
t6.Text = "Reference Temperature"
t7.Text = "Specific Internal Energy"

tt1.Text = "1,225"
```

```

tt2.Text = "717,6"
tt3.Text = "1,4"
tt4.Text = "0"
tt5.Text = "0"
tt6.Text = "15,05"
tt7.Text = "200000"

't25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt6.Visibility = False
tt7.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t6.Visibility = False
t7.Visibility = False

't25.Visibility = False

IzSv.Visibility = False
'comboBox.Visibility = False

```

Case HaveMat_3(1)

```

t1.Text = "Density"
t2.Text = "Specific Heat"
t3.Text = "Initial Yield Stredd Y"
t4.Text = "Maximum Yield Stress Ymax"
t5.Text = "Hardening Constant B"
t6.Text = "Hardening Exponent n"
t7.Text = "Derivative dG/dP G'P"
t8.Text = "Derivative dG/dT G'T"
t9.Text = "Derivative dY/dP Y'P"
t10.Text = "Melting Temperature Tmelt"
t11.Text = "Shear Modulus"
t12.Text = "Gruneisen Coefficient"
t13.Text = "Parameter C1"
t14.Text = "Parameter S1"
t15.Text = "Parameter Quadratic S2"

tt1.Text = "2707"
tt2.Text = "884"
tt3.Text = "40000000"
tt4.Text = "480000000"
tt5.Text = "400"
tt6.Text = "0,27"
tt7.Text = "1,767"
tt8.Text = "-16690000"
tt9.Text = "0,002608"
tt10.Text = "946,85"

```

```

tt11.Text = "27100000000"
tt12.Text = "1,97"
tt13.Text = "5386"
tt14.Text = "1,339"
tt15.Text = "0"

't25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt8.Visibility = False
tt9.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t8.Visibility = False
t9.Visibility = False

't25.Visibility = False

IzSv.Visibility = False
'comboBox.Visibility = False

```

Case HaveMat_3 (2)

```

t1.Text = "Density"
't2.Text = "Specific Heat"
't3.Text = "Initial Yield Stredd Y"
't4.Text = "Maximum Yield Stress Ymax"
't5.Text = "Hardening Constant B"
't6.Text = "Hardening Exponent n"
't7.Text = "Derivative dG/dP G'P"
't8.Text = "Derivative dG/dT G'T"
't9.Text = "Derivative dY/dP Y'P"
't10.Text = "Melting Temperature Tmelt"
't11.Text = "Shear Modulus"
t12.Text = "Gruneisen Coefficient"
t13.Text = "Parameter C1"
t14.Text = "Parameter S1"
t15.Text = "Parameter Quadratic S2"

tt1.Text = "2785"
'tt2.Text = "884"
'tt3.Text = "40000000"
'tt4.Text = "480000000"
'tt5.Text = "400"
'tt6.Text = "0,27"
'tt7.Text = "1,767"
'tt8.Text = "-16690000"
'tt9.Text = "0,002608"
'tt10.Text = "946,85"
'tt11.Text = "27100000000"
tt12.Text = "2"

```

```

tt13.Text = "5386"
tt14.Text = "1,338"
tt15.Text = "0"

't25.Text = "Isotropic Elastic Derive from"

tt1.Visibility = False
tt2.Visibility = False
tt3.Visibility = False
tt4.Visibility = False
tt5.Visibility = False
tt8.Visibility = False
tt9.Visibility = False

t1.Visibility = False
t2.Visibility = False
t3.Visibility = False
t4.Visibility = False
t5.Visibility = False
t8.Visibility = False
t9.Visibility = False

't25.Visibility = False

IzSv.Visibility = False
'comboBox.Visibility = False

    Case Else
End Select

End Sub

Private Sub Safe_Click(sender As Object, e As RoutedEventArgs) Handles Safe.Click

End Sub

Private Sub TakeMat_Click(sender As Object, e As RoutedEventArgs) Handles TakeMat.Click

    num = 0
    ReDim Arr(0)
    ReDim S(0)
    For index = 0 To itemList.Count - 1
        If itemList(index).Safe = True Then
            Arr(UBound(Arr)) = CInt(itemList(index).Number) - 1
            ReDim Preserve Arr(UBound(Arr) + 1)
            num += 1
        End If
    Next index
    CreatMat.Visibility = True
    CreateMat2.Visibility = False

    itemList3.Clear()
    ListView3.Items.Refresh()

    For i = 0 To num - 1
        S(i) = itemList(Arr(i)).Material
        ReDim Preserve S(UBound(S) + 1)

```

```

Next

item3 = New ListViewItemTemplate3
item3.Material = S(0)
item3.Safe = False
itemsList3.Add(item3)

For i = 1 To UBound(S) - 1
    item3 = New ListViewItemTemplate3
    item3.Material = S(i)
    item3.Safe = False
    itemsList3.Add(item3)
Next i

ListView3.ItemsSource = itemsList3
ListView3.Items.Refresh()

End Sub

Private Function LagrangeBasisFunction(ByVal n As Integer, ByVal k As Integer, ByVal t As Double) As Double
    Dim l As Double = 1
    Dim tj, tk As Double
    tk = k / (n - 1)

    For j As Integer = 0 To n - 1
        If j <> k Then
            tj = j / (n - 1)
            l *= (t - tj) / (tk - tj)
        End If
    Next
    Return l
End Function

Private Sub StartBut_Click(sender As Object, e As RoutedEventArgs) Handles StartBut.Click
    s1.Visibility = False
    button4.Visibility = True
    button3.Visibility = False
    button.Visibility = False
    Start.Visibility = True
    StartBut.Visibility = True
End Sub

Private Sub button11_Click(sender As Object, e As RoutedEventArgs) Handles button11.Click
    cnvMain.Children.Clear()
    MouseDownPointCollection.Clear()

    If ListOfBezierWeights.Count <> 0 Then ListOfBezierWeights(0).reset()

    ListOfBezierWeights.Clear()
End Sub

Private Sub button15_Click(sender As Object, e As RoutedEventArgs) Handles button15.Click
    IssCreate.Visibility = True
    FormCreate.Visibility = False
    labell13.Content = "Геометрия исследуемой области создана"
End Sub

Private Function GetLagrangianAtPointT(ByVal p As PointCollection, ByVal t As Double) As Point

```



```

Dim n As Integer = p.Count

Dim rx, ry As Double
rx = 0
ry = 0
For i As Integer = 0 To n - 1
    rx += LagrangeBasisFunction(n, i, t) * p(i).X
    ry += LagrangeBasisFunction(n, i, t) * p(i).Y
Next
Return New Point(rx, ry)
End Function

Private Function LagrangianInterpolation(ByVal p As PointCollection, Optional ByVal StepSize As Double = 0.01) As
PointCollection
    Dim result As New PointCollection

    For i As Double = 0 To 1 Step StepSize
        result.Add(GetLagrangianAtPointT(p, i))
    Next

    result.Add(p(p.Count - 1))
    Return result
End Function

Public Function ConvexHull2D(ByVal points As PointCollection) As PointCollection
    Dim SortedList As New PointCollection
    Dim Upper As New PointCollection
    Dim Lower As New PointCollection
    Dim tempPoints As New PointCollection

    For Each p As Point In points
        If Not tempPoints.Contains(p) Then tempPoints.Add(p)
    Next

    Dim res As New PointCollection

    SortedList = SortPoints(tempPoints, True)

    Upper = SortedList
    For i As Integer = SortedList.Count - 1 To 0 Step -1

        Lower.Add(SortedList(i))
    Next

    Dim j_set As Boolean = False
    Dim j As Integer = 0
    Dim r As Integer
    Do While j < Upper.Count - 2
        r = WhichSide(Upper(j + 1), Upper(j), Upper(j + 2))
        If r = -1 Or r = 0 Then
            Upper.RemoveAt(j + 1)
            j = 0
            j_set = True
        End If

        If Not j_set Then
            j += 1
        End If
    Loop

```

```

        End If

        j_set = False
    Loop

    j = 0
    j_set = False
    Do While j < Lower.Count - 2
        r = WhichSide(Lower(j + 1), Lower(j), Lower(j + 2))
        If r = -1 Or r = 0 Then
            Lower.RemoveAt(j + 1)
            j = 0
            j_set = True
        End If

        If Not j_set Then
            j += 1
        End If

        j_set = False
    Loop

    For Each p As Point In Upper
        res.Add(p)
    Next

    For Each p As Point In Lower
        If Not res.Contains(p) Then
            res.Add(p)
        End If
    Next

    res.Add(res(0))

    Return res
End Function

Private Function WhichSide(ByVal PointToBeEvaluated As Point, ByVal StartPointOnLine As Point, ByVal
EndPointOnLine As Point) As Integer
    Dim ReturnValueEquation As Double
    ReturnValueEquation = ((PointToBeEvaluated.Y - StartPointOnLine.Y) _
        * (EndPointOnLine.X - StartPointOnLine.X) - ((EndPointOnLine.Y - StartPointOnLine.Y) _
        * (PointToBeEvaluated.X - StartPointOnLine.X))

    If ReturnValueEquation > 0 Then
        Return -1
    ElseIf ReturnValueEquation = 0 Then
        Return 0
    Else
        Return 1
    End If

End Function

Public Function SortPoints(ByVal samplepoints As PointCollection, ByVal SortByXdirection As Boolean) As
PointCollection

    Dim copySamplePoints As Point() = New Point(samplepoints.Count - 1) {}

```

```

samplepoints.CopyTo(copySamplePoints, 0)

If SortByXdirection Then
    Array.Sort(copySamplePoints, New PointSort(PointSort.Mode.X))
Else
    Array.Sort(copySamplePoints, New PointSort(PointSort.Mode.Y))
End If

Dim result As New PointCollection

For Each p As Point In copySamplePoints
    result.Add(p)
Next

Return result
End Function

Private Sub button13_Click(sender As Object, e As RoutedEventArgs) Handles button13.Click
    Dim X, Y As String

    Dim myStr As String
    Dim g(256) As String

    X = textBox5.Text
    Y = textBox6.Text

    myStr = X + "," + Y

    Dim PointR As Point

    PointR = Point.Parse(myStr)
    Dim radius As Double
    radius = 2.5

    Dim c As New Ellipse
    c.Fill = Brushes.Red
    c.Width = radius * 2
    c.Height = radius * 2

    Canvas.SetLeft(c, PointR.X - c.Height / 2)
    Canvas.SetTop(c, PointR.Y - c.Width / 2)
    cnvMain.Children.Add(c)
    MouseDownPointCollection.Add(PointR)

End Sub

Private Sub button2001_Click(sender As Object, e As RoutedEventArgs) Handles button2001.Click
    First001.Visibility = Visibility.Hidden
    CreateMat.Visibility = Visibility.Visible
End Sub

Private Sub button5001_Click(sender As Object, e As RoutedEventArgs) Handles button5001.Click

End Sub

Private Sub button6001_Click(sender As Object, e As RoutedEventArgs) Handles button6001.Click
    CreateMat.Visibility = Visibility.Hidden

```

```

First001.Visibility = Visibility.Visible
End Sub

Private Sub button7001_Click(sender As Object, e As RoutedEventArgs) Handles button7001.Click

    NumUserLib2 = 8

    itemsList2.Clear()
    ListView2.Items.Refresh()

    If (NumUserLib <> 12) Then
        If (HaveLib(NumUserLib) = "-1") Then
            A = NumUserLib
            Select Case (NumUserLib)
                Case 8
                    B = 4
                Case 9
                    B = 3
                Case 10
                    B = 2
                Case 11
                    B = 1
            End Select
        Else
            A = 0
            B = 0
        End If
    Else
        A = 0
        B = 0
    End If

    item2 = New ListViewItemTemplate2
    item2.Library = HaveLib(NumUserLib2)

    item2.Safe2 = False
    itemsList2.Add(item2)

    For k1 = NumUserLib2 + 1 To UBound(HaveLib) - B
        item2 = New ListViewItemTemplate2
        item2.Library = HaveLib(k1)
        itemsList2.Add(item2)
    Next K1

    ListView2.ItemsSource = itemsList2
    ListView2.Items.Refresh()

    label2001.Content = "Пользовательские библиотеки загружены в таблицу. Выберите библиотеку для добавления в нее
    нового материала. имеется возможность выбора нескольких библиотек."

End Sub

Private Sub button001_Click(sender As Object, e As RoutedEventArgs) Handles button001.Click
    If (NumUserLib <> 12) Then
        CreateLib.Visibility = Visibility.Visible
    Else : label2001.Content = "Вы достигли ограничения в создании пользовательских библиотек (4 библиотеки)"
    End If
End Sub

```

```

Private Sub button3001_Click(sender As Object, e As RoutedEventArgs) Handles button3001.Click

End Sub

Private Sub button4001_Click(sender As Object, e As RoutedEventArgs) Handles button4001.Click
    HaveLib(NumUserLib) = textBox001.Text
    NumUserLib += 1
    textBlock001.Text = "Библиотека создана. Активирована кнопка создания материала. В целях сохранения
целостности структуры проекта добавление новых материалов возможно только в новые библиотеки."
    CreateLib.Visibility = Visibility.Hidden
    button2001.Visibility = Visibility.Visible
    itemList.Clear()
    ListView1.Items.Refresh()
End Sub

Private Sub button1001_Click(sender As Object, e As RoutedEventArgs) Handles button1001.Click

    itemList.Clear()
    ListView1.Items.Refresh()

    If (NumUserLib <> 12) Then
        If (HaveLib(NumUserLib) = "-1") Then
            A = NumUserLib
            Select Case (NumUserLib)
                Case 8
                    B = 4
                Case 9
                    B = 3
                Case 10
                    B = 2
                Case 11
                    B = 1
            End Select
        Else
            A = 0
            B = 0
        End If
    Else
        A = 0
        B = 0
    End If

    j = 1
    q = 1

    item = New ListViewItemTemplate
    item.Number = "1"
    item.Library = HaveLib(0)
    item.Material = HaveMat_1(0)
    item.Safe = False
    itemList.Add(item)

    For i = 1 To UBound(HaveMat_1)
        item = New ListViewItemTemplate
        Text = CStr(j + 1)
        item.Number = Text

```

```

        item.Material = HaveMat_1(i)
        item.Safe = False
        itemsList.Add(item)
        j += 1
    Next i

    For k1 = 1 To UBound(FullMat) - B
        item = New ListViewItemTemplate
        item.Number = CStr(j + 1)
        item.Library = HaveLib(q)
        item.Material = FullMat(k1)(0)
        item.Safe = False
        itemsList.Add(item)
        j += 1
        q += 1
        If (q = A) Then q -= 1

        For i = 1 To UBound(FullMat(k1))
            item = New ListViewItemTemplate
            Text = CStr(j + 1)
            item.Number = Text
            item.Material = FullMat(k1)(i)
            item.Safe = False
            itemsList.Add(item)
            j += 1
        Next i
    Next K1

    ListView1.ItemsSource = itemsList
    ListView1.Items.Refresh()

    textBlock001.Text = "Библиотеки загружены в таблицу"

End Sub

Private Sub comboBox_SelectionChanged(sender As Object, e As SelectionChangedEventArgs) Handles
comboBox.SelectionChanged
    BoxSel = comboBox.SelectedIndex
        Y=Cdbl(tt2.Text)
        V=Cdbl(tt3.Text)
        K=Cdbl(tt4.Text)
        G=Cdbl(tt5.Text)

    Select Case (BoxSel)
        Case 0
            tt5.Text = CStr(Y / (2 * (1 + V)))
            tt4.Text = CStr(Y / (3 * (1 - 2 * V)))
            tt4.IsReadOnly = True
            tt5.IsReadOnly = True
            tt2.IsReadOnly = False
            tt3.IsReadOnly = False

            tt4.Background = Brushes.Gray
            tt5.Background = Brushes.Gray
            tt2.Background = Brushes.White
            tt3.Background = Brushes.White

        Case 1
            tt2.Text = CStr((2 * G) * (1 + V))

```

```

tt4.Text = CStr((2 * G * (1 + V)) / (3 * (1 - 2 * V)))
tt4.IsReadOnly = True
tt2.IsReadOnly = True
tt3.IsReadOnly = False
tt5.IsReadOnly = False

tt4.Background = Brushes.Gray
tt2.Background = Brushes.Gray
tt3.Background = Brushes.White
tt5.Background = Brushes.White

```

Case 2

```

tt2.Text = CStr(G * (2 * (1 + V)))
tt5.Text = CStr((Y / (2 * (1 + V))))
tt2.IsReadOnly = True
tt5.IsReadOnly = True
tt3.IsReadOnly = False
tt4.IsReadOnly = False

tt2.Background = Brushes.Gray
tt5.Background = Brushes.Gray
tt3.Background = Brushes.White
tt4.Background = Brushes.White

```

Case 3

```

tt3.Text = CStr((Y - 2 * G) / (2 * G))
tt4.Text = CStr(Y / (3 * (1 - 2 * V)))
tt3.IsReadOnly = True
tt4.IsReadOnly = True
tt2.IsReadOnly = False
tt5.IsReadOnly = False

tt4.Background = Brushes.Gray
tt3.Background = Brushes.Gray
tt2.Background = Brushes.White
tt5.Background = Brushes.White

```

Case 4

```

tt3.Text = CStr(((3 * K) - Y) / (6 * K))
tt5.Text = CStr((3 * Y * K) / (9 * K - Y))
tt3.IsReadOnly = True
tt5.IsReadOnly = True
tt4.IsReadOnly = False
tt2.IsReadOnly = False

tt5.Background = Brushes.Gray
tt3.Background = Brushes.Gray
tt2.Background = Brushes.White
tt4.Background = Brushes.White

```

Case 5

```

tt2.Text = CStr((9 * G * K) / (3 * K + G))
tt3.Text = CStr((3 * K - 2 * G) / (2 * G + 6 * K))
tt2.IsReadOnly = True
tt3.IsReadOnly = True
tt4.IsReadOnly = False
tt5.IsReadOnly = False

```

```
tt3.Background = Brushes.Gray  
tt2.Background = Brushes.Gray  
tt4.Background = Brushes.White  
tt5.Background = Brushes.White
```

```
End Select
```

```
End Sub
```

```
End Class
```