

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Инженерная школа ядерных технологий
Направление подготовки: Прикладная математика и информатика
Отделение экспериментальной физики

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка создания геометрии объектов в граничном представлении для подготовки начальных данных инженерного анализа в ПК ЛОГОС

УДК 622.276.344:519.876

Студент

Группа	ФИО	Подпись	Дата
ОВМ71	Кузнецов Георгий Олегович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Начальник научно-исследовательской группы РФЯЦ-ВНИИЭФ ИТМФ	Фархутдинов В.Ф.			

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Меньшикова Е.В.	Кандидат философских наук		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель	Исаева Е.С.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Семенов М.Е.	Кандидат ф-м. наук		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
0BM71	Кузнецов Георгий Олегович

Школа	ИЯТШ	Отделение (НОЦ)	ОЭФ
Уровень образования	Магистр	Направление/специальность	Прикладная математика и информатика

Тема ВКР:

Разработка алгоритмов создания геометрических объектов в граничном представлении для подготовки начальных данных инженерного анализа в программном комплексе ЛОГОС.	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Целью магистерской диссертации является исследование геометрических ядер и их функционал, создание алгоритмов построения поверхностей. Работа выполнялась на базе НИ ТПУ, ОЭФ, 10 корпус, аудитория 427А.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	1. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018) 2. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования. 3. ГОСТ 12.2.049-80 ССБТ. Оборудование производственное. Общие эргономические требования. 4. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. 5. ГОСТ Р ИСО 14738-2007. Безопасность машин. Антропометрические требования при проектировании рабочих мест машин.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов	– Отклонение показателей микроклимата

2.2. Обоснование мероприятий по снижению воздействия	– Недостаточная освещенность рабочей Зоны – Повышенное образование электростатических зарядов – Наличие электромагнитных полей радиочастотного диапазона – Повышенный уровень шума – Нервно-психические перегрузки, связанные с напряженностью трудового процесса
3. Экологическая безопасность:	– Загрязнение литосферы побочными отходами трудовой деятельности
4. Безопасность в чрезвычайных ситуациях:	– Пожар, сильные морозы, диверсия

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель	Исаева Елизавета Сергеевна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0BM71	Кузнецов Георгий Олегович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
0BM71	Кузнецов Георгий Олегович

Школа	ИЯТШ	Отделение школы (НОЦ)	ОЭФ
Уровень образования	Магистр	Направление/специальность	Прикладная математика и информатика

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	1. Стоимость расходных материалов 2. Норматив заработной платы
2. Нормы и нормативы расходования ресурсов	1. Коэффициенты для расчета заработной платы
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	1. Отчисления во внебюджетные фонды (27,1%); 2. Расчеты дополнительной заработной платы

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	1. Потенциальные потребители результатов исследования; 2. Анализ конкурентных технических решений; 3. SWOT – анализ.
4. Планирование и формирование бюджета научных исследований	1. Структура работ в рамках научного исследования; 2. Определение трудоемкости выполнения работ и разработка графика проведения научного исследования; 3. Бюджет научно-технического исследования (НТИ)
1. Определение ресурсной, финансовой, экономической эффективности	1. Определение интегрального финансового показателя 2. Определение интегрального показателя ресурсоэффективности разработки; 3. Определение интегрального показателя эффективности.

Перечень графического материала (с точным указанием обязательных чертежей):

1. «Портрет» потребителя результатов НТИ 2. Сегментирование рынка 3. Оценка конкурентоспособности технических решений 4. Диаграмма FAST 5. Матрица SWOT 6. График проведения и бюджет НТИ 7. Оценка ресурсной, финансовой и экономической эффективности НТИ 8. Потенциальные риски

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Меньшикова Екатерина Викторовна	Кандидат философских наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0BM71	Кузнецов Георгий Олегович		

РЕФЕРАТ

Пояснительная записка к магистерской диссертации выполнена на 132 страницах машинописного текста, содержит 21 таблицу, 39 рисунков, 82 формулы, 27 источников, 4 приложения.

Ключевые слова: системы автоматизированного проектирования, геометрическое ядро, геометрическое представление объектов, кривые, поверхности.

Объект исследования: структура объектов геометрического ядра.

Цель исследования: исследование значимости разработки отечественного инженерно-технического программного комплекса, разработка алгоритмов дополняющих инструментарий ПК ЛОГОС.

Метода проведения исследования: теоретические и практические.

Полученные результаты: изучена структура геометрического ядра. Реализованы алгоритм построения поверхностей, которые были внедрены в ПК ЛОГОС.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	16
1. ОБЩАЯ ИНФОРМАЦИЯ	18
1.1. Системы автоматизированного проектирования CAD.....	18
1.2. Автоматизированные системы инженерного анализа CAE	19
1.3. Геометрическое (математическое) ядро	21
1.3.1. Основные типы геометрических ядер для создания CAD/CAE систем	21
1.3.2. ПК ЛОГОС на основе 3DTV	23
1.4. Типы геометрических представлений	23
1.4.1. B-rep	23
1.4.2. CSG	24
1.4.3. Faceted.....	24
2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	26
2.1. Геометрическое моделирование	26
Геометрические объекты	26
2.2. Точка.....	26
2.3. Линии. Геометрия кривой.....	26
2.3.1. Кривые Безье	28
2.3.2. Сплайн	29
2.3.3. B-сплайн	31
2.3.4. NURBS кривые	32
2.4. Геометрия поверхности	38
2.4.1. Поверхности движения.....	38
2.4.2. Линейчатые поверхности	40
2.4.3. Сплайновые поверхности	42
2.4.4. Поверхности Безье	44
2.4.5. Рациональные поверхности	45
2.4.6. NURBS – поверхности	48
2.5. Топология	53
3. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	55
3.1. Описание реализованных алгоритмов	55
3.2. Пример использования разработанного инструмента «Создание поверхности протягиванием»	
56	

3.3. Пример использования разработанного инструмента «Создание поверхности вращения»	66
4. Социальная ответственность	71
4.1. Правовые и организационные вопросы обеспечения безопасности	71
4.1.1. Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудоустройства	71
4.1.2. Организационные мероприятия при компоновке рабочей зоны.....	72
4.2. Профессиональная социальная ответственность	73
4.2.1. Анализ вредных и опасных факторов, которые может создать объект исследования...	73
4.2.2. Обоснование мероприятий по защите персонала предприятия от действия опасных и вредных.....	84
4.3. Экологическая безопасность.....	87
4.3.1. Анализ влияния объекта исследования на окружающую среду	87
4.3.2. Обоснование мероприятий по защите окружающей среды	87
4.4. Безопасность в чрезвычайной ситуации	87
4.4.1. Анализ вероятных ЧС, которые может инициировать объект исследований	87
4.4.2. Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС	88
4.5. Выводы и рекомендации	90
5. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	90
5.1. Потенциальные потребители результатов исследования	90
5.2. Анализ конкурентных технических решений	91
5.3. SWOT-анализ.....	93
5.4. Инициация проекта.....	95
5.5. Определение трудоемкости работ	97
5.6. Бюджет научно-технического исследования	100
5.7. Расчёт материальных затрат НТИ.....	101
5.8. Основная заработная плата	101
5.9. Дополнительная заработная плата	103
5.10. Отчисления во внебюджетные фонды	103
5.11. Накладные расходы	104
5.12. Формирование бюджета затрат НТИ	105
5.13. Реестр рисков проекта	105
5.14. Оценки сравнительной эффективности исследования.....	106
5.15. Выводы	108
ЗАКЛЮЧЕНИЕ	109

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	110
Приложение А	112
ПРИЛОЖЕНИЕ В. Код программы на С++ «Создание поверхности протягиванием»	122
ПРИЛОЖЕНИЕ С. Код программы на С++. Класс, отвечающий за создание контура	133
ПРИЛОЖЕНИЕ D. Код программы на С++ «Создание поверхности вращением кривой вокруг заданной оси»	135

ВВЕДЕНИЕ

В современной промышленности все большую роль играет компьютерное моделирование и инженерный анализ. Их использование позволяет во много раз сократить затраты времени и средств на разработку, создание и модернизацию различных технических устройств, от простейших конструкций до самолетов и атомных электростанций.

Опыт разработки новой техники и тенденции развития конструкции и методов исследований показывают, что ведущую роль в обеспечении конкурентных преимуществ играют технологии компьютерного моделирования отдельных свойств систем и виртуального проектирования объекта целиком. Это направление занимает все большее место в процессе проектирования, обеспечивая поиск оптимальных конструкторских решений и сокращение сроков разработки и снижения затрат на испытания [15].

В настоящее время в РФЯЦ-ВНИИЭФ ведутся работы по созданию и внедрению отечественного программного обеспечения ЛОГОС для комплексного имитационного моделирования, предназначенного для использования на предприятиях высокотехнологичных отраслей промышленности [16].

Создание комплекса программ инженерного анализа, позволяющих проводить исследование конструкций с учетом различных процессов, требует решения первоочередной задачи — это обработка конструкторских моделей для такого моделирования. Программный комплекс должен быть снабжен всем необходимым инструментарием для обработки и редактирования конструкторских моделей.

Неотъемлемой частью любого программного обеспечения для инженерного анализа, является геометрическое ядро. Геометрическое (математическое) ядро — это набор функций, выполнение которых обеспечивает импорт, экспорт и построение трехмерных моделей. Ядро создается для использования в прикладных программах. Доступ к функциям ядра конечному пользователю открывается через графический пользовательский интерфейс. Кроме того, ядро иногда называют «движком» системы геометрического моделирования. Подобно тому, как двигатель автомобиля определяет «потолок» его скорости, математическое ядро определяет предел функциональных возможностей использующей его САЕ-системы, в части работы с геометрией.

Была поставлена задача разработки алгоритмов создания геометрических объектов в граничном представлении для подготовки начальных данных инженерного анализа. Для выполнения поставленной задачи необходимо:

- изучить типы представления геометрии в программном комплексе ЛОГОС математические модели геометрических кривых и поверхностей;
- изучить организацию структур данных геометрического ядра и их соответствие со структурами данных ПреПостПроцессора;
- разработать алгоритмы «создания поверхности протягиванием вдоль направляющего вектора на заданное расстояние» и «создания поверхности вращением относительно заданной оси».

1. ОБЩАЯ ИНФОРМАЦИЯ

Система автоматизированного проектирования (САПР) - система, предназначенная для автоматизации процесса проектирования, которая облегчает процесс создания, изменения и анализа разрабатываемых проектов. Системы автоматизированного проектирования (CAD) обычно применяются совместно с системами автоматизированного инженерного анализа (CAE) и автоматизированным производством (CAM), позволяя решать задачи выбора оптимального облика конструкции и технологию ее производства.

1.1. Системы автоматизированного проектирования CAD

CAD (Computer Aided Design) – это система автоматизированного проектирования, обладающая различными компонентами, которые предоставляют инженерам возможность создавать и изменять различные инженерные, архитектурно-строительные проекты.

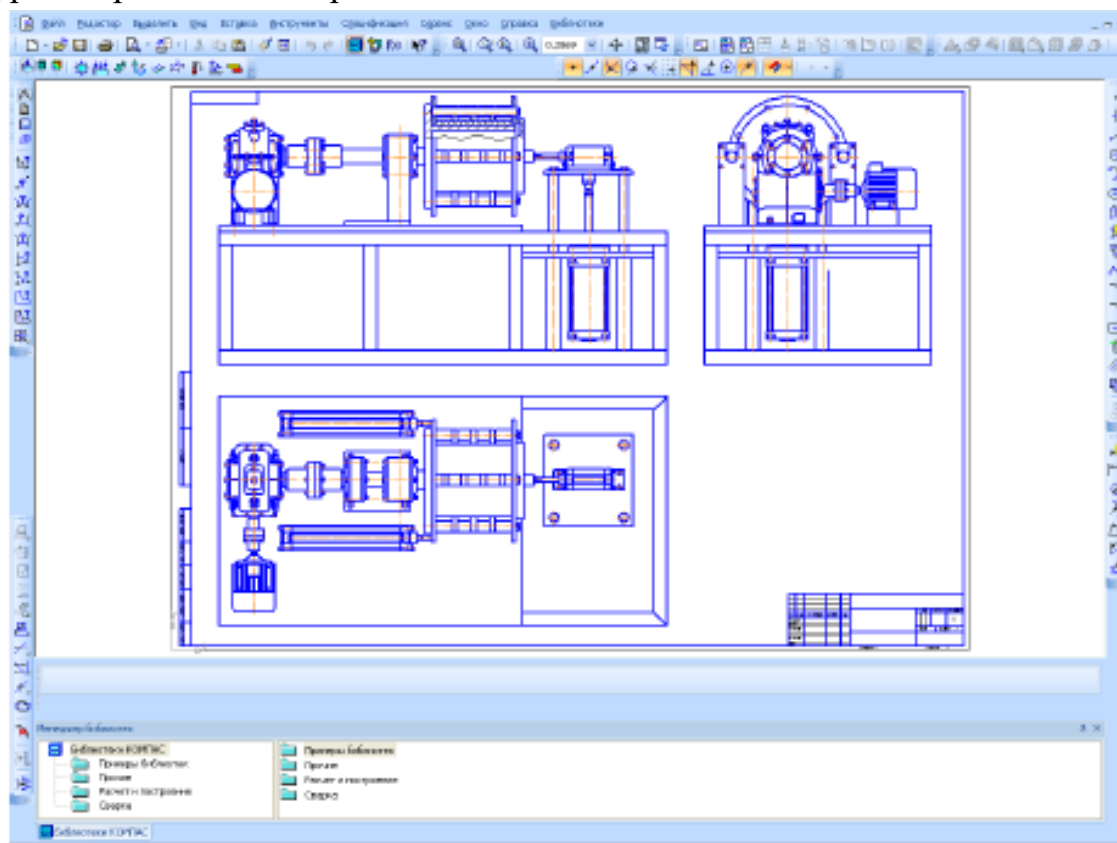


Рисунок 1 - Пример чертежа разрабатываемой конструкции в программе КОМПАС-3D

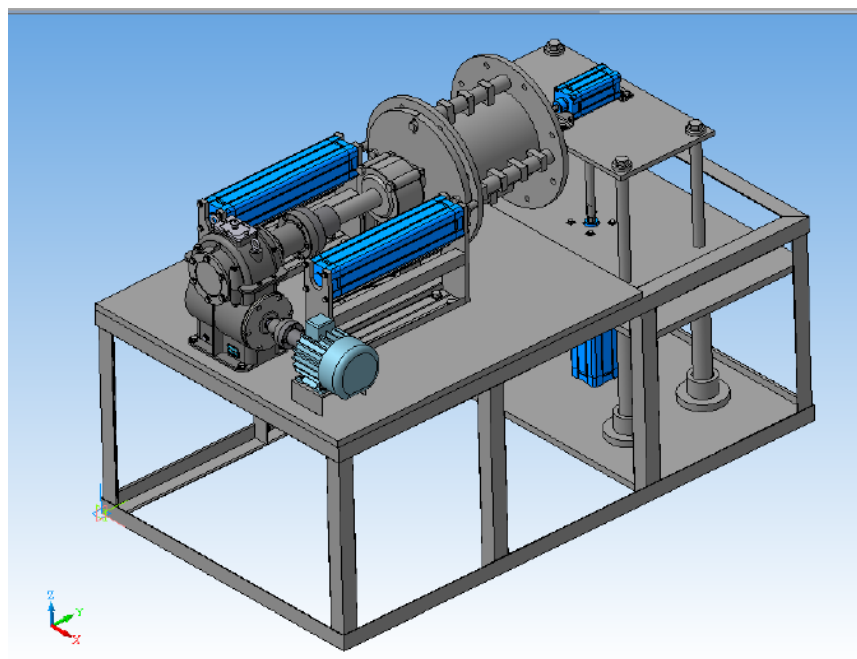


Рисунок 2 - Пример 3D модели конструкции в программе КОМПАС-3D

Основные функции САД-систем

CAD системы предоставляют возможность работы с двумерным (2D) и трехмерным (3D) моделированием, возможность проектирования от простейших деталей до сложных конструкций.

Двумерное моделирование позволяет создавать модель в виде плоской проекции на виртуальный лист бумаги. Примером 2D моделирования могут служить чертежи, спецификации.

Трехмерное моделирование предназначено для создания объектов в пространстве. Примеры САД систем представлены в таблице 1.

Таблица 1 - Примеры САД систем

Зарубежные	Отечественные
NX	T-FLEX
SolidWorks	КОМПАС – 3D
Autodesk	nanoCAD
CATIA	ADEM
FreeCAD	

1.2. Автоматизированные системы инженерного анализа САЕ

САЕ (Computer Aided Engineering) — автоматизированная система инженерного анализа, позволяет проводить расчетно-теоретические обоснования

разрабатываемых изделий в условиях приближенных к реальным, не прибегая к дорогостоящим опытам.

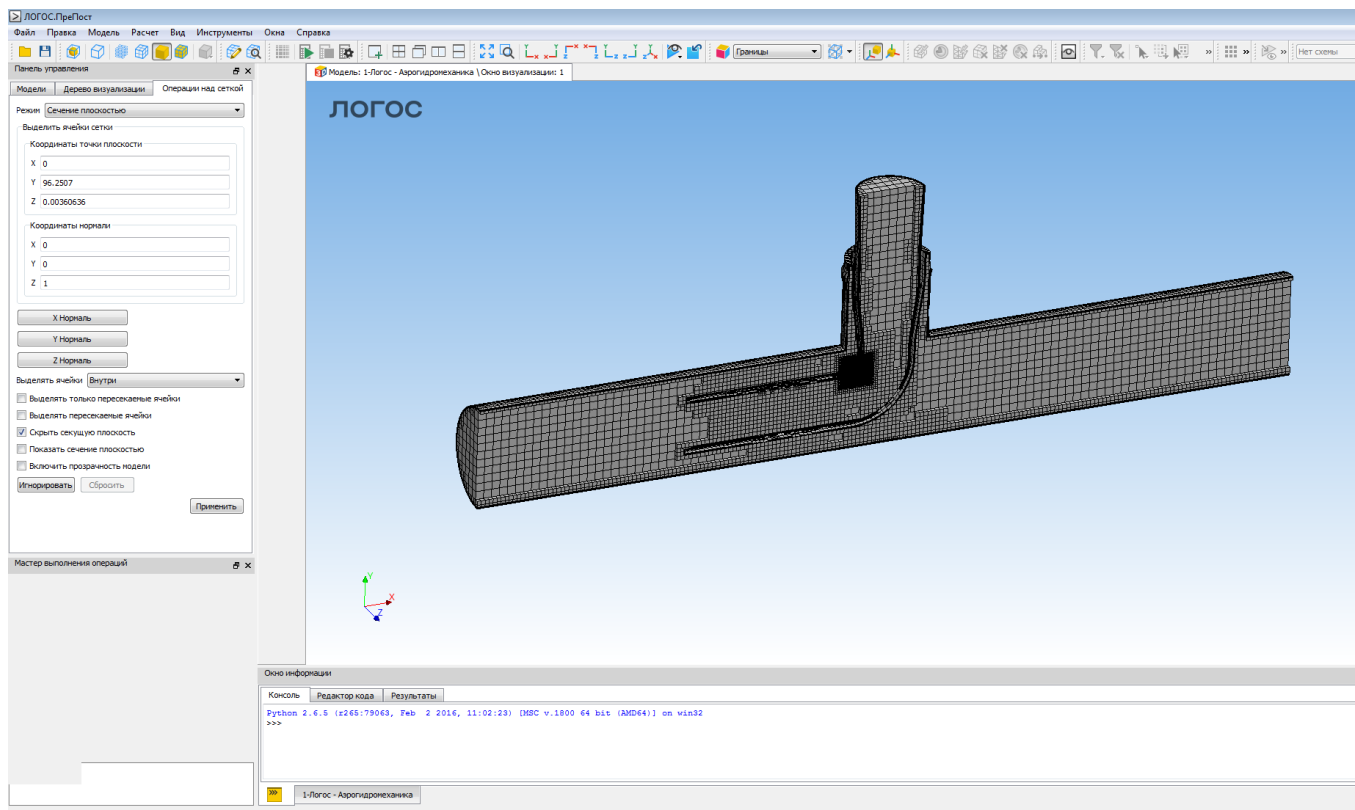


Рисунок 3 - Пример расчетной сетки в программном комплексе ЛОГОС

CAE системы могут быть ориентированы на проведение анализа при помощи моделирования различных физических процессов, таких как тепломассоперенос, прочностной анализ, кинематика тела и др. Результатом анализа может стать изменение в конструкции разрабатываемых изделий, или например, материала изготовления. Примеры CAE систем представлены в таблице 2.

Таблица 2 - Примеры CAE систем

Зарубежные	Отечественные
ANSYS	ПК ЛОГОС
Abaqus	Fidesys
HyperWorks (HyperMesh)	WIN Machine
NX NASTRAN	FlowVision

Все системы автоматизированного проектирования объединяет одно – их работа зависит от геометрического (математического) ядра, которое является основой программных комплексов представленных в таблице 1-2. Основные назначения геометрических ядер и их разновидности, рассмотрим в следующей главе.

1.3. Геометрическое (математическое) ядро

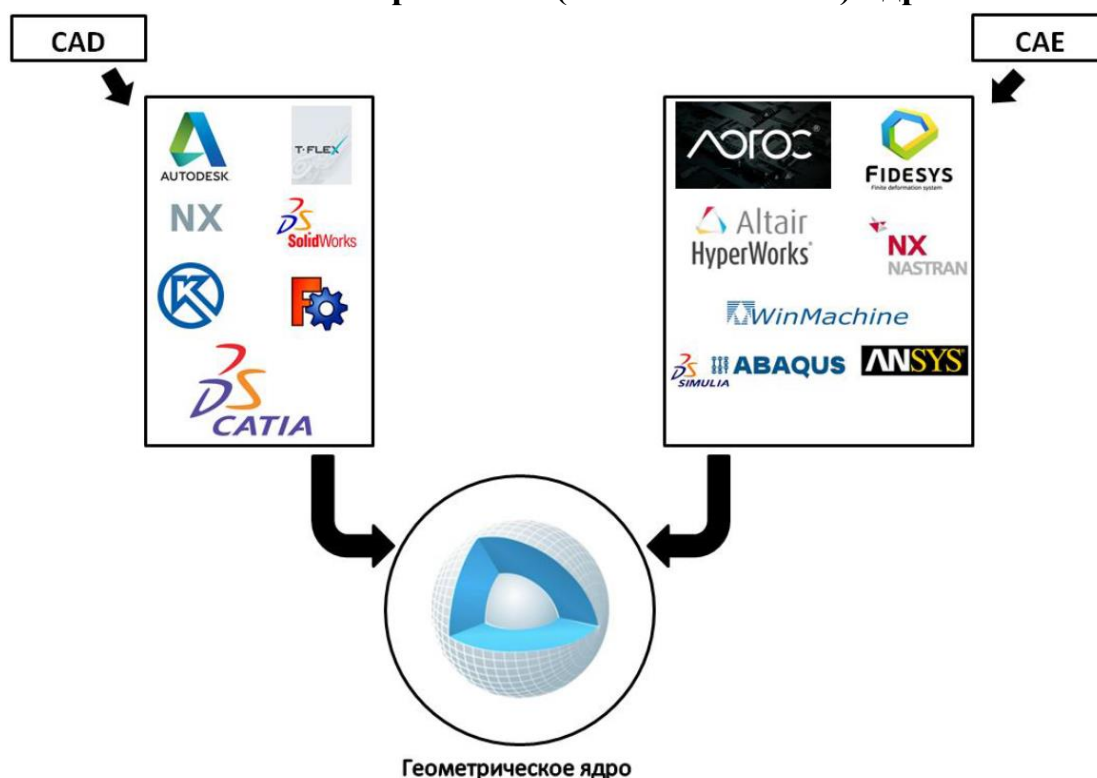


Рисунок 4 - Общий модуль CAD/CAE – систем.

Геометрическое ядро трехмерного моделирования — это программная компонента, предназначенная для использования в качестве базового инструментального средства при разработке программных систем, связанных с компьютерным моделированием трёхмерных объектов [13].

С точки зрения программиста геометрическое ядро — это библиотека функций/классов для создания геометрических объектов (точка, кривая, поверхность, твердое тело), их редактирования, визуализации модели и обмена трехмерными данными с другими программами.

Из истории создания геометрических ядер следует, что в разработке инженерного программного обеспечения это одна из самых трудных задач. Решение такой задачи требует сочетания высоких компетенций в математике, проектировании программного обеспечения и т.д.

1.3.1. Основные типы геометрических ядер для создания CAD/CAE систем

Существующие, на сегодняшний день, ядра сложные наукоемкие программные продукты, разрабатывавшиеся в течение 15-30 (и более) лет и активно

использующиеся в разработке прикладного программного обеспечения. Наиболее известные геометрические ядра:

- **Parasolid**

Parasolid (разработчик: Siemens PLM Software) — это ядро геометрического моделирования. Данный продукт разрабатывается и поддерживается компанией Siemens PLM Software. Компания Siemens PLM Software применяет Parasolid в собственных системах Solid Edge, Femap, NX, а также использует ядро в коммерческих целях, предоставляя лицензии независимым поставщикам программного обеспечения и конечным пользователям, таким как Altair HyperWorks, Abaqus и др.

- **ACIS**

3D ACIS Modeler (ACIS) - это ядро геометрического моделирования, разработанное Spatial Corporation (ранее Spatial Technology). Данное ядро является коммерческим продуктом, который используют в своих разработках такие компании как Autodesk, Bricsys, и др.

Одной из важных разработок является 3D Spatial InterOp – инструментарий для разработки программного обеспечения, используемый для трансляции 3D данных в различных системах.

- **RGK**

RGK (Russian Geometric Kernel) - данный продукт разработан в рамках проекта «Разработка отечественного лицензируемого 3D-ядра» по заказу Министерства промышленности и торговли Российской Федерации [13].

Преимуществом разработки такой системы для государства, является то, что данный продукт будет некоммерческим (все затраты по проекту идут на разработку системы, а не на закупку дорогостоящих лицензий). На данный момент ядро нуждается в доработке.

- **C3D Modeler (C3D Labs, дочернее предприятие АСКОН, Россия)**

В 1995 году компанией АСКОН разработала собственное ядро трехмерного моделирования для создания САПР КОМПАС-3D. Это было сделано для того, чтобы не платить за дорогостоящие лицензионные продукты зарубежных производителей в сфере создания геометрических ядер. Программа КОМПАС-3D была выпущена для коммерческого пользования в 2000 году. В 2012 году C3D стало коммерческим продуктом (до этого использовалось только в составе собственных разработок). Лицензии данного ядра используются для создания собственной программной продукции такими компаниями как: НТЦ АПМ, Нанософт, SolarTech, Эремикс и др.

- **Open CASCADE Technology (OCCT) (OPEN CASCADE S.A.S,**

Франция)

Данное ядро было создано в 1993 году фирмой Matra Datavision и имело название CAS.CADE (Computer Aided Software for Computer Aided Design and Engineering). Изначально CAS.CADE рассматривалась как набор программных инструментов для внутреннего использования, которая позволяла разработчикам ПО применять объектно-ориентированные технологии [14]. В 1999 году исходный код CAS.CADE был опубликован в сети интернет и получил название Open CASCADE. Сделано это для того, чтобы совершенствовать код данного ядра могли все желающие независимые разработчики. Программное обеспечение созданное на основе ядра Open CASCADE: SALOME, FreeCAD.

1.3.2. ПК ЛОГОС на основе 3DTV

В программном комплексе ЛОГОС используется геометрическое ядро 3DTV. Кроме привычных функциональных возможностей по работе с геометрией данное математическое ядро предоставляет инструментарий, обеспечивающий полную диагностику конструкторских моделей и включающий весь необходимый набор инструментов для быстрого и эффективного исправления геометрических моделей. Процесс диагностики и исправления высоко автоматизирован и может быть применен как к отдельным телам, так и к сборкам различной степени сложности. Исправление всегда осуществляется в рамках точности модели, что обеспечивает исходную целостность представления и исключает какие-либо деформации [12].

Средства проверки могут быть нацелены как на получение качественной геометрии (при трансляции данных), так и на приведение конструкторской модели в соответствие с существующими международными стандартами, такими как VDA 4955, JAMA и AIAG D-15, или задаваемыми пользователями критериям.

Геометрическое ядро 3DTV ориентировано на работу с геометрическими данными, как в граничном представлении (B-rep), так и в фасеточном (Faceted). Типы геометрических представлений рассмотрим в следующей главе.

1.4. Типы геометрических представлений

1.4.1. B-rep

Наиболее общий подход к описанию тел состоит в представлении тела совокупностью ограничивающих его объем оболочек, грани и ребра которых заданы параметрически. Каждая оболочка строится из набора стыкующихся друг с другом поверхностей произвольной формы, содержащих полную информацию о своих границах и связях с соседями. Такое описание тел называется граничным

представлением (*Bounded representation* или *B-rep*). Оно дает возможность выполнить над телами множество операций, сохраняя при этом единый способ их внутреннего устройства. Представление тел с помощью границ позволяет моделировать объекты произвольной формы и сложности.

Все перечисленные подходы в описании тел используют топологические объекты и удовлетворяют условиям связности, ориентируемости и замкнутости.

B-rep - ориентированные алгоритмы включают обычно следующие шаги:

- генерирование трехмерных вершин;
- создание ребер по полученным координатам вершин;
- создание граней из ребер;
- формирование трехмерного объекта из граней [1].

1.4.2. CSG

Конструктивная твердотельная геометрия (*Constructive Solid Geometry* или *CSG*) оперирует примитивами, к которым, как правило, относят прямоугольную призму, треугольную призму, сферу, цилиндр, конус и тор. Над примитивами и полученными из них телами можно выполнять различные операции (в первую очередь булевы операции). Используемые конструктивной твердотельной геометрией поверхности (сферическая, цилиндрическая, коническая, поверхность тора и плоскость) делят пространство на две части и для них можно указать, с какой стороны поверхности находится внутренний объем тела.

1.4.3. Faceted

Фасеточная геометрия описывает поверхность тела с помощью набора треугольников таким образом, чтобы отклонение полученной поверхности от исходной было не больше заданного значения. В качестве критерия оценки отклонения используются различные подходы:

- абсолютное – расстояние между поверхностями;
- относительное – отклонение от поверхности на единицу длины/площади, потеря объема, и т. д.

Преимущества фасеточного представления геометрии:

- простейший формат представления и хранения – для полного описания поверхности достаточно массива треугольников с координатами вершин;
- алгоритмы по обработке данной геометрии: проекция точки на поверхность, поиск кратчайшего пути по поверхности, пересечение поверхностей и т. д.

Недостатки:

- потеря точности: безвозвратное огрубление криволинейных поверхностей,
- в большинстве случаев – потеря или отсутствие информации о характерных элементах геометрии: ребер, граней, фрагментов, в связи с чем появляется необходимость ее восстановления.

Несмотря на недостатки, фасеточное представление геометрии незаменимо при решении множества практических задач:

- восстановление поверхностей тел по данным, полученным в результате сканирования реального объекта;
- построение оболочки для облака точек [2][17].

Большинство CAD/CAE систем используют граничное представление, т.к. по сравнению с представлением в виде конструктивной твердотельной геометрии (CSG), которая использует только примитивные объекты и булевы операции для их объединения, граничное представление является более гибким и имеет богатый набор операций. Это делает граничное представление самым подходящим выбором при разработке CAD/CAE систем.

Математическое моделирование геометрических объектов в граничном представлении рассмотрим в следующей главе.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Геометрическое моделирование

Геометрическое моделирование изучает методы построения математической модели, описывающей геометрические свойства объектов. Оно базируется на аналитической и дифференциальной геометрии, вычислительной математике, вариационном исчислении, топологии и разрабатывает собственные математические методы моделирования.

Инструментом для геометрического моделирования служат математические методы решения тех или иных задач. Используемые методы позволяют описывать геометрические свойства предметов, создавать их математические модели и исследовать их путем проведения различных расчетов и численных экспериментов.

Для описания геометрических свойств окружающих предметов, строятся различные твердые тела. Тела в свою очередь можно описать точками, линиями и поверхностями. Все они обладают определенными общими свойствами, поэтому ими можно оперировать как объектами [1].

Геометрические объекты

2.2. Точка

Самым простым объектом геометрического моделирования является точка. Точка в R пространстве описывается координатами u^1, u^2, u^3 некоторой системы координат. В декартовой системе координат точку можно описать с помощью радиус-вектора $r = [r_1 \ r_2 \ r_3]^T$. Радиус-вектор определяет преобразование переноса, переводящее начальную точку декартовой системы координат в заданную точку пространства. Компоненты радиус-вектора равны ее координатам.

2.3. Линии. Геометрия кривой

Кривая линия или просто кривая называется геометрическое место точек, координаты которых описываются непрерывными и однозначными функциями $r_1(t), r_2(t), r_3(t)$ параметра t , принимающего значение на отрезке $t_{min} \leq t \leq t_{max}$. В декартовой системе координат $Oe_1e_2e_3$ кривую можно описать радиус-вектором:

$$r(t) = \sum_{i=1}^3 r_i(t)e_i, \quad t_{min} \leq t \leq t_{max} \quad (1)$$

Кривая, представленная в виде (1), называется параметрической. Предполагаем, что координатные функции $r_i(t)$ имеют непрерывные производные

до любого порядка. Точка кривой $r(t)$ будет называться обыкновенной, если в этой точке не обращается в нуль длина вектора первой производной по параметру $r' = dr/dt$. В противном случае точка будет называться особой. Параметр t является внутренней координатой, от которой зависит положение точки кривой.

Координаты радиус-вектора линий могут быть представлены как аналитические функции некоторого параметра t . Простейшими из них могут являться канонические сечения и некоторые линии, полученные как траектория движения точек механизмов. К некоторым примерам таких линий можно отнести:

Прямая линия - одна из простейших линий. Она может быть описана точкой p и вектором a . Радиус-вектор прямой линии определим зависимостью

$$r(t) = p + ta, \quad -\infty < t < +\infty \quad (2).$$

Отрезок. Другой простейшей линией является отрезок прямой. Он может быть представлен через начальную p_1 и конечную p_2 точки:

$$r(t) = (1 - t)p_1 + tp_2, \quad 0 \leq t \leq 1$$

Введем обозначение $v = (1 - t)$, тогда радиус-вектор отрезка будет описываться формулой $r(t) = vp_1 + tp_2$, $0 \leq t \leq 1$, $0 \leq v \leq 1$, $t + v = 1$.

Величины v и t , на которые умножаются координаты p_1 и p_2 , являются барицентрическими координатами. С использованием барицентрических координат точек p_1 и p_2 формула отрезка приобретает симметричный вид.

Плоские кривые. Описанию плоских кривых в пространстве можно придать следующий вид:

$$r(t) = p + x(t)i_x + y(t)i_y,$$

где p – некоторая точка привязки характерной точки кривой i_x и i_y – не коллинеарные векторы, $x(t)$ и $y(t)$ – скалярные функции. Описание этих кривых в двухмерном пространстве имеет аналогичный вид. Разница будет заключаться лишь в том, что в двухмерном пространстве линии описываются точками и векторами, имеющими две компоненты, а в трехмерном пространстве имеющие три компоненты.

Кривые второго порядка

Кривая второго порядка — геометрическое место точек плоскости, прямоугольные координаты которых удовлетворяют уравнению вида:

$$C_{11}x^2 + C_{22}y^2 + 2C_{12}xy + C_{13}x + C_{23}y + C_{33} = 0$$

в котором по крайней мере один из коэффициентов C_{11}, C_{22}, C_{12} отличен от нуля. Кривые второго порядка могут быть невырожденными (эллипс, гипербола,

мнимый эллипс, парабола) или вырожденными (вещественная точка на пересечении двух мнимых прямых, пара вещественных пересекающихся прямых, вырожденная парабола и др.). Они могут возникать при выполнении тех или иных условий равенства инвариантов.

2.3.1. Кривые Безье

Координаты сплайнов являются полиномами параметра кривой. В общем случае сегменты таких кривых описываются векторными функциями вида:

$$r(t) = \sum_{i=0}^n t^i a_i. \quad (3)$$

Компоненты вектора a_i являются коэффициентами полиномов для компонент результирующего вектора. Векторы не несут геометрический смысл каждый по отдельности, но если провести перегруппировку правой части, то можно получить выражение, в котором векторные величины будут иметь определенный геометрический смысл. Это можно наблюдать на примере интерполяционной формулы Лагранжа, где радиус-вектор кривой представлен в виде:

$$r(t) = \sum_{i=0}^n L_i(t) P_i. \quad (4)$$

Векторные коэффициенты P_i являются радиус-векторами характеристических точек кривой, а зависимость кривой от параметра сосредотачивалась в скалярных функциях $L_i(t)$, индивидуальных для каждой характеристической точки P_i . Аналогичный подход для построения кривых был предложен французскими математиками П. Де Кастелье и П. Безье независимо друг от друга. Данные кривые, впоследствии, получили название «кривые Безье».

Кривая Безье описывается формулой:

$$r(t) = \sum_{i=0}^n B_i^n(t) P_i = \sum_{i=0}^n \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} P_i, \quad 0 \leq t \leq 1, \quad (5)$$

где P_i – радиус-векторы точек, а n – число этих точек минус единица. Совокупность функций $B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = C_n^i t^i (1-t)^{n-i}$ называется базисом Бернштейна. Можно сказать, что кривая Безье является полиномом Бернштейна с векторными коэффициентами.

2.3.2. Сплайн

Во многих практических задачах требуется построить плавную кривую линию, проходящую через заданные точки. Для этих целей и нужны сплайны. Данный термин для кривых линий, взял свое название от чертежного инструмента – упругой гибкой линейки, которая способна изгибаться так, чтобы проходить через заданные точки. Линии, построенные по точкам можно назвать точечно-заданными. Простейшей точечно-заданной линией является ломаная линия. Она состоит из отрезков, последовательно соединяющих заданные точки. Значение параметра в каждой заданной точке должно быть больше значения параметра в предыдущей точке $t_i < t_{i+1}$. Радиус-вектор ломаной определяется равенством:

$$\begin{aligned} r(t) &= P_i(1 - w) + P_{i+1}w, \\ w &= \frac{t - t_i}{t_{i+1} - t_i}, \quad t_0 \leq t \leq t_n, \end{aligned} \quad (6)$$

где t_i – точки на числовой оси (узлы), P_i – точки, определяющие линию и являющиеся ее внутренними данными (характеристические точки кривой), значения параметра t_i (узловое), параметр w будем называть местным параметром на участке кривой между точками P_i и P_{i+1} .

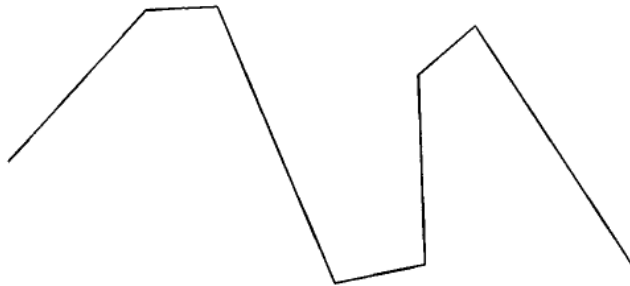


Рисунок 5 - Ломаная линия

Ломаная линия может быть замкнутой, в этом случае первая характеристическая точка одновременно является и последней. Параметрическая длина ломаной линии равна числу точек, на которых она задана. К свойствам ломаной линии можно отнести: легко вычислить точки, легко редактировать (добавить или убрать точку, сдвинуть имеющуюся), можно разрезать на части, каждая из которых также будет являться ломаной.

По назначению можно выделить три основные группы сплайнов:

- интерполяционные сплайны или функциональные сплайны — проходящие точно через заданные точки;
- сглаживающие сплайны — проходящие через заданные точки с учётом погрешностей их определения;

- корреляционные сплайны — проходящие через корреляционное множество точек и отображающие его генеральную зависимость (тренд, регрессию).

Интерполяционные и функциональные сплайны используют в задачах геометрического моделирования.

Существуют различные сплайны: сплайн Эрмита, кубический сплайн, сплайн Лагранжа, сплайн Ньютона и др. Рассмотрим подробно кубический сплайн, так как именно сплайны данного вида применяются при работе в программе.

Кубический сплайн. Построим на заданной совокупности характеристических точек сплайн, который бы имел непрерывными первые и вторые производные радиус-вектора. На каждом участке между соседними характеристическими точками будем описывать радиус-вектор кривой кубическим полиномом типа:

$$r_i(w) = a_0 + a_1 w + a_2 w^2 + a_3 w^3$$

$$w = \frac{t - t_i}{t_{i+1} - t_i}, \quad 0 \leq w \leq 1. \quad (7)$$

Обозначим вторые производные как $r''(t_i) = s_i$. Вторая производная радиус-вектора на участке $t_i \leq t \leq t_{i+1}$ является линейной функцией параметра t

$$\frac{d^2 r}{dt^2} = s_i \frac{t_{i+1} - t}{t_{i+1} - t_i} + s_{i+1} \frac{t - t_i}{t_{i+1} - t_i}.$$

После двукратного интегрирования получим:

$$r(t) = s_i \frac{(t_{i+1} - t)^3}{6(t_{i+1} - t_i)} + s_{i+1} \frac{(t - t_i)^3}{6(t_{i+1} - t_i)} + c_1 t + c_2. \quad (8)$$

Постоянные интегрирования c_1 и c_2 определим из условий на концах участка $r(t_i) = p_i$ и $r(t_{i+1}) = p_{i+1}$. После вычислений получим:

$$r(t) = s_i \frac{(t_{i+1} - t)^3}{6(t_{i+1} - t_i)} + s_{i+1} \frac{(t - t_i)^3}{6(t_{i+1} - t_i)} + \left(\frac{p_i}{t_{i+1} - t_i} - s_i \frac{t_{i+1} - t_i}{6} \right) (t_{i+1} - t) + \left(\frac{p_{i+1}}{t_{i+1} - t_i} - s_{i+1} \frac{t_{i+1} - t_i}{6} \right) (t - t_i). \quad (9)$$

Выражение (9) описывает кубический полином на отрезке $t_i \leq t \leq t_{i+1}$ и содержит две неизвестные величины s_i и s_{i+1} . Для их определения приравняем первую производную сплайна на правом конце отрезка $t_{i-1} \leq t \leq t_i$ первой производной сплайна на левом конце отрезка $t_i \leq t \leq t_{i+1}$. После дифференцирования (9) и подстановки $t = t_i$ получим:

$$\frac{dr}{dt}|_{t_i} = -\frac{(2s_i + s_{i+1})(t_{i+1} - t_i)}{6} + \frac{p_{i+1} - p_i}{t_{i+1} - t_i} \quad (10)$$

После замены в (9) i на $i - 1$ дифференцирования и ставки $t = t_i$ получим:

$$\frac{dr}{dt}|_{t_i} = -\frac{(2s_i + s_{i-1})(t_i - t_{i-1})}{6} + \frac{p_i - p_{i-1}}{t_i - t_{i-1}}. \quad (11)$$

Таких уравнений можно составить $n - 1$ для характеристических точек (в случае незамкнутого сплайна). Подлежат определению $n + 1$ неизвестных векторов s_i . Два недостающих уравнения для определения всех неизвестных составим, исходя из условий на концах кривой. В некоторых случаях можно принять $s_0 = s_1, s_n = s_{n-1}$, тогда на концевых отрезках сплайн будет иметь постоянную кривизну. Дополнительные условия зависят от физических или геометрических условий в каждом конкретном случае. Если требуется построить замкнутую кривую, то мы будем иметь равное число неизвестных и уравнений. Таким образом, неизвестные векторы вторых производных s_i в характеристических точках найдем из системы линейных уравнений. Матрица этой системы линейных уравнений трехдиагональная, что значительно упрощает решение системы. После вычисления векторов s_i сплайн будет описываться формулой:

$$r(t) = (1 - w)p_i + wp_{i+1} + \left((-2w + 3w^2 - w^3)s_i + (-w + w^3)s_{i+1}\right) \frac{(t_{i+1} - t_i)^2}{6}, \quad (12)$$

где $w = \frac{t - t_i}{t_{i+1} - t_i}$, а $t_i \leq t \leq t_{i+1}$. Полученная кривая называется кубическим сплайном.

2.3.3. В-сплайн

Существует несколько различных определений и обозначений В-сплайна. Вводятся такие определения как ненормированный В-сплайн и нормированный В-сплайн.

Ненормированный В-сплайн для неубывающей последовательности m узлов есть разделенная разность m -ого порядка усеченной степенной функции со смещенным началом $\sigma_{m-1}(z) = (z - 1)^{m-1}$.

Нормированный В-сплайн для неубывающей последовательности m узлов есть разделенная разность m -ого порядка усеченной степени функции со смещенным началом $\sigma_{m-1}(z) = (z - t)^{m-1}$.

Ненормированный и нормированный В-сплайны связаны между собой только коэффициентами и с точностью до множителя их определения одинаковы.

2.3.4. NURBS кривые

В-сплайны используют для построения рациональных кривых линий в форме:

$$r(t) = \frac{\sum_{i=1}^n N_i(t)w_i p_i}{\sum_{i=1}^n N_i(t)w_i}, \quad t_{min} \leq t \leq t_{max}, \quad (13)$$

где n – число вершин, p_i – радиус-вектор i -й вершины характеристической ломаной кривой, w_i – вес i -й вершины, $N_i(t)$ – функция параметра кривой, дающая вклад i -й вершины в общую сумму при данном t .

Формула для вычисления радиус-вектора рациональной параметрической кривой на основе В-сплайнов $N_{i,m}(t)$, построенной по вершинам p_i , $i = 1, 2, \dots, n$ ($n \geq m$), обладающим весами w_i имеет вид:

$$r(t) = \frac{\sum_{i=1}^n N_{i,m}(t)w_i p_i}{\sum_{i=1}^n N_{i,m}(t)w_i}, \quad t_{min} \leq t \leq t_{max}. \quad (14)$$

Радиус-вектор кривой равен частному от деления некоторого вектора на вес кривой в данной точке. Каждый из В-сплайнов i -порядка $N_{i,m}(t)$ построен на последовательности из $m + 1$ узлов $t_i, t_{i+1}, t_{i+2}, \dots, t_{i+m}$. Значение узлов должны образовывать неубывающую последовательность, а в остальном на них не накладывается никаких ограничений. Кривые линии, построенные на множестве не равноотстоящих узлов, называются неоднородными. Параметрическое расстояние между соседними узлами у неоднородных кривых меняется при переходе от узла к узлу. Кривая (14) в общем случае является неоднородной и рациональной. Кривая (14) носит имя NURBS (*Non-uniform Rational B-spline*) кривой.

Когда все вершины NURBS кривой имеют равные веса, то формула (14) для вычисления радиус-вектора кривой на основе В-сплайнов принимает вид:

$$r(t) = \sum_{i=1}^n N_{i,m}(t)p_i, \quad t_{min} \leq t \leq t_{max} \quad (15)$$

Эта кривая также может быть использована в моделировании, так как она обладает определенным порядком гладкости, достаточно легко вычислить и легко управляется своими вершинами.

В формуле (14) для построения совокупности из n В-сплайнов i -го порядка требуется $n + m$ узлов в случае незамкнутой кривой и $n + 2m$ узлов в случае замкнутой кривой. Число узлов всегда больше числа вершин, поэтому множество узлов называется расширенным. Пусть вершины В-сплайна (14) пронумерованы от 1 до n . Пронумеруем узлы, на которых построены В-сплайны от 1 до $n + m$. Чтобы незамкнутая NURBS кривая проходила через первую и последнюю вершину, первый В-сплайн должен иметь кратными первые m узлов (из $m + 1$ узлов, на которых он

строится), а последний В-сплайн должен иметь кратными последние m узлов (из $m + 1$ узлов, на которых он строится). Для построения незамкнутой $(m - 2)$ – раз дифференцируемой NURBS кривой первые m узлов должны иметь равные значения: $t_1 = t_2 = \dots = t_m$, следующие $n - m$ узлов должны образовывать возрастающую последовательность: $t_{m+i} < t_{m+i+1}$ ($i = 1, 2, \dots, n - m$), оставшиеся m узлов должны принимать равные значения: $t_{n+1} = t_{n+2} = \dots = t_{n+m}$. Для построения замкнутой $(m - 2)$ – раз дифференцируемой NURBS кривой последовательность узлов должна отражать замкнутость: значения первых $n + m$ узлов должны образовать возрастающую последовательность: $t_i < t_{i+1}$ ($i = 1, 2, \dots, n + m$), следующие m узлов должны идти через интервалы, повторяющие первые m интервалов между узлами: $t_{i+n+m} = t_{n+m} + t_i$ ($i = 1, 2, \dots, m$). В общем случае параметр NURBS кривой изменяется от значения узла $t_{min} = t_m$ до значения узла $t_{max} = t_{n+m}$. Построенная на таких последовательностях узлов кривая будет иметь непрерывные производные до $(m - 2)$ -го порядка включительно на всей области определения.

Вычисление радиус-вектора кривой. Для вычисления радиус-вектора NURBS кривой используется следующая схема. По значению параметра t из условия $t_i \leq t \leq t_{i+1}$ определяется номер i от нуля В-сплайна первого порядка и вычисляется его значение, исходя из определения:

$$M_{i,1}(t) = \begin{cases} \frac{1}{t_{i+1} - t}, & \text{если } t_i \leq t \leq t_{i+1} \\ 0, & \text{если } t < t_i \text{ или } t \geq t_{i+1} \end{cases} \quad (16)$$

Далее, используя рекуррентное соотношение Кокса-Де Бура

$$M_{i,m}(t) = \frac{(t_{i+m} - t)M_{i+1,m-1}(t) + (t - t_i)M_{i,m-1}(t)}{t_{i+m} - t_i} \quad (17)$$

последовательно вычисляются все отличные от нуля при данном параметре t В-сплайны до m -го порядка включительно: $M_{i-m+1,m}(t), M_{i-m+2,m}(t), \dots, M_{i,m}(t)$:

$$\begin{array}{r} M_{i,1}(t) \\ M_{i-1,2}(t) \quad M_{i,2}(t) \\ M_{i-2,3}(t) \quad M_{i-1,3}(t) \quad M_{i,3}(t) \\ \dots \end{array}$$

$$\begin{array}{cccccc}
M_{i-m+2,m-1}(t) & \dots & M_{i-2,m-1}(t) & M_{i-1,m-1}(t) & M_{i,m-1}(t) & \\
M_{i-m+1,m}(t) & M_{i-m+2,m}(t) & \dots & M_{i-2,m}(t) & M_{i-1,m}(t) & M_{i,m}(t)
\end{array}$$

Данная треугольная таблица вычисляется построчно, поскольку (16) известен элемент первой строки, а каждый элемент следующей строки можно построить по двум соседним элементам в предыдущей строке с помощью (17). При вычислении крайних элементов каждой строки используется тот факт, что один из элементов в предыдущей строке равен нулю. Далее В-сплайны m -го порядка нормируются

$N_{j,m}(t) = (t_{j+m} - t_j)M_{j,m}(t)$, $j = i - m + 1, i - m + 2, \dots, i$, и подставляются в формулу (14), которая примет вид

$$r(t) = \frac{\sum_{j=1-m+1}^i N_{j,m}(t)w_j p_j}{\sum_{j=1-m+1}^i N_{j,m}(t)w_j}. \quad (18)$$

Одновременно с вычислением В-сплайнов может проводиться вычисление их производных. Формулы (16) и (17) можно считать определением В-сплайна.

Свойства кривой. Каждый В-сплайн отличен от нуля только на части области изменения параметра кривой, поэтому при изменении радиус-вектора одной из вершин подлежит пересчету не вся кривая, а только ее часть. В-сплайны являются неотрицательными функциями. Сумма значений всех нормированных В-сплайнов при любом параметре t NURBS кривой равна единице: $\sum_i N_{i,m}(t) = 1$, а площадь под любым ненормированным В-сплайном удовлетворяет равенству

$$\int_{-\infty}^{+\infty} M_{i,m}(t)dt = \int_{t_i}^{t_{i+m}} M_{i,m}(t)dt = \frac{1}{m}. \quad (19)$$

Из данного равенства следуют:

$$\begin{aligned}
\int_{-\infty}^{+\infty} N_{i,m}(t)dt &= (t_{i+m} - t_i) \int_{t_i}^{t_{i+m}} M_{i,m}(t)dt = \frac{t_{i+m} - t_i}{m}, \\
\int_{-\infty}^{+\infty} \left(\sum_{i=1}^n N_{i,m}(t)w_i p_i \right) dt &= \frac{1}{m} \sum_{i=1}^n (t_{i+m} - t_i)w_i p_i,
\end{aligned} \quad (20)$$

$$\int_{-\infty}^{+\infty} \left(\sum_{i=1}^n N_{i,m}(t) w_i \right) dt = \frac{1}{m} \sum_{i=1}^n (t_{i+m} - t_i) w_i.$$

NURBS кривая (14) обладает достаточной гибкостью. Ее можно легко редактировать путем изменения положения вершин. Например, легко можно модифицировать имеющуюся кривую так, чтобы она проходила через указанную точку или чтобы она касалась указанной кривой в заданной точке.

Кривая Безье как частный случай NURBS кривой. Нормированные В-сплайны $N^n_i(t)$ в частном случае последовательности узлов $t_{-1-n} = t_{-n} = t_{1-n} = \dots = t_{-1} = 0$, $t_0 = t_1 = \dots = t_n = 1$ совпадает с базисными функциями Бернштейна $B^n_i(t)$, а NURBS кривая (1), записанная в виде

$$r(t) = \frac{\sum_{i=1}^n N^n_i(t) w_i p_i}{\sum_{i=1}^n N^n_i(t) w_i}, \quad t_{min} \leq t \leq t_{max}, \quad (21)$$

Совпадает с рациональной кривой Безье. Это происходит, когда число вершин на единицу больше порядка В-сплайна $N^n_i(t)$. Область изменения параметра NURBS кривой в этом случае равна $0 \leq t \leq 1$.

Равенства В-сплайнов $N^n_i(t)$ построенных на последовательности узлов $t_{-1-n} = t_{-n} = t_{1-n} = \dots = t_{-1} = 0$, $t_0 = t_1 = \dots = t_n = 1$ и коэффициентов Бернштейна $B^n_i(t)$ можно доказать следующим образом.

Коэффициенты Бернштейна связаны рекуррентным соотношением

$$B^n_i(t) = t B^{n-1}_{i-1}(t) + (1-t) B^{n-1}_i(t), \quad i = 0, 1, 2, \dots, n.$$

Получим все коэффициенты Бернштейна, начав вычисление с $B^0_0(t) = 1$. В-сплайны $N^n_i(t)$ связаны рекуррентным соотношением

$$\begin{aligned} N^n_i(t) &= \frac{t_i - t}{t_i - t_{i-n}} N^{n-1}_i(t) + \frac{t - t_{i-n-1}}{t_i - t_{i-n-1}} N^{n-1}_{i-1}(t) \\ &= (1-t) N^{n-1}_i(t) + t N^{n-1}_{i-1}(t), \quad i = 0, 1, 2, \dots, n, \end{aligned} \quad (22)$$

где только $N^0_0(t) = 1$, а остальные В-сплайны нулевого порядка равны нулю. Кривые Безье являются частным случаем NURBS кривой.

NURBS представление кубического сплайна. Пусть построен незамкнутый кубический сплайн $a(x)$, проходящий через точки a_j , при значениях параметра x_j , $j = 1, 2, \dots, n$. Построим NURBS кривую $r(t) = \sum_{i=1}^{n+2} N_{i,4}(t) p_i$ четвертого порядка ($m = 4$) на последовательности узлов $t_1 = t_2 = t_3 = t_4 = x_1$, $t_5 = x_2$, $t_6 = x_3$, $\dots$, $t_{n+m-1} = t_{n+m} = t_{n+m+1} = t_{n+m+2} = x_n$. Для этого найдем $n+2$ вершины p_i , $i = 1, 2, \dots, n+2$ из условий:

$$\begin{aligned} r(x_1) &= a_1, \quad \left. \frac{da}{dx} \right|_{x=x_1} = a'_1, \\ r(x_j) &= a_j, \quad j = 2, 3, \dots, n-1, \\ \left. \frac{da}{dx} \right|_{x=x_n} &= a'_n, \quad r(x_n) = a_n. \end{aligned} \quad (23)$$

Веса вершины p_i равны единице. Кубический сплайн $a(x)$ и NURBS кривая $r(t)$ будут полностью совпадать, так как они состоят из гладко стыкующихся (до второй производной включительно) участков кривых третьей степени. Таким образом, вершины p_i , $i = 1, 2, \dots, n + 2$ мы найдем из системы (24) $n + 2$ уравнений:

$$\begin{aligned}
& N_{1,4}(x_1)p_1 + N_{2,4}(x_1)p_2 + \dots + N_{n+1,4}(x_1)p_{n+1} + N_{n+2,4}(x_1)p_{n+2} = a_1, \\
& 3N_{2,3}(x_1)\frac{p_2 - p_1}{t_5 - t_2} + 3N_{3,3}(x_1)\frac{p_3 - p_2}{t_6 - t_3} + \dots + 3N_{n+2,3}(x_1)\frac{p_{n+2} - p_{n+1}}{t_{n+5} - t_{n+2}} = a'_1, \\
& N_{1,4}(x_2)p_1 + N_{2,4}(x_2)p_2 + \dots + N_{n+1,4}(x_2)p_{n+1} + N_{n+2,4}(x_2)p_{n+2} = a_2, \\
& N_{1,4}(x_3)p_1 + N_{2,4}(x_3)p_2 + \dots + N_{n+1,4}(x_3)p_{n+1} + N_{n+2,4}(x_3)p_{n+2} = a_3, \\
& \\
& N_{1,4}(x_{n-1})p_1 + N_{2,4}(x_{n-1})p_2 + \dots + N_{n+1,4}(x_{n-1})p_{n+1} + \\
& \qquad \qquad \qquad + N_{n+2,4}(x_{n-1})p_{n+2} = a_{n-1}, \\
& 3N_{2,3}(x_n)\frac{p_2 - p_1}{t_5 - t_2} + 3N_{3,3}(x_n)\frac{p_3 - p_2}{t_6 - t_3} + \dots + 3N_{n+2,3}(x_n)\frac{p_{n+2} - p_{n+1}}{t_{n+5} - t_{n+2}} = a'_n, \\
& N_{1,4}(x_n)p_1 + N_{2,4}(x_n)p_2 + \dots + N_{n+1,4}(x_n)p_{n+1} + N_{n+2,4}(x_n)p_{n+2} = a_n.
\end{aligned} \tag{24}$$

Используя формулы (16) и (17), определим, что для каждого значения параметра $t = t_i$, $i = 4, 5, \dots, n + m - 1$, отличны от нуля только три В-сплайна четвертого порядка:

$$\begin{aligned} N_{i-3,4}(t_i) &= \frac{t_{i+1} - t_i}{t_{i+1} - t_{i-2}} \frac{t_{i+1} - t_i}{t_{i+1} - t_{i-1}}, \\ N_{i-2,4}(t_i) &= \frac{t_i - t_{i-1}}{t_{i+2} - t_{i-1}} \frac{t_{i+2} - t_i}{t_{i+1} - t_{i-1}} + \frac{t_{i+1} - t_i}{t_{i+1} - t_{i-2}} \frac{t_i - t_{i-2}}{t_{i+1} - t_{i-1}} \end{aligned} \quad (25)$$

$$N_{i-1,3}(t_i) = \frac{t_i - t_{i-1}}{t_{i+2} - t_{i-1}} \frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}}$$

и два В-сплайна третьего порядка:

$$N_{i-2,3}(t_i) = \frac{t_{i+1} - t_i}{t_{i+1} - t_{i-1}}, \quad N_{i-1,3}(t_i) = \frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}} \quad (26)$$

После подстановки (25) и (26) в (24) система уравнений для определения вершин примет вид:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -b & b & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{n-1,n-1} & a_{n-1,n} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{n,n-1} & a_{nn} & a_{n,n+1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -c & c \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ \dots \\ p_{n-1} \\ p_n \\ p_{n+1} \\ p_{n+2} \end{bmatrix} = \begin{bmatrix} a_1 \\ a'_1 \\ a_2 \\ a_3 \\ a_4 \\ \dots \\ a_{n-2} \\ a_{n-1} \\ a'_n \\ a_n \end{bmatrix}$$

где

$$\begin{aligned} b &= \frac{3}{x_2 - x_1}, \quad c = \frac{3}{x_n - x_{n-1}}, \\ a_{kk-1} &= \frac{x_k - x_{k-1}}{x_k - x_{k-3}} \frac{x_k - x_{k-1}}{x_k - x_{k-2}}, \\ a_{kk} &= \frac{x_{k-1} - x_{k-2}}{x_{k+1} - x_{k-3}} \frac{x_{k+1} - x_{k-1}}{x_k - x_{k-2}} + \frac{x_k - x_{k-1}}{x_k - x_{k-3}} \frac{x_{k-1} - x_{k-3}}{x_k - x_{k-2}}, \\ a_{kk+1} &= \frac{x_{k-1} - x_{k-2}}{x_{k+1} - x_{k-2}} \frac{x_{k-1} - x_{k-2}}{x_k - x_{k-2}}. \end{aligned}$$

Матрица системы уравнений является трехдиагональной. Для замкнутого кубического сплайна система уравнений (8) упрощается, так как в ней исчезнут второе и предпоследнее уравнения, определяющие производные кривой на краях. Число вершин p_i , $i = 1, 2, \dots, n$, NURBS кривой будет равно числу значимых точек a_i , $i = 1, 2, \dots, n$. Последовательность узлов будет иметь вид $t_i = x_{i-2}$, $i = 1, 2, \dots, n$.

2.4. Геометрия поверхности

Поверхностью называется геометрическое место точек, координаты которых описываются непрерывными и однозначными функциями $r_1(u, v), r_2(u, v), r_3(u, v)$ двух параметров u и v , принадлежащих непрерывной и связной двумерной области Ω . В декартовой системе координат $Oe_1e_2e_3$ можно описать радиус-вектором:

$$r(u, v) = \sum_{i=1}^3 r_i(u, v)e_i, \quad u, v \in \Omega \quad (28)$$

Аналитические поверхности

Аналитические поверхности – это такие поверхности, координаты радиус-вектора которых могут быть представлены аналитическими функциями двух параметров. Описание аналитических поверхностей можно представлять в виде:

$$r(u, v) = p + x(u, v)i_x + y(u, v)i_y + z(u, v)i_z, \quad (29)$$

где p – некоторая точка привязки характерной точки поверхности, i_x, i_y и i_z – взаимно ортогональные векторы единичной длины $x(u, v), y(u, v)$ и $z(u, v)$ – координаты функции.

Плоскость

Простейшей поверхностью является плоскость. Она может быть описана точкой p и двумя неколлинеарными векторами i_1, i_2 . Радиус-вектор плоскости описывается зависимостью:

$$r(u, v) = p + ui_1 + vi_2, \quad -\infty < u < \infty, \quad -\infty < v < \infty. \quad (30)$$

Длина i_1 и i_2 и их взаимная ориентация в общем случае может быть произвольной. Параметры u и v являются координатами местной декартовой системы координат, связанной с плоскостью. В качестве конструктивного элемента плоскость используется в совокупности с двумерными контурами, описывающими область Ω определения ее параметров u и v . Такая плоскость поверхности называется ограниченной контурами. Радиус-вектор ограниченной контурами плоскости описывается зависимостью:

$$r(u, v) = p + ui_1 + vi_2, \quad u, v \in \Omega. \quad (30.1)$$

2.4.1. Поверхности движения

Поверхность можно получить путем движения кривой по заданной траектории. Такие поверхности называются поверхностями движения. Область изменения параметров таких поверхностей представляет собой прямоугольник. Кривая,

движением которой получается поверхность, называется образующей, а такие траектории движения некоторой ее точки называется направляющей. Поверхность движения в качестве данных содержит образующую кривую и в том или ином виде направляющую линию. В общем случае направляющей может быть произвольная кривая. Среди всевозможных направляющих выделяют две наиболее простые – отрезок прямой линии и дугу окружности. Если направляющей служит отрезок прямой, то поверхность называется поверхностью выдавливания. Если направляющей служит дуга или вся окружность, то поверхность называется поверхностью вращения. Во всех остальных случаях поверхность будет называться кинематической поверхностью.

Поверхность выдавливания

Поверхность выдавливания, полученная движением кривой $c(t)$, $t_{min} \leq t \leq t_{max}$, вдоль вектора $\mathbf{d}$, описывается радиус-вектором:

$$r(u, v) = c(u) + v\mathbf{d}, \quad t_{min} \leq u \leq t_{max}, \quad 0 \leq v \leq 1. \quad (31)$$

Длина поверхности выдавливания определяется длиной вектора $\mathbf{d}$. В зависимости от замкнутости кривой $c(t)$ поверхность выдавливания может быть замкнутой или нет по параметру u .

Поверхность вращения

Поверхность, полученная вращением кривой $c(t)$, $t_{min} \leq t \leq t_{max}$, на угол α вокруг оси, заданной единичным вектором i и точкой p , описывается радиус-вектором:

$$\begin{aligned} r(u, v) = p + ((c(u) - p) \cdot i) i + (c(u) - p - ((c(u) - p) \cdot i) i) \cos(v) + i \\ \times (c(u) - p) \sin(v) = p + r_3(u) + r_1(u) \cos(v) + r_2(u) \sin(v), \\ t_{min} \leq u \leq t_{max}, \quad 0 \leq v \leq \alpha, \end{aligned} \quad (32)$$

где $r_3(u) = ((c(u) - p) \cdot i) i$ – составляющая вектора $c(u) - p$, параллельная оси вращения, $r_1(u) = c(u) - p - r_3(u)$ – составляющая вектора $c(u) - p$, перпендикулярная оси вращения, $r_2(u) = i \times (c(u) - p)$ – ортогональный первым двум векторам вектор, длина которого равна длине вектора $r_1(u)$. Замкнутость поверхности вращения по параметру u совпадает с замкнутостью образующей кривой $c(t)$. Если угол вращения $\alpha = 2\pi$, то поверхность замкнута по параметру v , а если $\alpha < 2\pi$, то поверхность вращения не замкнута по параметру v .

Кинематические поверхности

Кинематическая поверхность, является общим случаем поверхности движения. Пусть образующая описывается кривой $c(t)$, $t_{min} \leq t \leq t_{max}$, а направляющей является кривая $g(v)$, $v_{min} \leq v \leq v_{max}$. Пусть при построении кинематической

поверхности параметр u совпадает с параметром образующей кривой $c(u)$, а параметр v совпадает с параметром направляющей кривой $g(v)$. При движении образующей кривой вдоль направляющей кривой ориентация первой относительно второй может меняться или может оставаться неизменной. Если образующая выполняет плоскопараллельное движение, т.е. остается параллельной своему начальному положению, то такую поверхность будем называть поверхностью сдвига, в противном случае поверхность будем называть поверхностью заметания.

Поверхность сдвига. Поверхностью сдвига проще поверхности заметания. Радиус-вектор поверхности сдвига описывается формулой :

$$r(u, v) = g(v) + (c(u) - g(v_{min}) - h), \quad (33)$$

$$t_{min} \leq u \leq t_{max}, \quad v_{min} \leq v \leq v_{max},$$

где h - вектор привязки образующей к направляющей. Вектор h сдвигает кривую $c(u)$ на некоторую величину из ее начального положения относительно направляющей и сохраняет этот сдвиг во время движения. Радиус-вектор поверхностного сдвига строится как сумма двух векторов: вектора точки на направляющей кривой $g(v)$ и вектора положения точки образующей относительно начальной точки направления со сдвигом $c(u) - g(v_{min}) - h$.

2.4.2. Линейчатые поверхности

Линейчатая поверхность представляет собой геометрическое место отрезков прямых, соединяющих соответствующие точки заданных двух линий. Ее можно получить путем движения прямой линии по двум направляющим кривым линиям, при этом каждой точке одной кривой должна соответствовать вполне определенная точка другой кривой. Пусть заданы две направляющие кривые $a(t)$ и $d(w)$, где параметры кривых t и w изменяются в пределах $t_{min} \leq t \leq t_{max}$ и $w_{min} \leq w \leq w_{max}$. Радиус-вектор линейчатой поверхности определяется формулой

$$r(u, v) = (1 - v)a(t) + vd(w),$$

$$t = t_{min}(1 - u) + t_{max}u, \quad w = w_{min}(1 - u) + w_{max}u, \quad (34)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1.$$

Для направления кривых неявно производится репараметризация – приведение области изменения параметров к отрезку от 0 до 1. Можно было бы репараметризовать только одну кривую – привести область изменения одной из них к другой. Если параметризация кривых $a(t)$ и $d(u)$ совпадает, то радиус-вектор линейчатой поверхности может быть описан функцией:

$$r(u, v) = (1 - v)a(u) + vd(u), \quad u_{min} \leq u \leq u_{max}, \quad 0 \leq v \leq 1. \quad (35)$$

Если обе направляющие замкнуты, то линейчатая поверхность замкнута по

параметру u . По другому параметру линейчатая поверхность всегда незамкнута. Если направляющими линиями являются отрезки прямых линий, то поверхность является линейчатой по обоим параметрам и для ее построения достаточно знать радиус-векторы концевых точек отрезков. Пусть один направляющий отрезок проведен из точки p_1 в точку p_2 , а другой направляющий отрезок проведен из точки p_3 в точку p_4 . Тогда линейчатая по двум параметрам поверхность определится векторной функцией

$$\begin{aligned} r(u, v) &= (1 - v)(p_1(1 - u) + p_2u) + v(p_3(1 - u) + p_4u) = \\ &= (1 - u)(1 - v)p_1 + u(1 - v)p_2 + (1 - u)vp_3 + uvp_4, \\ &0 \leq u \leq 1, \quad 0 \leq v \leq 1. \end{aligned} \quad (36)$$

Такая поверхность называется билинейной. Она всегда незамкнута. Если все четыре точки лежат в одной плоскости, то поверхность представляет собой часть плоскости. В некоторых случаях удобно описывать плоскость по четырем точкам, лежащих в углах квадрата, в виде зависимости (36). Описанная таким образом плоскость отличается от (30) тем, что ее параметрические размеры не зависят от размеров самой плоскости. При выполнении пересечений и других операций над поверхностями желательно, чтобы у поверхностей области определения параметров не зависели от размеров поверхностей. Если точки p_1 и p_3 или p_2 и p_4 совпадают, то мы получим треугольную поверхность, которая всегда плоская.

Частным случаем линейчатой поверхности является секториальная поверхность. Она получается из линейчатой поверхности, если одну из кривых, например $d(w)$, заменить точкой p . Радиус-вектор секториальной поверхности описывается формулой

$$r(u, v) = (1 - v)a(u) + vp, \quad t_{min} \leq u \leq t_{max}, \quad 0 \leq v \leq 1. \quad (37)$$

Если направляющая $a(t)$ замкнута, то секториальная поверхность также замкнута по u . По другому параметру секториальная поверхность всегда не замкнута.

Треугольная и секториальная поверхности имеют точку, в которой одна из частных производных ее радиус-вектора равно нулю. Аналогичные точки имеют сферические и канонические поверхности. Эти точки не являются особыми, так как другая частная производная радиус-вектора поверхности не равно нулю. Если рассматривать кривые на поверхностях, то в таких точках находятся отрезки двумерных координатных линий.

С помощью линейчатой поверхности можно моделировать, детали с уклоном и фаски деталей. Направляющими кривыми поверхности фаски служат кривые на

поверхностях, отстоящие от ребра на заданном расстоянии [1].

2.4.3. Сплайновые поверхности

Корпуса судов, фюзеляжи и крылья самолетов часто проектируются с помощью поперечных или продольных сечений, проходящих в заданных местах. Эти сечения соединяются плавными кривыми, в результате чего получается единая трехмерная форма. Поверхности, построенные таким образом, называются сплайновыми поверхностями. Существуют различные виды сплайновой поверхности, такие как: поверхность Эрмита, поверхность Лагранжа, поверхность перехода и другие. Рассмотрим подробнее поверхность Эрмита.

Пусть имеется семейство кривых $r_j(t_j), j = 0, 1, 2, \dots, n$. Кривые семейства не должны пересекать друг друга. На этом семействе можно построить гладкую поверхность, имеющую непрерывные производные по всем направлениям. Приведем все кривые к одной параметрической длине, так чтобы область изменения параметра была $u_{min} \leq u \leq u_{max}$. Кривые переобозначим в соответствии с их местом на поверхности: $r(u, j) = r_j(t_j), j = 0, 1, 2, \dots, n$. Пусть параметр поверхности v между кривыми $r(u, i)$ и $r(u, k)$ изменяется от i до k .

Радиус-вектор сглаживания поверхности, построенной по семейству сечений, являющийся аналогом составного сплайна Эрмита, описывается формулой

$$\begin{aligned}
 r(u, v) &= [\alpha_0(w) \quad \alpha_1(w) \quad \beta_0(w) \quad \beta_1(w)] \cdot \begin{bmatrix} r(u, j) \\ r(u, j+1) \\ r_v(u, j) \\ r_v(u, j+1) \end{bmatrix} \\
 &= r(u, j)(1 - 3w^2 + 2w^3) + r(u, j+1)(3w^2 - 2w^3) \\
 &\quad + r_v(u, j)(w - 2w^2 + w^3) + r_v(u, j+1)(-w^2 + w^3), \\
 w &= w(v) = v - j, \quad u_{min} \leq u \leq u_{max}, \quad 0 \leq v \leq n,
 \end{aligned} \tag{38}$$

где j равно целой части параметра v и номеру кривой, $n + 1$ – число кривых. Данная поверхность является кубической по v направлению. Параметр w является местным параметром данной части поверхности в v направлении, он изменяется от 0 до 1.

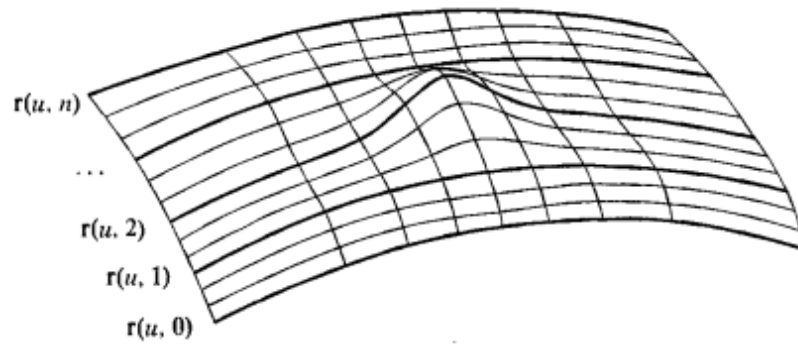


Рисунок 6 - Поверхность на семействе кривых

Производные на линиях сетки $r_v(u, j)$ могут быть вычислены по соседним линиям, аналогично тому, как вычисляются производные в характеристических точках составного сплайна Эрмита, например,

$$r_v(u, j) = \frac{r(u, j+1) - r(u, j-1)}{2} \quad (39)$$

для внутренних линий и

$$\begin{aligned} r_v(u, 0) &= 2r(u, 1) - 2r(u, 0) - r_v(u, 1), \\ r_v(u, n) &= 2r(u, n) - 2r(u, n-1) - r_v(u, n-1) \end{aligned} \quad (40)$$

для крайних линий.

Эти зависимости обеспечивают равенство нулю третьих производных на крайних кривых поверхности в трансверсальных к краю направлениях. Если все кривые семейства являются замкнутыми, то поверхность (6) получается в соответствующем направлении. Поверхность может быть замкнутой и в другом параметрическом направлении. Производные $r_v(u, j)$ могут быть вычислены и по иным формулам, приведенным для сплайна Эрмита. Поверхность Эрмита, построенная на семействе NURBS кривых, приведена на рисунке 7 [1].

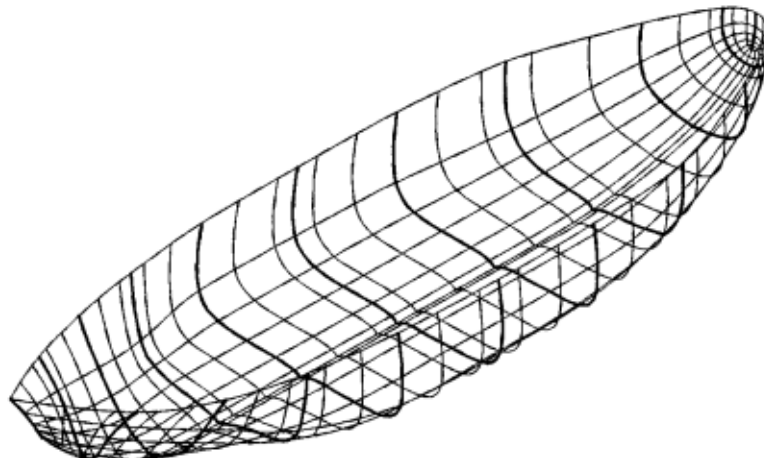


Рисунок 7 - Поверхность Эрмита на семействе кривых

2.4.4. Поверхности Безье

Рассмотренные ранее кривые Безье (5) характерны тем, что их радиус-вектор представлен в виде суммы радиус-векторов точек в пространстве, умноженных на некоторые коэффициенты, являющиеся скалярными функциями параметра кривой. Скалярными функциями являются функции Бернштейна. Аналогичным образом строятся поверхности Безье, радиус-вектор которых представляется в виде суммы радиус-векторов точек в пространстве, умноженных на коэффициенты, являющиеся функциями Бернштейна параметров поверхности.

Пусть имеется совокупность точек p_{ij} , образующих сетку, т.е. условно расположенных в виде $n+1$ рядов по $m+1$ точек в каждом. Индексы точки p_{ij} означают, что данная точка расположена j -й по счету в i -ом ряду (первый индекс равен номеру ряда, второй - номеру точки в ряду). Нумеровать ряды будем от 0 до n , а точки в них будем нумеровать от 0 до m . Поверхность Безье, построенная на совокупности точек p_{ij} , определяется радиус-вектором

$$r(u, v) = \sum_{i=0}^n \sum_{j=0}^m B^m_j(u) B^n_i(v) p_{ij} = \sum_{i=0}^n \sum_{j=0}^m B^{nm}_{ij}(v, u) p_{ij}, \quad (41)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1,$$

где $B^{nm}_{ij}(v, u) = B^n_i(v) B^m_j(u)$ функции базиса Бернштейна:

$$B^n_i(v) = \frac{n!}{(n-i)! i!} v^i (1-v)^{n-i} = C_n^i v^i (1-v)^{n-i},$$

$$B^m_j(u) = \frac{m!}{(m-j)! j!} u^j (1-u)^{m-j} = C_m^j u^j (1-u)^{m-j}. \quad (42)$$

Коэффициент Бернштейна удовлетворяет равенству:

$$\sum_{i=0}^n B^n_i(t) = (t + (1-t))^n = 1^n = 1, \quad (43)$$

в соответствии, с которым функция $B^{nm}_{ij}(v, u)$ удовлетворяет равенству:

$$\sum_{i=0}^n \sum_{j=0}^m B^{nm}_{ij}(v, u) = 1. \quad (44)$$

Если представить что на каждых четырех соседних точках p_{ij} , $p_{i,j+1}$, $p_{i+1,j}$, $p_{i+1,j+1}$ построена билинейная поверхность, то мы получим многогранник, который будем называть характеристическим. Точки p_{ij} будем называть вершинами характеристического многогранника. Поверхность Безье вместе со своим характеристическим многогранником на рисунке ниже.

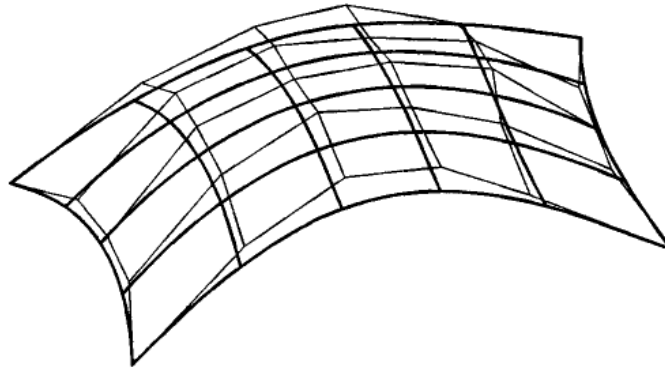


Рисунок 8 - Поверхность Безье и ее характеристический многогранник

Так как из коэффициента Бернштейна только нулевой и последний принимают максимально возможные значения, то поверхность Безье проходит только через угловые вершины сетки: $p_{00}, p_{0m}, p_{n0}, p_{nm}$. Сечение по линиям $u = \text{const}$ или $v = \text{const}$ представляет собой кривые Безье. Характеристический многогранник дает общее представление о поверхности Безье, а сама поверхность как бы сглаживает углы этого многогранника. Чем выше порядок поверхности, тем она более гладкая. Порядок поверхности в u -направлении жестко связан с количеством рядов вершин. Область изменения параметров поверхности представляет собой квадрат со стороной равной 1. Степень поверхности Безье, а соответственно и ее гладкость, жестко связаны с количеством вершин, по которым она построена. Это может создать некоторые неудобства при использовании поверхности Безье. Преодолеть эти неудобства могут помочь составные поверхности. Составная поверхность Безье получается в результате стыковки отдельных поверхностей Безье, имеющих вдоль стыкуемых краев одинаковые степени. В общем случае для того чтобы части поверхности Безье гладко стыковались друг с другом, нужно чтобы ребра, прилегающие к углам стыкуемых частей поверхности, лежали в одной плоскости.

2.4.5. Рациональные поверхности

В предыдущем разделе было описано, что кривые Безье можно обобщить так, чтобы она могла описывать заданное коническое сечение. Аналогичные рассуждения приведут нас к тому, что поверхность Безье можно обобщить так, чтобы она могла описать требуемую часть поверхности второго порядка — эллипсоид, гиперболоид, параболоид или их частные случаи.

Наделим каждую вершину кривой ... p_{ij} весом w_{ij} и будем описывать поверхность в однородных координатах. Поверхности, характеристические точки

которых обладают весами, называются рациональными. Сам радиус-вектор такой поверхности тоже приобретает вес. Вес радиус-вектора вычисляется по точно таким же формулам, что и его координаты, поэтому все принято считать дополнительной координатой точки. Рациональные поверхности Безье можно получить подобно тому, как строились рациональные кривые Безье. Расширенный радиус-вектор рациональной поверхности Безье описывается формулой, по виду совпадающей с ... , в которой обычные векторы $r(u, v)$ и p_{ij} заменены соответствующими расширенными векторами $R(u, v)$ и P_{ij} .

Обычный трехмерный радиус-вектор рациональной поверхности Безье определяется формулой

$$r(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m B^n_i(v) B^m_j(u) w_{ij} p_{ij}}{\sum_{i=0}^n \sum_{j=0}^m B^n_i(v) B^m_j(u) w_{ij}}, \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1. \quad (45)$$

С помощью поверхности (45) может быть построена часть поверхности второго порядка или часть поверхности тора. Для этого проще всего использовать рациональную поверхность Безье второй степени, построенную по девяти вершинам:

$$r(u, v) = \frac{[(1-v)^2 \quad 2v(1-v) \quad v^2] \cdot \begin{bmatrix} P_{00} & w_u P_{01} & P_{02} \\ w_v P_{01} & w_v w_v P_{11} & w_v P_{12} \\ P_{20} & w_u P_{21} & P_{22} \end{bmatrix} \cdot \begin{bmatrix} (1-u)^2 \\ 2u(1-u) \\ u^2 \end{bmatrix}}{((1-u)^2 + 2u(1-u)w_u + u^2)((1-u)^2 2u(1-u)w_v + v^2)}, \quad (46)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1.$$

В (46) веса угловых вершин положены равными единице, а для остальных вершин введены веса w_u и w_v , имеющие индексы параметрических направлений поверхностей. Тип поверхности, получаемой по формуле (46) зависит от положения вершин и от весов w_u и w_v . С помощью составной рациональной поверхности Безье может быть построена полностью любая поверхность второго порядка. Для этого нужно использовать несколько поверхностей (46), гладко стыкующихся между собой.

Поверхность скругления. Рациональной поверхностью можно описать поверхность скругления ребра детали. Сечение поверхности скругления в одном из параметрических направлений должно представлять собой дугу окружности. Это дугу можно описать с помощью рациональной кривой Безье, в которой вершины p_0, p_1, p_2 и угол дуги α будем считать функциями одного из параметров поверхности скругления. Радиус-вектор поверхности скругления выражается формулой

$$r(u, t) = \frac{(1-t)^2 p_0(u) + 2t(1-t)w(u)p_1(u) + t^2 p_2(u)}{(1-t)^2 + 2t(1-t)w(u) + t^2}, \quad (47)$$

$$u_{min} \leq u \leq u_{max}, \quad 0 \leq t \leq 1,$$

где $w(u) = \cos\left(\frac{\alpha(u)}{2}\right)$ вес линии $p_1(u)$, вычисленный по углу дуги в данном месте.

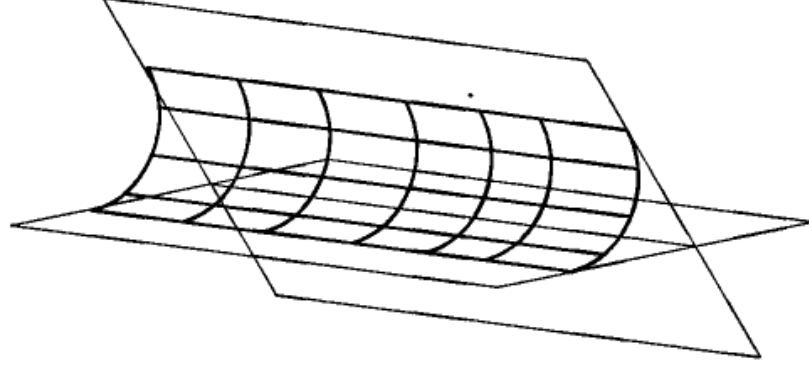


Рисунок 9 - Поверхность скругления.

Простейший пример построения поверхности скругления между двумя плоскостями представлен на рис. 9.

Радиус-вектор рациональной параметрической поверхности, построенной на mn вершинах, определяется формулой

$$r(u, v) = \frac{\sum_{i=1}^n \sum_{j=1}^m F_{ij}(v, u) w_{ij} p_{ij}}{\sum_{i=1}^n \sum_{j=1}^m F_{ij}(v, u) w_{ij}}, \quad (48)$$

$$u_{min} \leq u \leq u_{max},$$

$$v_{min} \leq v \leq v_{max},$$

где p_{ij} – радиус-векторы j -й вершины, расположенной в i -м ряду, характеристического многогранника поверхности, w_{ij} – вес соответствующей вершины, $F_{ij}(v, u)$ – функция j -й вершины в i -м ряду, n – число рядов вершин, m – число вершин в каждом ряду.

В терминах однородных координат расширенный радиус-вектор рациональной поверхности определяется формулой

$$R(u, v) = \sum_{i=1}^n \sum_{j=1}^m F_{ij}(v, u) p_{ij}, \quad u_{min} \leq u \leq u_{max}, \quad v_{min} \leq v \leq v_{max}, \quad (49)$$

где $p_{ij} = [w_{ij} \quad p_{ij} \quad w_{ij}]^T$ – расширенные векторы поверхности.

Радиус-вектор рациональной поверхности вычисляется как частное от деления двух функций параметров u и v , при вычислении произвольных рациональной поверхности правая часть (49) следует рассмотреть как сложную функцию. Если

условно обозначим радиус-вектор рациональной поверхности как $r = \frac{wr}{w}$, то произвольный радиус-вектор рациональной поверхности находится по формулам

$$\frac{\partial r}{\partial u} = \frac{1}{w} \frac{\partial(wr)}{\partial u} - \frac{wr}{w^2} \frac{\partial w}{\partial u}, \quad (50)$$

$$\frac{\partial r}{\partial v} = \frac{1}{w} \frac{\partial(wr)}{\partial v} - \frac{wr}{w^2} \frac{\partial w}{\partial v}, \quad (51)$$

$$\frac{\partial^2 r}{\partial u^2} = \frac{1}{w} \frac{\partial^2(wr)}{\partial u^2} - \frac{2}{w^2} \frac{\partial(wr)}{\partial u} \frac{\partial w}{\partial u} - \frac{wr}{w^2} \frac{\partial^2 w}{\partial u^2} + \frac{2wr}{w^3} \left(\frac{\partial w}{\partial u} \right)^2, \quad (52)$$

$$\frac{\partial^2 r}{\partial v^2} = \frac{1}{w} \frac{\partial^2(wr)}{\partial v^2} - \frac{2}{w^2} \frac{\partial(wr)}{\partial v} \frac{\partial w}{\partial v} - \frac{wr}{w^2} \frac{\partial^2 w}{\partial v^2} + \frac{2wr}{w^3} \left(\frac{\partial w}{\partial v} \right)^2, \quad (53)$$

$$\frac{\partial^2 r}{\partial u \partial v} = \frac{1}{w} \frac{\partial^2(wr)}{\partial u \partial v} - \frac{1}{w^2} \frac{\partial(wr)}{\partial u} \frac{\partial w}{\partial v} - \frac{1}{w^2} \frac{\partial(wr)}{\partial v} \frac{\partial w}{\partial u} - \frac{wr}{w^2} \frac{\partial^2 w}{\partial u \partial v} + \frac{2wr}{w^3} \frac{\partial w}{\partial u} \frac{\partial w}{\partial v}, \quad (54)$$

Для производных более высокого порядка расчеты аналогичные [19].

2.4.6. NURBS – поверхности

В качестве функции $F_{ij}(v, u)$ в формуле (48) будем использовать произведение В-сплайнов, один из которых зависит от параметра u , а другой зависит от параметра v . В результате получим неоднородную рациональную поверхность, определяемую В-сплайнами, которая приобрела название NURBS поверхности (*Non-Uniform Rational B-Spline surface*). NURBS поверхность, как и поверхность Безье, строится на сетке вершин – совокупности точек p_{ij} , условно расположенных в виде n рядов по m точек в каждом ряду. Индексы вершины p_{ij} обозначают то, что данная точка расположена j -ой по счету в i -м ряду. Пусть вес вершины p_{ij} равен w_{ij} . Пусть В-сплайны $N_{jk}(u)$ и $N_{iq}(v)$ – направления имеют порядок k , а В-сплайны $N_{iq}(v)$ – направления имеют порядок q . Радиус-вектор NURBS поверхности определяется равенством

$$r(u, v) = \frac{\sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) w_{ij} p_{ij}}{\sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) w_{ij}} = \frac{\sum_{i=1}^n \sum_{j=1}^m N_{ij,qk}(v, u) w_{ij} p_{ij}}{\sum_{i=1}^n \sum_{j=1}^m N_{ij,qk}(v, u) w_{ij}}, \quad (55)$$

$$u_{min} \leq u \leq u_{max}, \quad v_{min} \leq v \leq v_{max},$$

где $N_{ij,qk}(v, u) = N_{iq}(v)N_{jk}(u)$ – базисные функции, n – число рядов вершин, m – число вершин в каждом ряду. Производные радиус-вектора NURBS поверхности определяется равенствами (50)-(54).

Каждый из В-сплайнов $-го$ порядка $N_{jk}(u)$ построен на последовательности из $k + 1$ узлов $u_j, u_{j+1}, u_{j+2}, \dots, u_{j+k}$; всего для построения совокупности m В-сплайнов k -го порядка требуется $m + k$ узлов u_j в случае незамкнутой кривой и $m + 2k$ узлов в случае замкнутой кривой. Каждый из В-сплайнов $-го$ порядка $N_{iq}(v)$ построен на последовательности $q + 1$ узлов $v_i, v_{i+1}, v_{i+2}, \dots, v_{i+q}$; всего для построения совокупности n В-сплайнов q -го порядка требуется $n + q$ узлов v_i в случае незамкнутой кривой и $n + 2q$ узлов в случае замкнутой кривой. Как следует из (55), NURBS поверхность имеет две последовательности узлов, одну в u – направлении, а другую в v – направлении. Значение узлов должны образовать неубывающие последовательности. Будем называть их u – последовательности узлов и v – последовательности узлов. Пронумеруем узлы каждой последовательности, начав с 1 [18].

Для незамкнутой NURBS поверхности используются следующие последовательности узлов. Первые k узлов u – последовательности имеют значения, равные нулю: $u_1 = u_2 = \dots = u_k = 0$; следующие $m - k$ узлов принимают целочисленные значения от 1 до $m - k$: $t_{k+j} = j$ ($j = 1, 2, \dots, m - k$); оставшиеся k узлов принимают значение $m - k + 1$: $t_{m+1} = t_{m+2} = \dots = t_{m+k} = m - k + 1$. Первые q узлов v – последовательности имеют значения, равные нулю: $v_1 = v_2 = \dots = v_q = 0$; следующие $n - q$ узлов принимают целочисленные значения от 1 до $n - q$: $t_{q+i} = i$ ($i = 1, 2, \dots, n - q$); оставшиеся q узлов принимают значение $n - q + 1$: $t_{n+1} = t_{n+2} = \dots = t_{n+q} = n - q + 1$.

Для замкнутой NURBS поверхности используются равномерные последовательности узлов с единичным шагом $u_j = j - m, j = 1, 2, \dots, m + 2k, v_i = i - n, i = 1, 2, \dots, n + 2q$.

Приведенные примеры узловых последовательностей не являются единственными возможными. Как и для любой поверхности, внутренняя параметризация NURBS поверхности может быть произвольной.

Области изменения параметров NURBS поверхности представляет собой прямоугольник: $u_k \leq u \leq u_{m+k}, v_q \leq v \leq v_{n+q}$. Для указанных последовательностей узлов параметры NURBS поверхности используем формулы (кривая Безье 2-4). По значению параметра u из условия $u_j \leq u \leq u_{j+1}$ определим номер j единственного отличного от нуля В-сплайна первого порядка:

$$M_{j,1}(u) = \frac{1}{u_{j+1} - u_j}. \quad (56)$$

Далее, используя рекуррентное соотношение:

$$M_{r,k}(u) = \frac{(u_{r+k} - u) M_{r+1,k-1}(u) + (u - u_r) M_{r,k-1}(u)}{(u_{r+k} - u_r)} \quad (57)$$

$r = j - k + 1, j - k + 2, \dots, j$, последовательно вычислим все отличные от нуля при данном u В-сплайны до k -го порядка, нормируем в соответствии с (кБ 4)

$$N_{r,k}(u) = (u_{r+k} - u_r) M_{r,k}(u), \quad r = j - k + 1, j - k + 2, \dots, j.$$

Аналогично по параметру v из условия $v_i \leq v \leq v_{i+1}$ можно определить номер i отличного от нуля В-сплайна первого порядка.

Используя рекуррентные соотношения, описанные выше, получим:

$$N_{r,q}(v) = (v_{r+q} - v_r) M_{r,q}(v), \quad r = i - q + 1, i - q + 2, \dots, i. \quad (58)$$

Порядок NURBS поверхности в том или ином параметрическом направлении равен порядку соответствующих В-сплайнов, а порядок В-сплайна на единицу больше степени полиномов, из которых он построен.

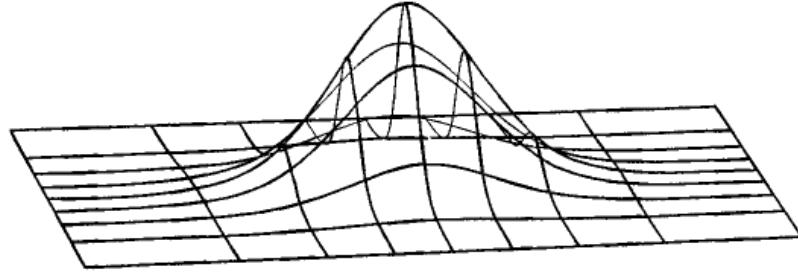


Рисунок 10 - NURBS поверхность

На рисунке 10 показана NURBS поверхность 4-го порядка по обоим параметрическим направлениям, построенная на сетке, состоящей из 7×7 вершин. Все вершины, кроме центральной, лежат в одной плоскости. Все вершины имеют вес равный 1. Если на каждой из четырех характеристических точек построить билинейные поверхности, то мы получим некоторый многогранник поверхности, приведенной на рис. 10, показан на рисунке 11.

В терминах однородных координат расширенный радиус-вектор NURBS поверхности определяется формулой:

$$R(u, v) = \sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) p_{ij}, \quad (59)$$

$$u_{min} \leq u \leq u_{max}, \quad v_{min} \leq v \leq v_{max},$$

где $p_{ij} = [w_{ij} \ p_{ij} \ w_{ij}]^T$ - расширенные векторы вершин поверхности.

Когда все вершины NURBS поверхности имеют равные веса, то формула (55) в силу свойства В-сплайнов

$$\sum_i N_{i,m}(t) = \sum_i N_{m,i}(t) = 1 \quad (60)$$

примет вид:

$$r(u, v) = \sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) p_{ij}, \quad (61)$$

$$u_{min} \leq u \leq u_{max}, \quad v_{min} \leq v \leq v_{max},$$

Эта поверхность называется В-сплайн поверхностью. Она также может быть использована в проектировании, так как обладает определенной гладкостью, и для вычисления ее производных не требуется пользоваться формулой (50-54).

Если построить NURBS поверхность (61) на базе В-сплайнов второго порядка, то она совпадает со своим характеристическим многогранником и представляет собой совокупность билинейных поверхностей.

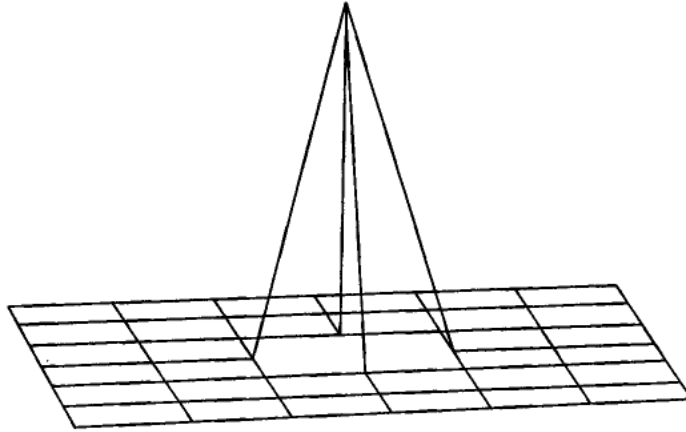


Рисунок 11 - Характеристический многогранник NURBS поверхности

Если $n = q$ и $m = k$, т.е. если число рядов вершин n равно порядку q В-сплайнов v -направления, число вершин в ряду m равно порядку k В-сплайнов u -направления, узлы u -последовательности имеют значение $u_1 = u_2 = \dots = u_m = 0$, $u_{m+1} = u_{m+2} = \dots = u_{2m} = 1$, узлы v - последовательности имеют значение $v_1 = v_2 = \dots = v_n = 0$, $v_{n+1} = v_{n+2} = \dots = v_{2n} = 1$, то NURBS поверхность 10 совпадает с рациональной поверхностью Безье 11. Это следует из того что, функции Бернштейна являются частными случаями В-сплайнов. Область изменения

параметров NURBS поверхности в этом случае представляет собой квадрат со стороной, равной 1. При помощи поверхности (10) может быть построена любая поверхность второго порядка.

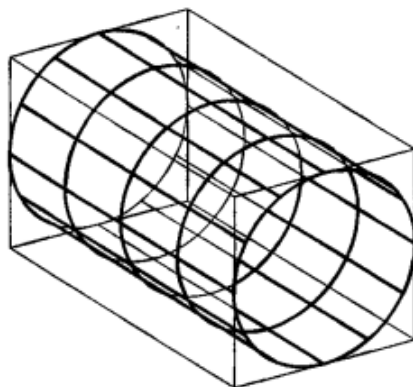


Рисунок 12 - Цилиндрическая NURBS поверхность

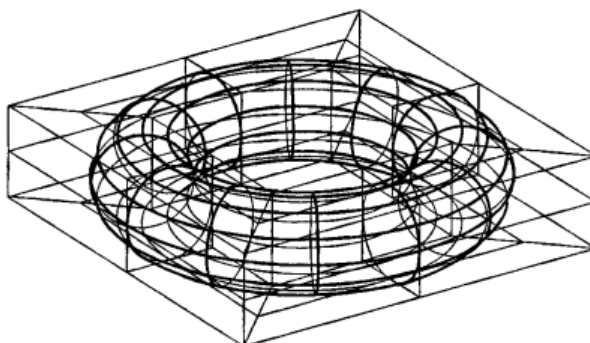


Рисунок 13 - Тороидальная NURBS поверхность

На рисунке 12 приведена цилиндрическая NURBS поверхность и ее характеристический многогранник. На рис 13 приведена тороидальная NURBS поверхность и ее характеристический многогранник. Обе поверхности имеют кратные узлы. По своему внешнему виду NURBS поверхности и поверхности Безье похожи. Но в отличие от последних порядок NURBS поверхности не связан жестко с количеством вершин и предоставляет возможность строить поверхности невысокого порядка (3-го или 4-ого) на большом числе вершин. Эта возможность придает поверхности большую гибкость, так как поверхности Безье построенные на большом числе вершин, имеют высокий порядок и являются слишком гладкими.

NURBS поверхности являются обобщением большинства поверхностей: многогранной билинейной поверхности, поверхности Безье, поверхности второго порядка, рациональных поверхностей. Каждая линия $u = \text{const}$ или $v = \text{const}$ на поверхности является NURBS кривой. Но можно построить рациональные неоднородные поверхности на основе В-сплайнов, которые не обязательно давали бы NURBS кривые в качестве всех своих параметрических u – линий или v – линий.

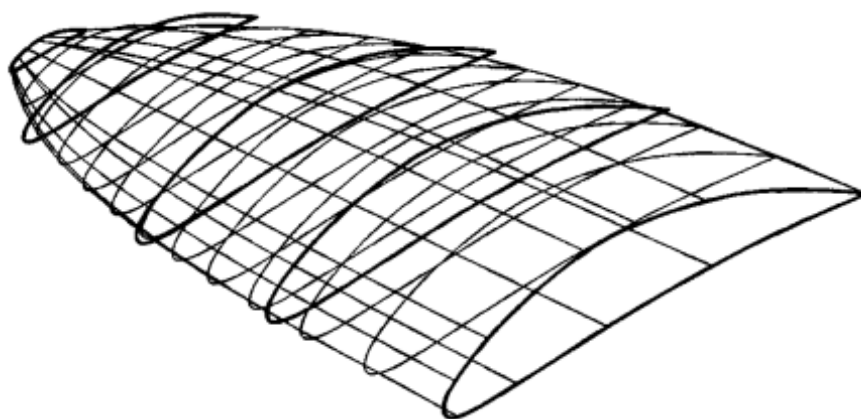


Рисунок 14 - NURBS поверхности на семействе кривых

Можно получить поверхность, если в формуле (14) вместо вершин характеристической ломаной подставить кривые $p_i = p_i(u), i = 1, 2, \dots, n$, обладающие весами $w_i = w_i(u)$, и имеющие одинаковые параметрические длины $u_{min} \leq u \leq u_{max}$. В результате, получим поверхность, описываемую формулой

$$r(u, v) = \frac{\sum_{i=1}^n N_{i,m}(v) w_i(u) p_i(u)}{N_{i,m}(v) w_i(u)}, \quad (61)$$

$$u_{min} \leq u \leq u_{max}, \quad v_{min} \leq v \leq v_{max}.$$

Только параметрические линий $u = const$ на поверхности (14) являются NURBS кривыми. На рисунке 14 показана NURBS поверхность и пять кривых, по которым она построена. Частным случаем этой поверхности является поверхность скругления [18].

2.5. Топология

Оболочки. Поверхности могут быть замкнутыми по одному или двум параметрическим направлениям или незамкнутыми. Незамкнутые поверхности имеют границу. Границей будем называть линию на поверхности, соответствующую движению ее параметров по границе их области определения. Линия на замкнутой поверхности стыкуется сама с собой. Совокупность стыкующихся по границам поверхностей будем называть оболочкой. Оболочка может состоять из одной поверхности или нескольких поверхностей. Также как и отдельная поверхность, оболочка может быть замкнутой и незамкнутой. Замкнутая оболочка не имеет границ. Незамкнутая оболочка имеет одну или несколько границ.

Топология изучает общий случай оболочек, которые могут самопересекаться, иметь или не иметь границы, уходить в бесконечность. Топология оперирует своими объектами, которые несут информацию о их взаимной связи друг с другом, и устанавливает между ними соотношения. При моделировании окружающих нас

объектов мы будем строить оболочки топологических объектов. Они будут нести и количественную геометрическую информацию, и топологическую информацию. Количественная геометрическая информация топологического объекта содержится в его геометрическом носителе, которым может являться точка, кривая или поверхность.

Вершины, ребра, циклы, грани. Топологические объекты несут в себе одновременно и метрическую и топологическую информацию. Одним из топологических объектов является оболочка. При построении оболочки используются такие топологические объекты, как грани, ребра, вершины и циклы. Все топологические объекты имеют общие принципы построения.

Гранью будем называть топологический объект, построенный на основе поверхности. Фактически грань представляет собой поверхность плюс информация о том, какая сторона поверхности является наружной стороной грани, и информация об ее положении в оболочке, т.е. информация об ее соседях.

Цикл – это топологический объект, который описывает одну из границ грани, и содержит информацию о том, где и как к данной грани примыкают соседние грани. Так как вдоль одного цикла к данной грани могут примыкать несколько соседних граней, то цикл состоит из нескольких участков. Каждый участок цикла опирается на некоторое ребро.

Ребром будем называть топологический объект, построенный на основе линии стыковки соседних граней или на основе граничной линии оболочки. Грани стыкуются только по ребрам. Таким образом, каждая грань со всех сторон окружена ребрами. Вершиной будем называть топологический объект, построенный на основе точки, в которой стыкуются ребра. Вершины могут лежать только на краях ребер. Каждое ребро начинается и оканчивается в вершине. Если ребро замкнуто, то оно начинается и заканчивается в одной и той же вершине. Цикл состоит из ребер, образующих замкнутую линию вдоль одной из границ грани. Цикл всегда замкнут и ему приписывается определенное направление. Грань может содержать несколько циклов, причем один из них является внешним, а остальные – внутренними и целиком лежащими внутри внешнего цикла.

Грани, ребра и вершины строятся на базе известных геометрических объектов (точек, кривых и поверхностей) добавление к ним информации о своих соседях и взаимной ориентации. В результате геометрические объекты приобретают новое качество, чем и обусловлено введение топологических объектов [20].

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1. Описание реализованных алгоритмов

В математическом отделении РФЯЦ-ВНИИЭФ при создании программного обеспечения используется принцип коллективной разработки. Под коллективной разработкой подразумевается сложная инфраструктура с правилами работы исполнителей т.д.

Для снижения трудозатрат разработки сложных систем дополнительно используются кроссплатформенные средства, такие как: SDK, VTK, Qt, Cmake и т.д.

В ходе работы были изучены следующие компоненты:

- SDK (Software development kit) – это пакет библиотек, собирающий в себе данные о разработке определенного программного обеспечения;
- VTK (Visualisation ToolKit) – библиотека, предназначенная для создания компьютерной 3D графики, обработки изображений и визуализации;
- Qt – кроссплатформенная библиотека инструментария разработки программного обеспечения;
- Cmake – кроссплатформенная система автоматизации сборки программного обеспечения из исходного кода.

Были разработаны алгоритмы «Построение поверхности протягиванием заданного ребра» и «Построение поверхности вращения кривой вокруг заданной оси» и реализованы в ПК ЛОГОС на языке программирования C++ в среде MS Visual Studio с использованием вышеописанных компонент.

Создание поверхности протягиванием

Для реализации операции создания поверхности протягиванием заданного ребра был разработан класс `urpCreateSurfaceExtrusionDialog`. Входными данными для выполнения операции являются:

- Ребра;
- Вектор направления;
- Расстояние;
- Масштаб.

Ребро - это кривая на поверхности или в пространстве ограниченная двумя вершинами. Для того чтобы создать ребро необходимо задать вершины. Класс `urpCreateVertexDialog` позволяет создавать вершины по заданным координатам. Каждой созданной вершине присваивается уникальный идентификационный номер (id), который записывается в структуру 3DTV и несет в себе данные об объекте. На

основании полученных вершин построим ребра. Ребра могут быть представлены различными типами: ломаная линия, сплайн, дуга и т.д. Для создания кривых такого вида используются классы: `uppCreatePolylineDialog` – класс, позволяющий создать ломаную линию по заданным вершинам и `uppCreateSplineDialog` – класс для создания сплайна по заданным вершинам. Созданные ребра, также как и вершины, получают уникальный `id`, который несет в себе информацию о данном объекте в структуре 3DTV.

Геометрический объект «ребро» подается на вход при помощи класса `Qwidget`, который выделяет выбранный объект, считывает его данные из структуры 3DTV и записывает в специально отведенной памяти для работы в рамках задачи. Направление задается аналогичным способом. Данные об исходном ребре и векторе направления позволяют создать новое ребро при помощи матрицы трансформации и коэффициента масштабирования. Для получения грани необходимым условием является наличие замкнутого контура. Для создания замкнутого контура выделим начальные и конечные вершины исходного и трансформируемого ребра и соединим их между собой. Производим проверку на замкнутость полученного контура, если она выполняется, то строится грань, в ином случае выводим оповещение об ошибке.

Создание поверхности вращения

Для реализации операции создания поверхности вращением кривой вокруг заданной оси был разработан класс `uppCreateSolidRevolutionByCurveDialog`. Входными данными для выполнения операции являются:

- Ребро;
- Ось.

Геометрический объект «ребро» и направление, в рамках данной задачи «ось вращения», задаются идентично способу, указанному в алгоритме решения задачи «построение поверхности протягиванием заданного ребра». Информация о выбранных ребрах и направления подается в программу для обработки. Ось вращения, заданная вектором направления, имеет начальную точку в пространстве. Производим вращение объекта вокруг заданной оси.

3.2. Пример использования разработанного инструмента «Создание поверхности протягиванием»

Разработанный инструментарий внедрен в ПК ЛОГОС и доступен конечному пользователю с использованием графического пользовательского интерфейса. Интерфейс ПК ЛОГОС представлен на рисунке 15.

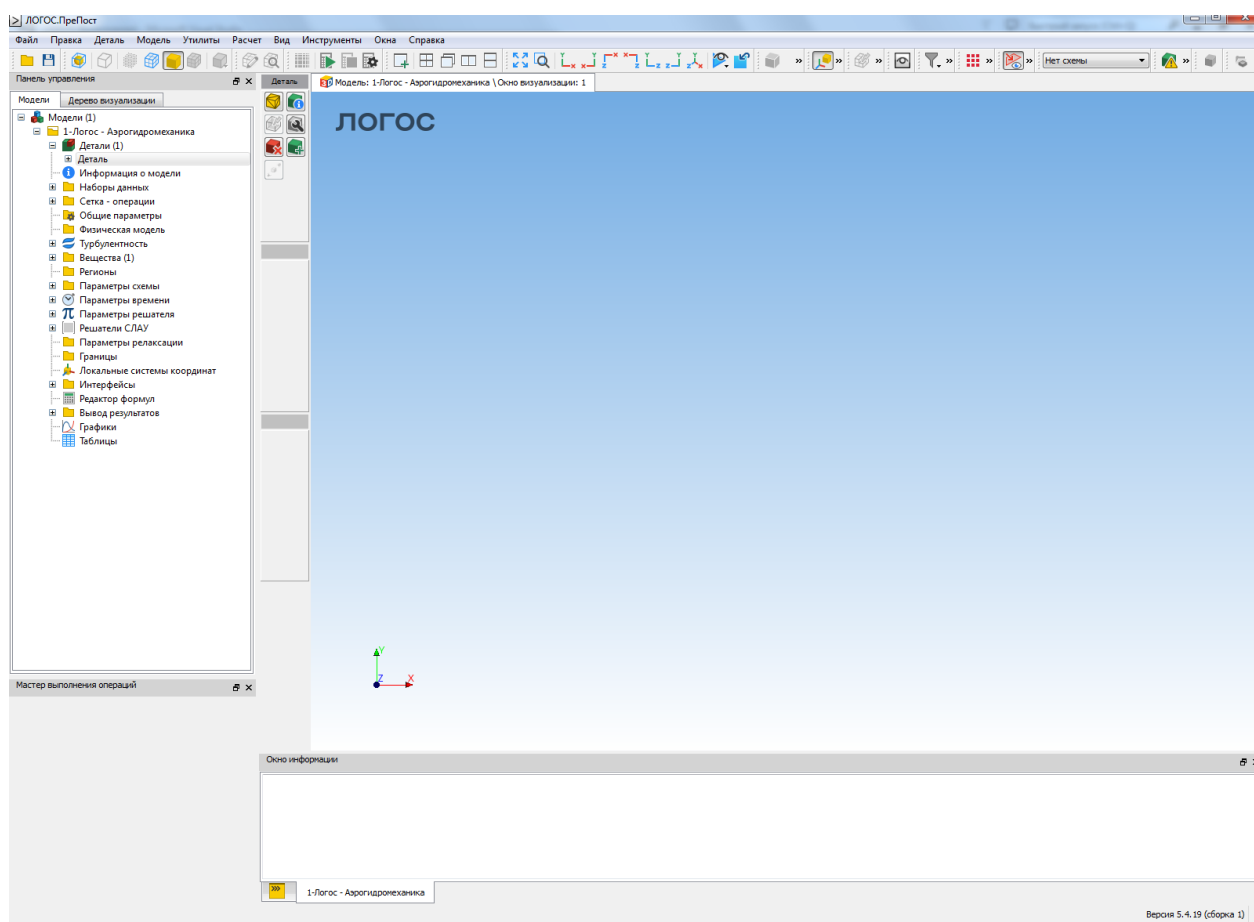


Рисунок 15 - Интерфейс ПК ЛОГОС

Для демонстрации работоспособности алгоритма построения поверхности протягиванием использованы тестовые примеры. Их параметры указаны в таблице 3.

Таблица 3 - Входные данные

Объект	Параметры объекта
Вершина	Были построены объекты, имеющие следующие координаты $r1(0,0,0)$ и $r2(1,0,0)$. Вершины представлены на рисунке 16.
Ребро	На основании уже построенных вершин $r1, r2$, мы можем создать ребро, соединив их между собой. Ребро представлено на рисунке 17.
Ломаная линия	Ломаная линия создается на основании нескольких вершин ($r1(0,1,0), r2(2,0,0), r3(4,2,0), r4(4, -1,0), r5(6,1,0)$), Ломаная линия представлена на рисунке 18.
Сплайн	Сплайн создается на основании заданных вершин $r1(0,0,0), r2(2,2,0), r3(4, -1,0), r4(6,1,0), r5(7,0,0)$, Сплайн представлен на рисунке 19.

Для вызова процедуры создания вершины необходимо выделить элемент дерева обрабатываемой детали. Активируется модуль «Деталь» и на вспомогательной панели инструментов появятся функциональные кнопки по работе с геометрией детали. Далее необходимо на вспомогательной панели инструментов перейти в режимы: «Режим создания геометрических сущностей» затем «Режим создания геометрических вершин» и выбрать процедуру «Создание вершины».

После запуска процедуры создания вершины на панели мастера выполнения операций появится диалог создания вершины. По умолчанию координаты создаваемой вершины (0,0,0). Вершина отображается в окне визуализации в виде белой точки с фиолетовым контуром. При изменении координат отображаемая вершина автоматически меняет положение. После того как были заданы координаты необходимо нажать на кнопку «Выполнить операцию» или на кнопку «Заккрыть», если операция нуждается в отмене. Вершины, созданные по координатам указанным в таблице 3, представлены на рисунке 16.

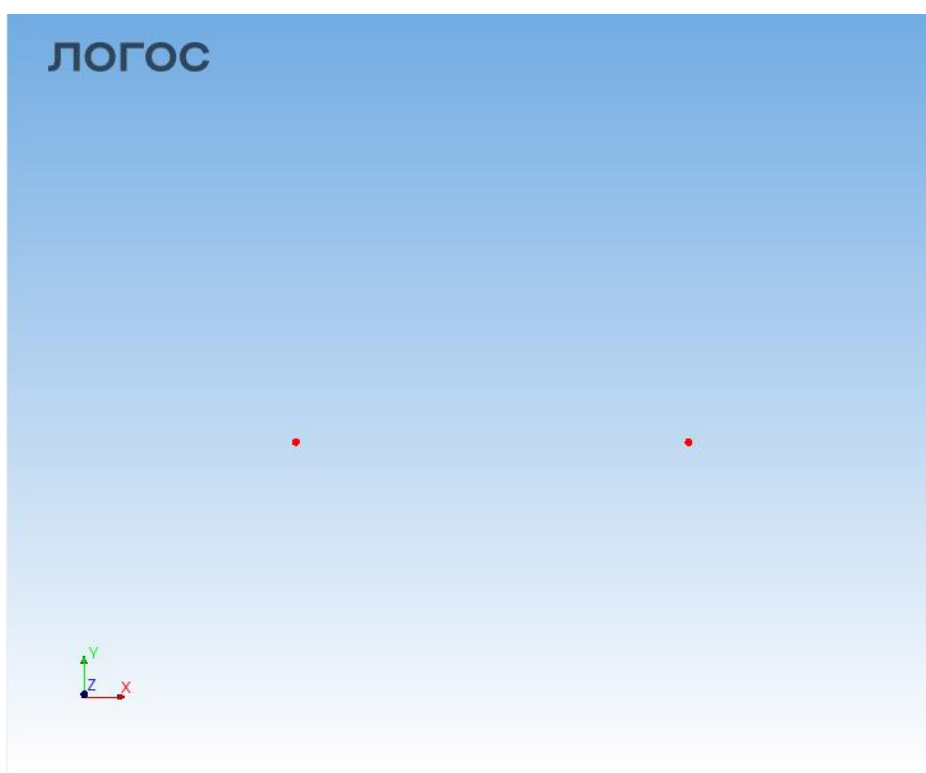


Рисунок 16 - Созданные вершины

Для вызова процедуры создания ломаной линии необходимо выделить элемент дерева обрабатываемой детали. Активируется модуль «Деталь» и на вспомогательной панели инструментов появятся функциональные кнопки по работе с геометрией детали. В дереве моделей необходимо перейти в пункт «Фрагмент» для

соответствующей детали и перейти в режимы: «Режим создания геометрических сущностей», а затем в «Режим создания геометрических ребер» и выбрать процедуру «Добавить ломаную линию».

После запуска процедуры создания ломаной линии на панели мастера выполнения операций появится диалог добавления ломаной линии. Для создания ломаной линии необходимо с помощью левой кнопки мыши в окне визуализации выбрать ее контрольные точки – вершины геометрии. В процессе выбора контрольных точек в окне визуализации отобразится предварительная визуализация ломаной линии. Для добавления ломаной линии необходимо нажать кнопку «Выполнить операцию». Результат выполнения процедуры представлен на следующем рисунке.

В окно информации будет выведен результат выполнения процедуры.

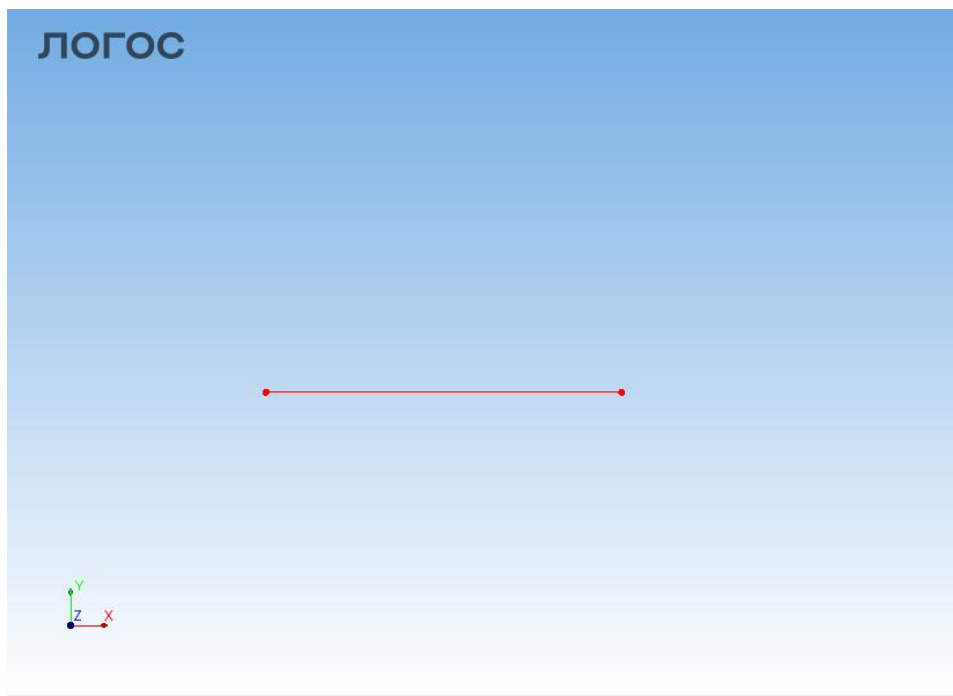


Рисунок 17 – Ребро, построенное по заданным вершинам

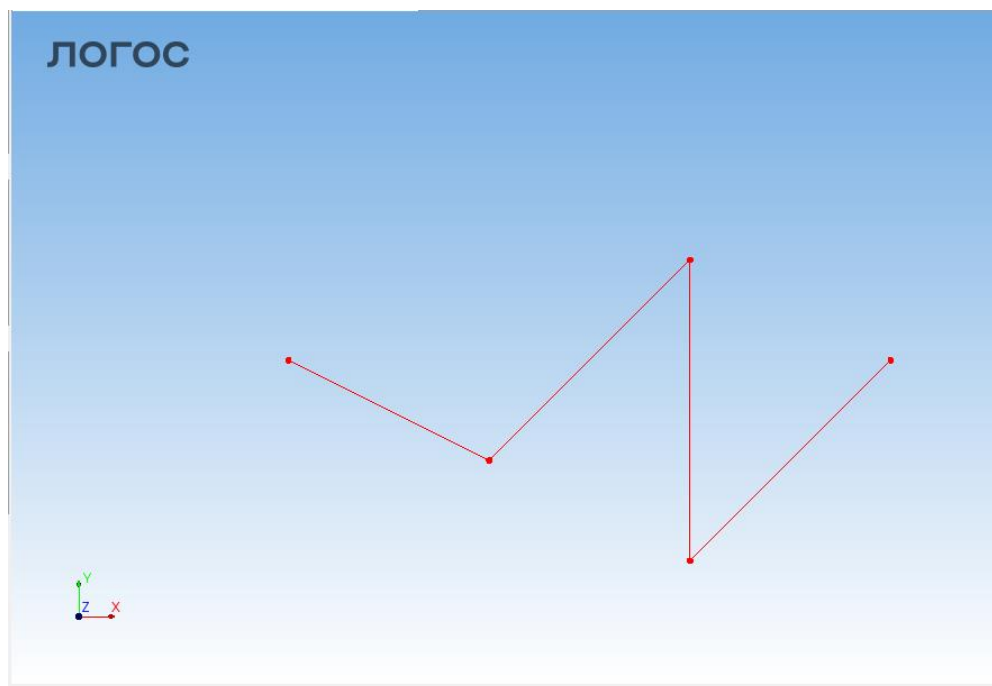


Рисунок 18 - Ломаная линия, построенная по заданным вершинам

Для вызова процедуры создания сплайна необходимо выделить элемент дерева обрабатываемой детали. Активируется модуль «Деталь» и на вспомогательной панели инструментов появятся функциональные кнопки по работе с геометрией детали. В дереве моделей необходимо перейти в пункт «Фрагмент» для соответствующей детали и перейти в режимы «Режим создания геометрических сущностей» - «Режим создания геометрических ребер» и выбрать процедуру «Создание сплайна» на вспомогательной панели инструментов.

После запуска процедуры создания сплайна на панели мастера выполнения операций появится диалог добавления сплайна. Для создания сплайна необходимо с помощью левой кнопки мыши в окне визуализации выбрать его контрольные точки – вершины геометрии. В процессе выбора контрольных точек в окне визуализации отобразится предварительная визуализация сплайна. Для добавления сплайна необходимо нажать кнопку «Выполнить операцию». Результат выполнения процедуры представлен на рисунке 19.

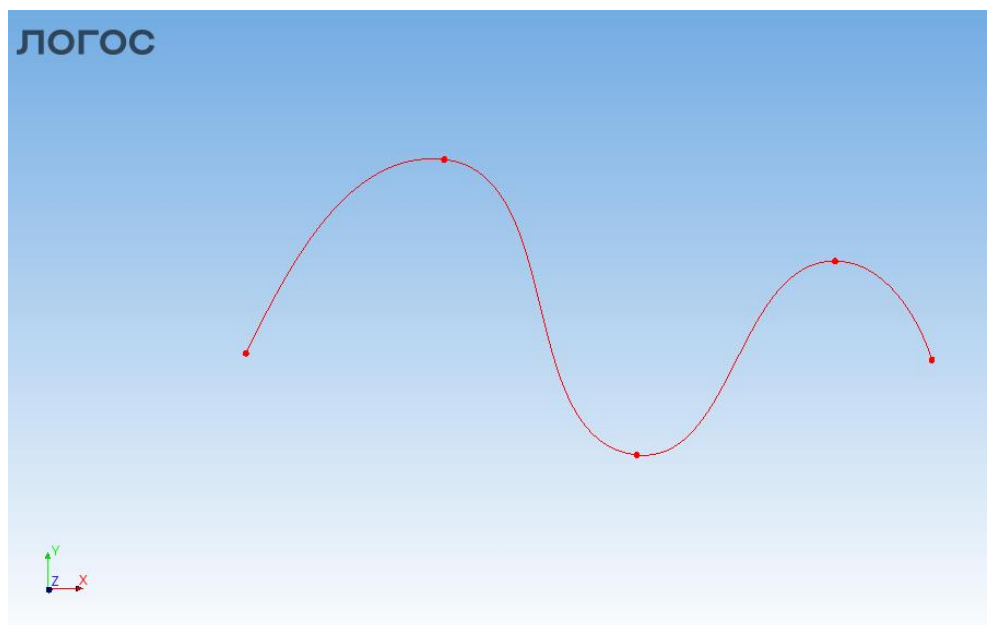


Рисунок 19 - Сплайн

После построения объектов из таблицы 3 перейдем к просмотру работы процедуры создания поверхности протягиванием.

Для вызова данной процедуры необходимо выделить элемент дерева обрабатываемой детали. Активируется модуль «Деталь» и на вспомогательной панели инструментов появятся функциональные кнопки по работе с геометрией детали. В дереве моделей необходимо перейти в пункт «Фрагмент» для соответствующей детали и перейти в режимы «Режим создания геометрических сущностей» - «Режим создания тел» и выбрать процедуру «Создание поверхности протягиванием» на вспомогательной панели инструментов. После запуска процедуры создания сплайна на панели мастера выполнения операций появится диалог создания поверхности протягиванием, который представлен на рисунке 20.

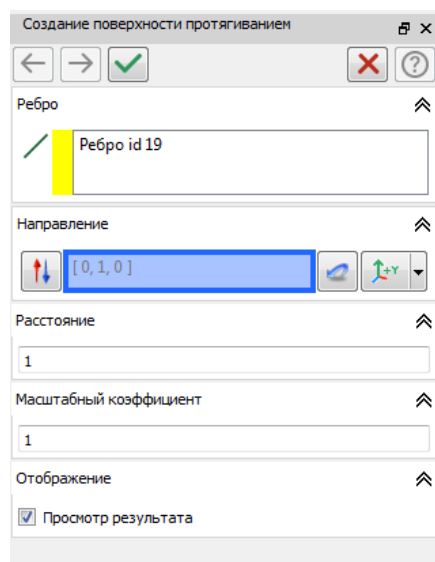


Рисунок 20 - Диалог создания поверхности протягиванием

В данном диалоге имеются пять активных элементов для задания параметров:

- Поле для отображения заданного ребра;
- Поле для выбора направления;
- Поле для расстояния;
- Поле для масштабного коэффициента;
- Режим включения/выключения просмотра результата (предварительного просмотра).

Для создания поверхности протягиванием необходимо с помощью левой кнопки мыши в окне визуализации выбрать ребро, в нашем случае это объект, представленный на рисунке 17. Задаем следующие параметры:

- направление вдоль оси Y;
- расстояние = 1;
- масштабный коэффициент = 1.

В процессе задания параметров в окне визуализации отобразится предварительная визуализация поверхности. Предварительный просмотр проведения операции с заданными параметрами представлен на рисунке 21.

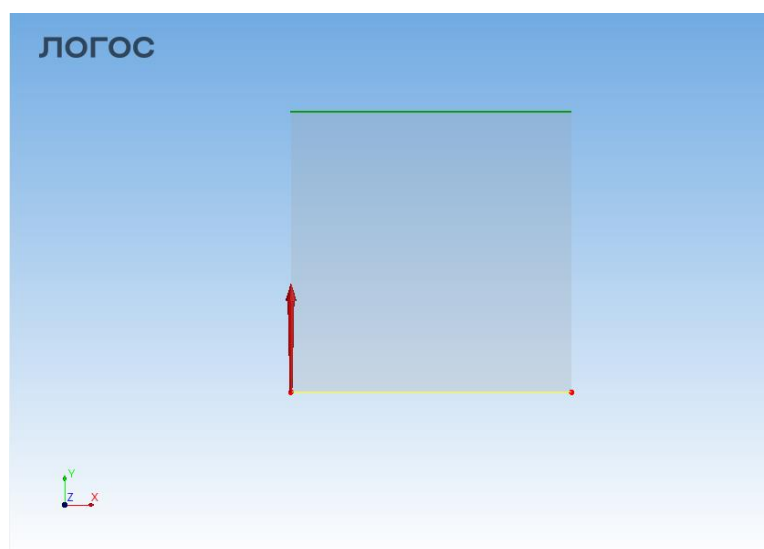


Рисунок 21 - Предварительный просмотр

Поменяем значение масштабного коэффициента сначала на 0.01, а затем на 2. Предварительный просмотр проведения операции с применением измененного масштабного коэффициента к исходному объекту можно увидеть на рисунках 22 (а-б).

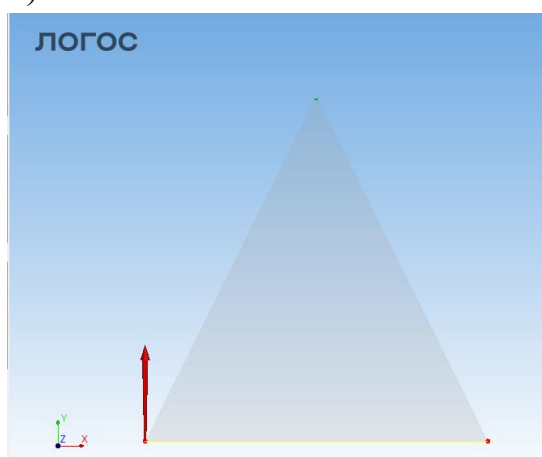


Рисунок 22 (а)

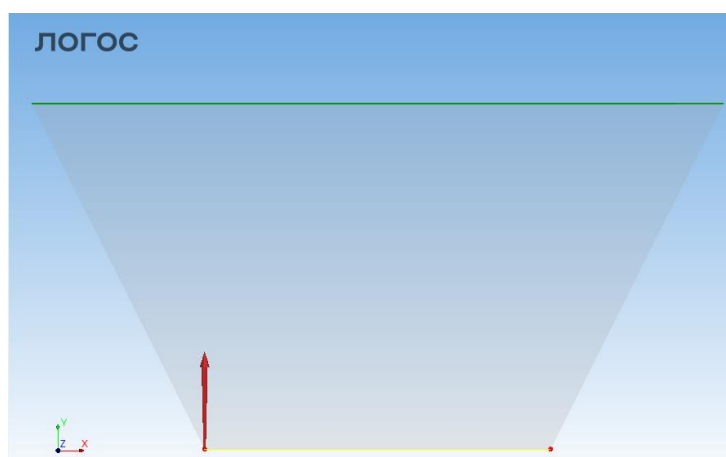


Рисунок 22 (б)

Рисунок 22 - Применение масштабного коэффициента 0.01 (а) и 2 (б) к исходному объекту

Применяем следующие параметры:

- направление вдоль оси Y;
- расстояние = 1;
- масштабный коэффициент = 1.

И нажмем кнопку «Выполнить операцию» в диалоговом окне. Полученный результат представлен на рисунке 23.

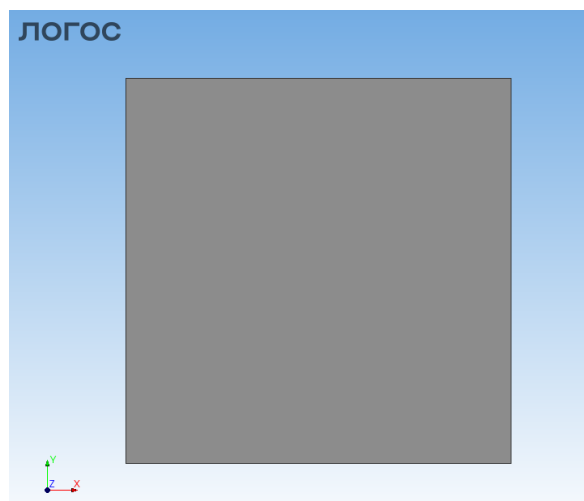


Рисунок 23 - Получение поверхности из ребра

В ПК ЛОГОС можно получить информацию о конечном объекте, при помощи обращения к операции «Информация о геометрии», за вызов которой отвечает класс `urpGeometryElementInfoDialog`. Результат работы данной операции представлен на рисунке 24.

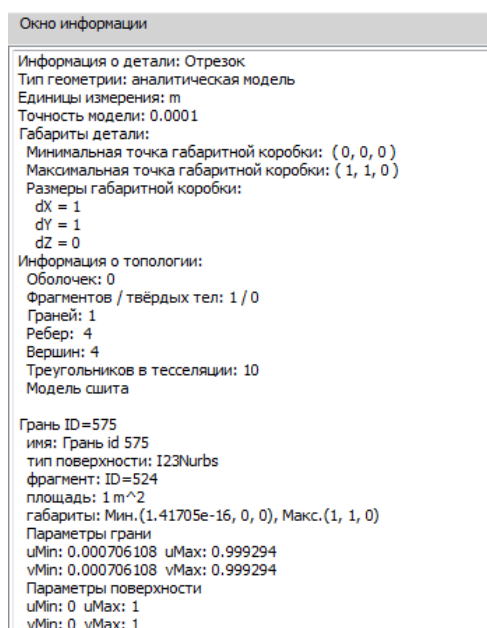


Рисунок 24 - Информация об объекте

Применим операцию к объекту ломаная линия. Геометрический объект «ломаная линия» возьмем из таблицы 3. Для этого выделим все ребра, составляющие наш объект, и зададим направление, например по оси Y. Предпросмотр и результат можно увидеть на рисунках 25-26.

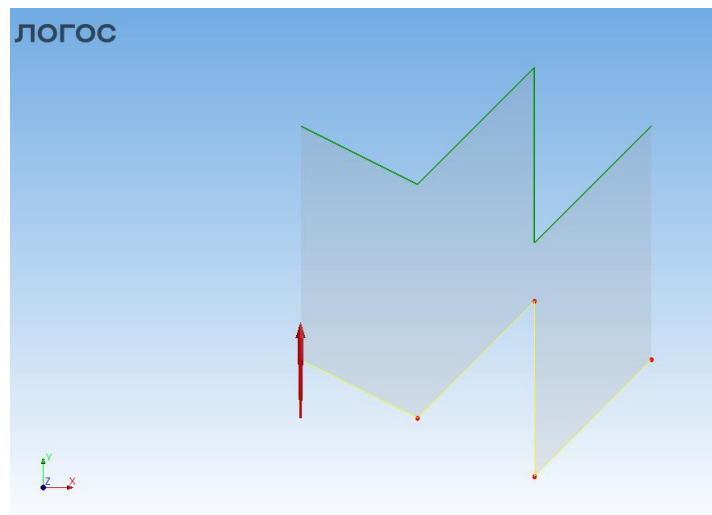


Рисунок 25 - Предпросмотр построения поверхности протягиванием ломаной линии

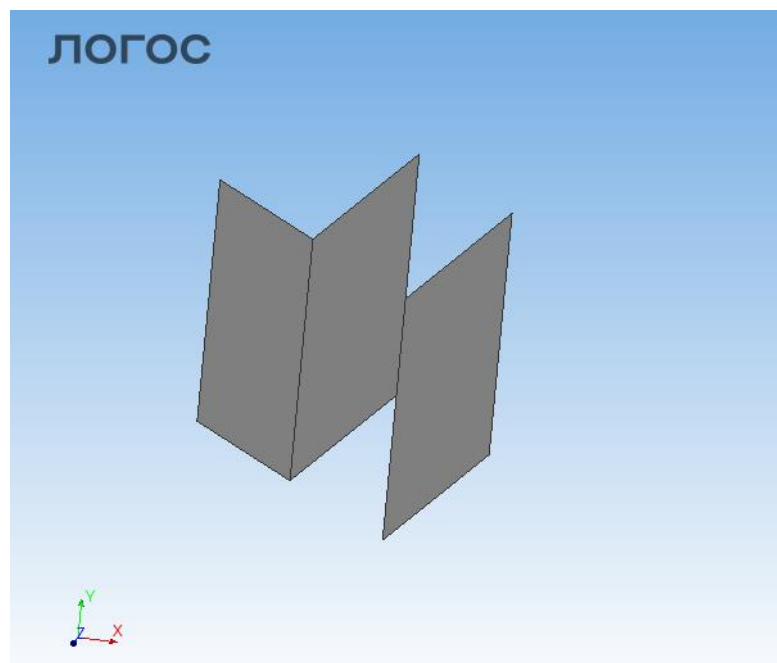


Рисунок 26 - Результат выполнения операции построения поверхности протягиванием ломаной линии

Применим операцию к объекту сплайн. Геометрический объект «сплайн» возьмем из таблицы 3. Выделим ребро, и зададим направление, по Z. Предпросмотр и результат можно увидеть на рисунках 27-28.

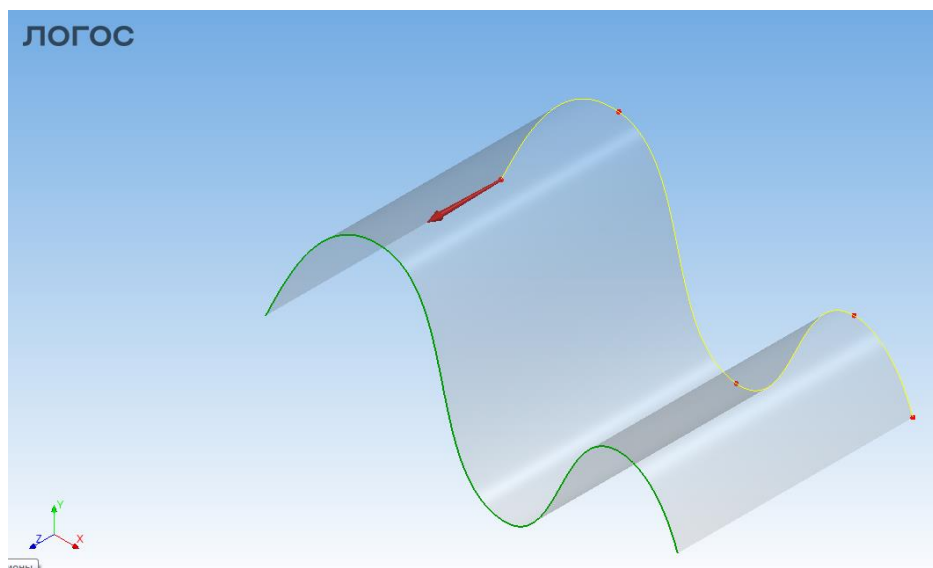


Рисунок 27 - Предпросмотр

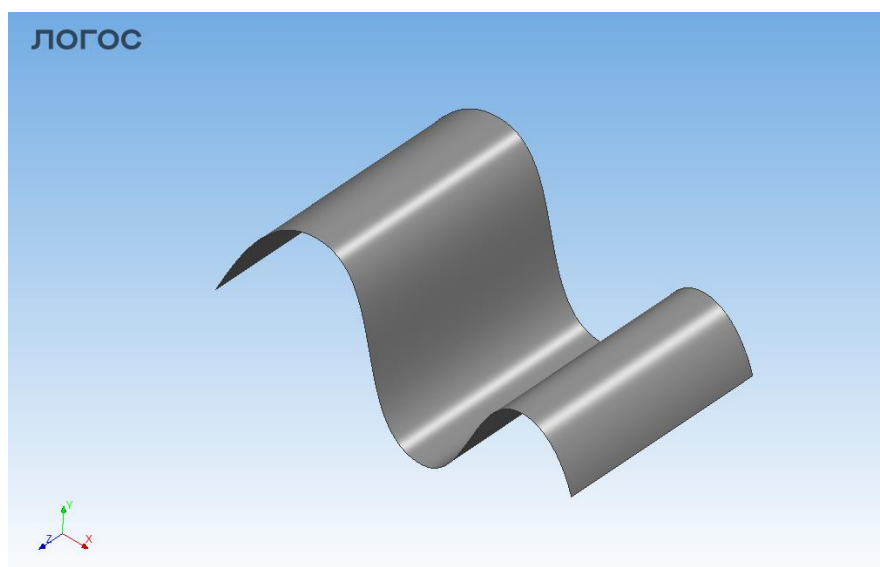


Рисунок 28 - Результат выполнения операции

3.3. Пример использования разработанного инструмента «Создание поверхности вращения»

Рассмотрим работу инструмента «Создание поверхности вращением вокруг заданной оси» в ПК ЛОГОС. Для вызова данной процедуры необходимо выделить элемент дерева обрабатываемой детали. Активируется модуль «Деталь» и на вспомогательной панели инструментов появятся функциональные кнопки по работе с геометрией детали. В дереве моделей необходимо перейти в пункт «Фрагмент» для соответствующей детали и перейти в режимы «Режим создания геометрических

сущностей» - «Режим создания тел» и выбрать процедуру «Создание поверхности вращением вокруг заданной оси» на вспомогательной панели инструментов. После запуска процедуры создания сплайна на панели мастера выполнения операций появится диалог создания поверхности вращением, который представлен на рисунке 29.

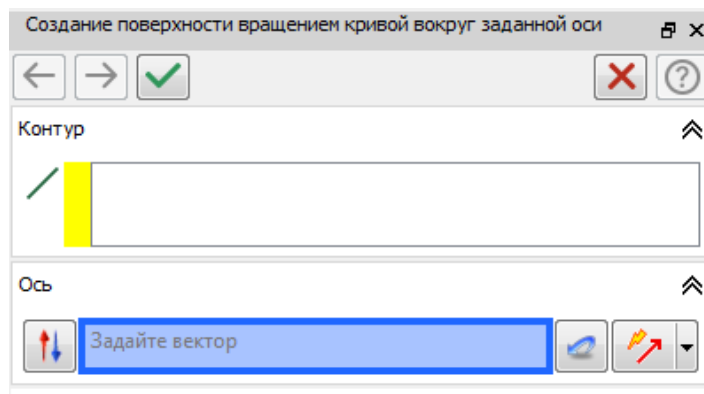


Рисунок 29 - Диалог создания поверхности вращением

В данном диалоге есть два активных элемента:

- Поле для отображения заданного ребра;
- Поле для выбора оси.

Для создания поверхности вращения необходимо с помощью левой кнопки мыши в окне визуализации выбрать ребро. Возьмем геометрический объект «ребро» из таблицы 3. Зададим ось вдоль Z. Для выполнения операции необходимо нажать на кнопку «Выполнить операцию», полученный результат представлен на рисунке 31.

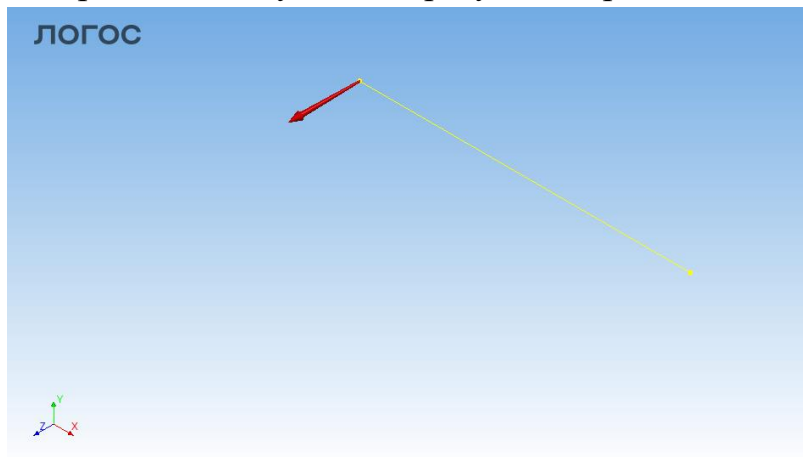


Рисунок 30 - Ребро и ось заданная вдоль Z

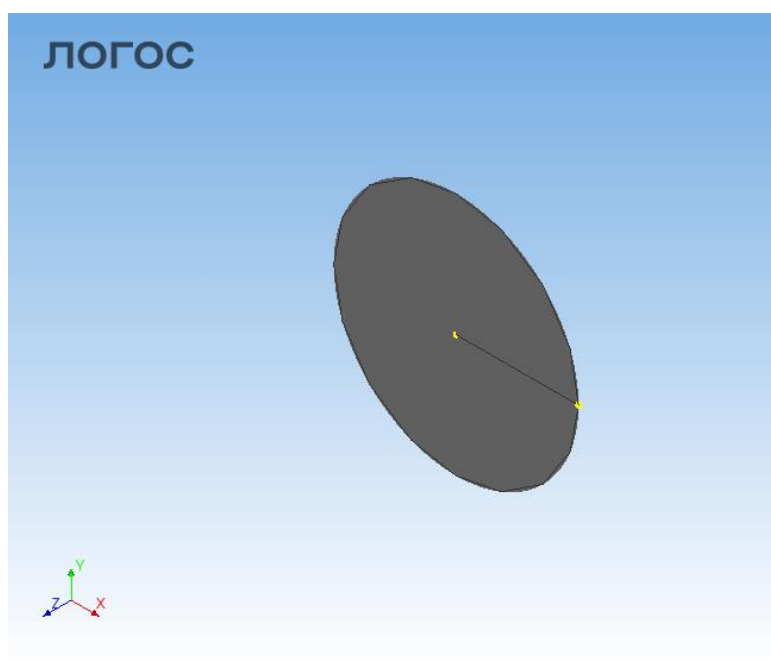


Рисунок 31 - Результат вращения вдоль оси Z

Для того же объекта зададим ось с произвольными координатами (3,1,2) и выполним операцию, результат представлен на рисунке 33.

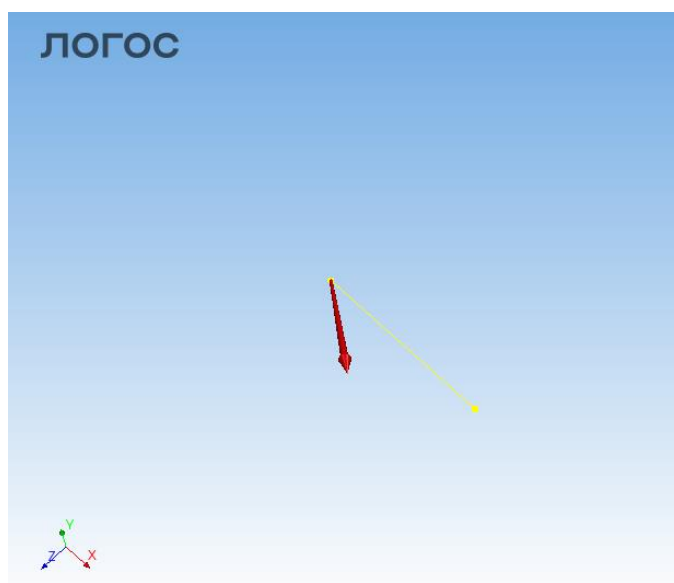


Рисунок 32 - Ребро и ось с произвольными координатами

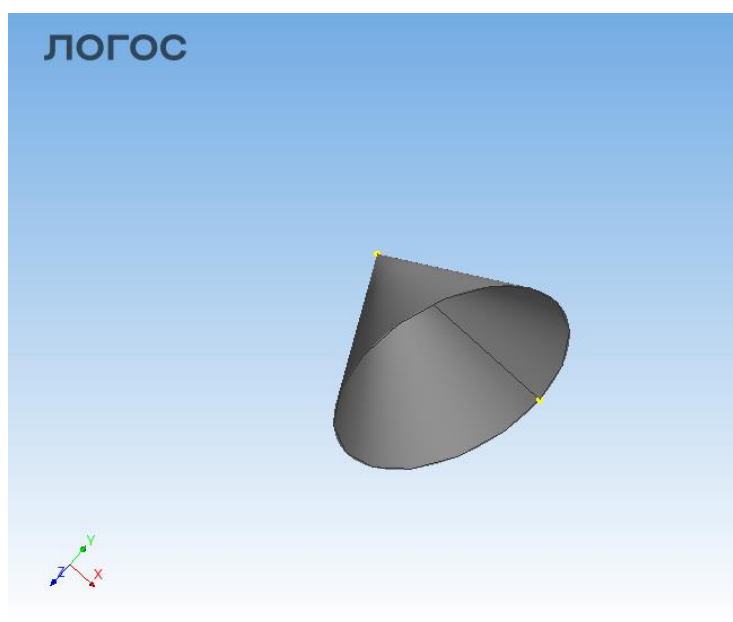


Рисунок 33 - Результат вращения вдоль оси с произвольными координатами

Применим операцию к объекту ломаная линия. Геометрический объект «ломаная линия» возьмем из таблицы 3. Для этого выделим все ребра, составляющие наш объект, и зададим направление вдоль оси Y, полученный результат можно увидеть на рисунках 34-35.

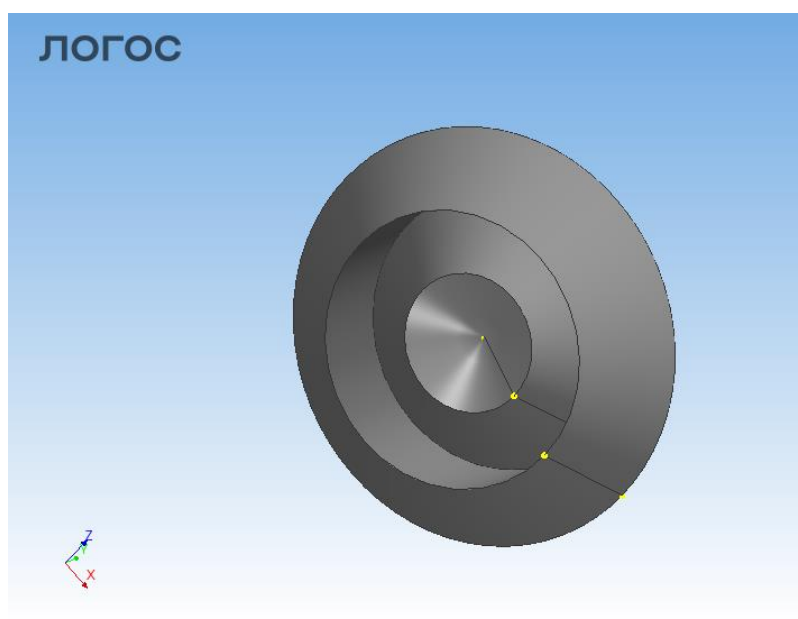


Рисунок 34 - Результат вращения ломаной вдоль оси Y

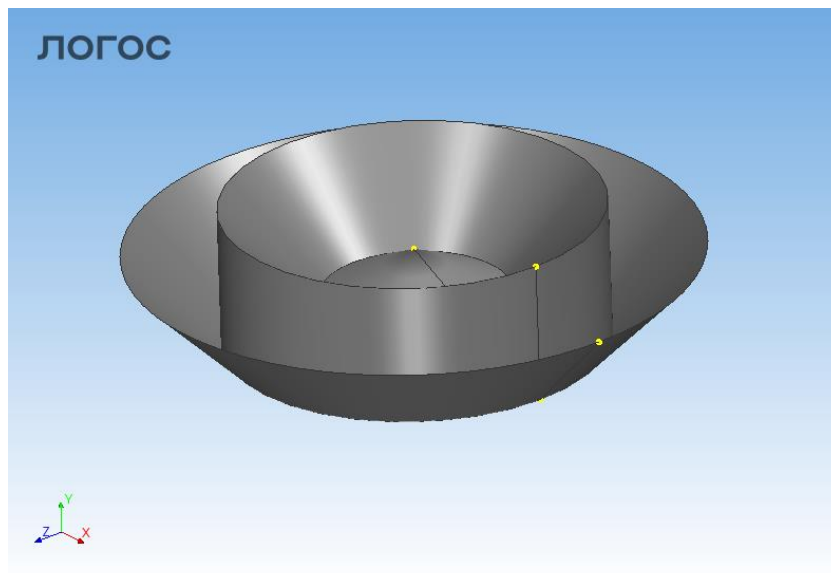


Рисунок 35 - Результат вращения ломаной вдоль оси Y

Можем выбрать отдельные ребра, составляющего ломаную линию и задать для каждого свою ось вращения. Результат представлен на рисунках 36-37.

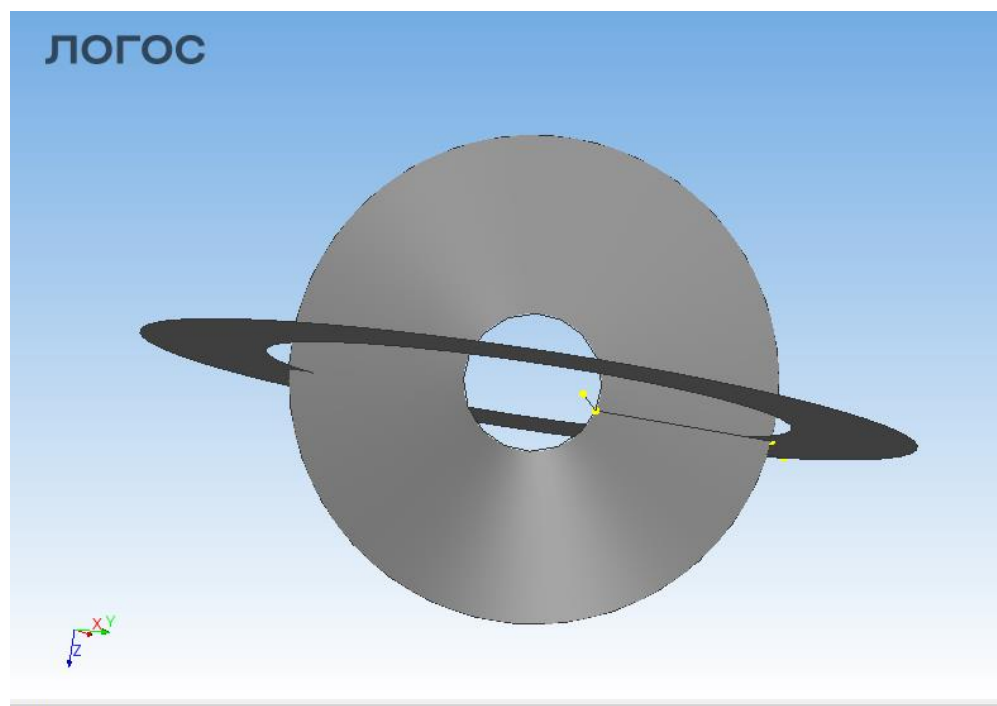


Рисунок 36 - Результат вращения заданных частей ломаной линии

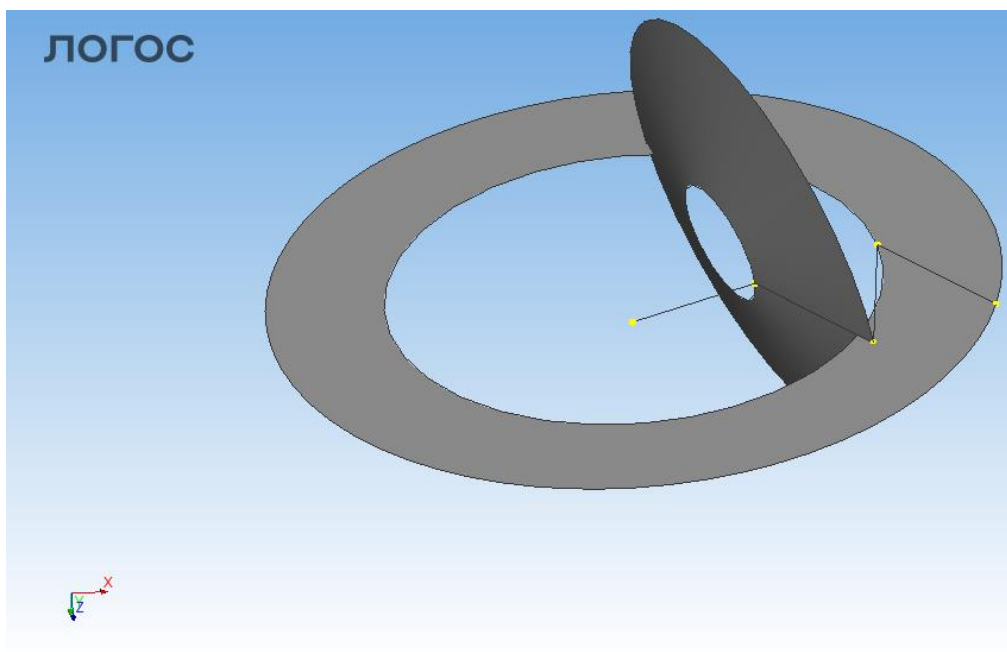


Рисунок 37 - Результат вращения заданных частей ломаной линии

4. Социальная ответственность

В настоящее время все больше внимания уделяется вопросам безопасности трудящихся на рабочем месте. Одной из главных задач является охрана здоровья трудящихся, ликвидация различных видов производственных травм и заболеваний.

С каждым годом все большее применение находят электронно-вычислительные машины (ЭВМ) как на производстве, так и для научно-исследовательских и конструкторских работ, а также в сфере управления и образования. Компьютеры уже завоевали свое место на предприятиях, в организациях, офисах и даже в домашних условиях. Однако компьютер является источником вредного воздействия на организм человека, а, следовательно, и источником профессиональных заболеваний. Это предъявляет к каждому пользователю персонального компьютера требование: знать о вредном воздействии ПЭВМ на организм человека и необходимых мерах защиты от этих воздействий.

4.1. Правовые и организационные вопросы обеспечения безопасности

4.1.1. Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства

В данной работе рассмотрена проектировка рабочего места и помещения, котором оно находится. Под проектированием рабочего места понимается целесообразное пространственное размещение в горизонтальной и вертикальной

плоскостях функционально взаимоувязанных средств производства (оборудования, оснастки, предметов труда и др.), необходимых для осуществления трудового процесса.

При проектировании рабочих мест должны быть учтены освещенность, температура, влажность, давление, шум, наличие вредных веществ, электромагнитных полей и другие санитарно-гигиенические требования к организации рабочих мест.

При проектировании рабочей зоны необходимо уделить внимание охране окружающей среды, а в частности, организации безотходного производства. Также необходимо учитывать возможность чрезвычайных ситуаций. Так как рабочая зона находится в городе Томске, наиболее типичной ЧС является мороз.

4.1.2. Организационные мероприятия при компоновке рабочей зоны

Характеристика помещения, где была разработана магистерская диссертация: ширина комнаты составляет $b = 4$ м, длина $a = 6$ м, высота $H = 2,8$ м. Тогда площадь помещения будет составлять $S = ab = 24\text{ м}^2$, объем $V = abh = 72\text{ м}^3$. В помещении имеется окно, через которое осуществляется вентиляция помещения. В помещении отсутствует принудительная вентиляция, т.е. воздух поступает и удаляется через дверь и окно, вентиляция является естественной. В зимнее время помещение отапливается, что обеспечивает достаточное, постоянное и равномерное нагревание воздуха. В помещении используется комбинированное освещение - искусственное и естественное. Искусственное освещение создается люминесцентными лампами типа ЛБ. Рабочая поверхность имеет высоту 0,75 м. Конструкция стола соответствует нормам СН 245-78. Стол оборудуется специальными ящиками с необходимыми для работы предметами. Электроснабжение сети переменного напряжения 220 В. Помещение без повышенной опасности в отношении поражения человека электрическим током по ГОСТ Р 12.1.019-2009.

Компьютер, расположенный на рабочей поверхности высотой 0.74 м, обладает следующими характеристиками: процессор Intel® Core™ i5-7300HQ, оперативная память 8 ГБ, система Microsoft Windows 10, частота процессора – 2,50 ГГц, MSI 15,6-и дюймовый монитор с разрешением 1366 на 768 точек и частотой 60 Гц.

При устройстве рабочего места человека, работающего за ПК необходимо соблюсти следующие основные условия: наилучшее местоположение оборудования и свободное рабочее пространство. Работа выполнена на персональном компьютере

типа ноутбук, в котором отсутствуют элементы стационарного ПК (системный блок, отдельный монитор, клавиатура и т.д.), таким образом основным требованием к организации рабочего места является размещение ноутбука по центру письменного стола строго напротив пользователя, т.к. это обеспечивает положение монитора ноутбука на уровне глаз оператора, а также комфортное положение рук оператора над клавиатурой.

Проектирование письменного стола должно учитывать следующие требования. Высота рабочей поверхности стола рекомендуется в пределах 680– 800 мм. Высота рабочей поверхности, на которую устанавливается клавиатура, должна быть 650 мм. Рабочий стол должен быть шириной не менее 700 мм и длиной не менее 1400 мм. Должно иметься пространство для ног высотой не менее 600 мм, шириной — не менее 500 мм, глубиной на уровне колен — не менее 450 мм и на уровне вытянутых ног — не менее 650 мм.

Рабочее кресло должно быть подъёмно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки до переднего края сиденья. Рекомендуется высота сиденья над уровнем пола 420– 550 мм. Конструкция рабочего кресла должна обеспечивать: ширину и глубину поверхности сиденья не менее 400 мм.

Монитор должен быть расположен на уровне глаз оператора на расстоянии 500–600 мм. Согласно нормам угол наблюдения в горизонтальной плоскости должен быть не более 45° к нормали экрана. Лучше если угол обзора будет составлять 30°. Кроме того должна быть возможность выбирать уровень контрастности и яркости изображения на экране. Должна предусматриваться возможность регулирования экрана.

4.2. Профессиональная социальная ответственность

4.2.1. Анализ вредных и опасных факторов, которые может создать объект исследования

Вредными называется производственный фактор, воздействие которого на сотрудника в определенных условиях приводит к заболеванию или снижению работоспособности. При изменении уровня и времени воздействия вредные производственные факторы могут стать опасными. Опасными считаются производственные факторы, воздействие которых на работающего в конкретных условиях может привести к травмам, а также другим внезапным резким ухудшениям здоровья.

Для идентификации потенциальных факторов использован ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация». Перечень опасных и вредных факторов, характерных для проектируемой производственной среды представлен в виде таблицы 4.

Таблица 4 - Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
1. Отклонение показателей микроклимата	+	+	+	СанПиН 2.2.2/2.4.1.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [22] СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» [23]
2. Недостаточная освещенность рабочей зоны	+	+	+	СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» [24]
3. Повышенный уровень шума	+	+	+	СанПиН 2.2.4/2.1.8.10-32-2002 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» [25]

4. Повышенное образование электростатических зарядов	+	+	+	СанПиН 2.2.2.542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ» [26]
5. Наличие электромагнитных полей радиочастотного диапазона	+	+	+	СанПиН 2.2.4.1191-03 «Электромагнитные поля в производственных условиях» [27]
6. Нервно-психические перегрузки, связанные с напряженностью трудового процесса	+	+	+	СанПиН 2.2.2/2.4.1.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [22]

Эти факторы могут привести к ухудшению здоровья пользователя, а также к профессиональным заболеваниям. Кроме того, вынужденная неудобная рабочая поза (в большинстве случаев в ограниченном пространстве), длительное сосредоточенное наблюдение, из которого 20% приходится на непосредственное наблюдение за экраном ВДТ, вызывают повышенное напряжение мышц зрительного аппарата, а в комплексе с неблагоприятными производственными факторами обуславливают развитие общего утомления и снижение работоспособности.

Анализ опасных и вредных производственных факторов

Отклонение показателей микроклимата

Микроклимат производственных помещений – это климат внутренней среды помещений, который определяется действующими на организм человека сочетаниями температур воздуха и поверхностей, относительной влажности воздуха, скорости движения воздуха и интенсивности теплового излучения. Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма.

Оптимальные микроклиматические при воздействии на человека в течение рабочей смены обеспечивают сохранение теплового состояния организма и не вызывают отклонений в состоянии здоровья. Допустимые микроклиматические условия могут приводить к незначительным дискомфортным тепловым ощущениям. Возможно временное (в течение рабочей смены) снижение работоспособности, без нарушения здоровья.

Нормы оптимальных и допустимых показателей микроклимата при работе с ЭВМ устанавливает СанПиН 2.2.2/2.4.1340-03 . Все категории работ разграничиваются на основе интенсивности энергозатрат организма в ккал/ч (Вт). Работа, производимая сидя и сопровождающаяся незначительным физическим напряжением, относится к категории Ia – работа с интенсивностью энергозатрат до 120 ккал/ч (до 139 Вт). Для данной категории допустимые нормы микроклимата представлены в таблице 5.

Таблица 5 - Допустимые нормы микроклимата в рабочей зоне производственных помещений

Сезон года	Категория тяжести выполняемых работ	Температура, С°		Относительная влажность, %		Скорость движения воздуха, м/сек	
		Фактическое значение	Допустимое значение	Фактическое значение	Допустимое значение	Фактическое значение	Допустимое значение
Холодный	Ia	(22÷24)	(19÷24)	55	(15÷75)	0,1	≤0,1

Теплый	Ia	(23÷25)	(20÷28)	55	(15÷75)	0,1	≤0,2
--------	----	---------	---------	----	---------	-----	------

Анализируя таблицу 5, можно сделать вывод, что в рассматриваемом помещении параметры микроклимата соответствуют нормам СанПиН. Допустимый уровень микроклимата помещения обеспечивается системой водяного центрального отопления и естественной вентиляцией.

Недостаточная освещенность рабочей зоны

Свет является естественным условием жизни человека. Правильно спроектированное и выполненное освещение обеспечивает высокий уровень работоспособности, оказывает положительное психологическое действие на человека и способствует повышению производительности труда. На рабочей поверхности должны отсутствовать резкие тени, которые создают неравномерное распределение поверхностей с различной яркостью в поле зрения, искажает размеры и формы объектов различия, в результате повышается утомляемость и снижается производительность труда.

Существует три вида освещения: естественное – за счёт солнечного излучения, искусственное – за счёт источников искусственного света и совмещенное – освещение, включающее в себя как естественное, так и искусственное освещения.

В данном рабочем помещении используется комбинированное освещение: искусственное и естественное. Искусственное освещение создается люминесцентными лампами типа ЛД.

Расчёт общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента светового потока, учитывающим световой поток, отражённый от потолка и стен. Длина помещения $a = 6$ м, ширина $b = 4$ м, высота $H = 2,8$ м. Высота рабочей поверхности над полом $h_p = 0,75$ м. Интегральным критерием оптимальности расположения светильников является величина λ , которая для люминесцентных светильников с защитной решёткой лежит в диапазоне 1,1–1,3.

Выбираем лампу дневного света ЛД-40, световой поток которой равен ФЛД = 2300 Лм. Выбираем светильники с люминесцентными лампами типа ОДОР-2-30. Этот светильник имеет две лампы мощностью 40 Вт каждая, длина светильника

равна 925 мм, ширина – 265 мм. На первом этапе определим значение индекса освещенности i .

$$i = \frac{S}{(a+b)h} \quad (62)$$

где S – площадь помещения;

h – расчетная высота подвеса светильника, м;

a и b – длина и ширина помещения, м.

Высота освещения над рабочей поверхностью :

$$h = H - h_p - h_c = 2,8 - 0,75 - 0,3 = 1,55 \quad (63)$$

где H – высота помещения

h_p - высота рабочей поверхности, м;

В результате проведенных расчетов, индекс освещенности i равен

$$i = \frac{S}{(a+b)h} = \frac{24}{(4+6) \cdot 1,55} = 1,5 \quad (64)$$

Расстояние между соседними светильниками или рядами определяется по формуле:

$$L = \lambda \cdot h = 1,1 \cdot 1,55 = 1,6 \text{ м} \quad (65)$$

Число рядов светильников в помещении:

$$Nb = \frac{b}{L} = \frac{4}{1,6} = 2,5 \approx 3 \quad (66)$$

Число светильников в ряду:

$$Na = \frac{a}{L} = \frac{6}{1,6} = 3,75 \approx 4 \quad (67)$$

Общее число светильников:

$$N = Na \cdot Nb = 4 \cdot 3 = 12 \quad (68)$$

Учитывая, что в каждом светильнике установлено две лампы, общее число ламп в помещении $N = 24$.

Расстояние от крайних светильников или рядов до стены определяется по формуле:

$$l = \frac{L}{3} = \frac{1,6}{3} = 0,53, м \quad (69)$$

Размещаем светильники в три ряда. План помещения и размещения светильников с люминесцентными лампами представлен на рисунке

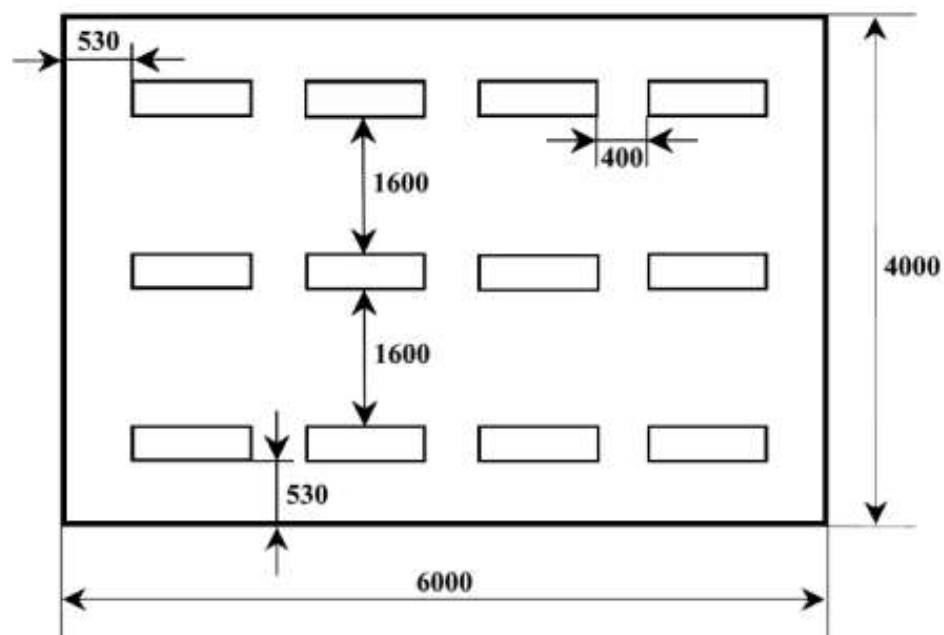


Рисунок – 38. План размещения светильников с люминесцентными лампами (размер указан в мм)

Световой поток лампы определяется по формуле:

$$\Phi = \frac{E_n \cdot S \cdot K_z \cdot Z}{N \cdot \eta} \quad (70)$$

где E_H – нормируемая минимальная освещенность по СНиП 23-05-95, лк;

S – площадь освещаемого помещения, м²;

K_z – коэффициент запаса, учитывающий загрязнение светильника (источника света, светотехнической арматуры, стен и пр., т. е. отражающих поверхностей), наличие в атмосфере цеха дыма, пыли;

Z – коэффициент неравномерности освещения, отношение $E_{ср} / E_{min}$. Для люминесцентных ламп при расчётах берётся равным 1,1;

N – число ламп в помещении;

η – коэффициент использования светового потока.

Данное помещение относится к типу помещения со средним выделением пыли, в связи с этим $K_z = 1,5$; состояние потолка – свежепобеленный, поэтому значение коэффициента отражения потолка $\rho_n = 70\%$; состояние стен – побеленные бетонные стены, поэтому значение коэффициента отражения стен $\rho_c = 50\%$. Коэффициент использования светового потока, показывающий какая часть светового потока ламп попадает на рабочую поверхность, для светильников типа ОДОР с люминесцентными лампами при $\rho_n = 70\%$, $\rho_c = 50\%$ и индексе помещения $i = 1,5$ равен $\eta = 0,47$.

Нормируемая минимальная освещенность при использовании ЭВМ и одновременной работе с документами должна быть равна 600лк.

$$\Phi = \frac{E_n \cdot S \cdot K_z \cdot Z}{N \cdot \eta} = \frac{600 \cdot 24 \cdot 1,5 \cdot 1,1}{24 \cdot 0,47} = 2106 \text{ Лм}$$

Для люминесцентных ламп с мощностью 40Вт и напряжением сети 220В, стандартный световой поток ЛД равен 2300 Лм.

$$-10\% \leq \frac{\Phi_{л\delta} - \Phi_{л.расч}}{\Phi_{л\delta}} \cdot 100\% \leq 20\%$$

$$\frac{2300 - 2106}{2300} \cdot 100\% = 8,43\%$$

$$-10\% \leq 8,43\% \leq 20\%$$

Таким образом необходимый световой поток светильника не выходит за пределы требуемого диапазона.

Наличие электромагнитных полей радиочастотного диапазона

ЭМП обладает способностью биологического, специфического и теплового воздействия на организм человека, что может повлечь следующие последствия: биохимические изменения в клетках и тканях; нарушения условно-рефлекторной деятельности, снижение биоэлектрической активности мозга, изменения межнейронных связей, отклонения в эндокринной системе; вследствие перехода ЭМП в тепловую энергию может наблюдаться повышение температуры тела, локальный избирательный нагрев тканей и так далее.

Согласно СанПиН 2.2.2.542-96:

1. Напряженность электромагнитного поля на расстоянии 50 см вокруг ВДТ по электрической составляющей должна быть не более:
 - в диапазоне частот 5Гц-2кГц - 25В/м;
 - в диапазоне частот 2кГц/400кГц - 2,5В/м.
2. Плотность магнитного потока должна быть не более:
 - в диапазоне частот 5Гц-2кГц - 250нТл;
 - в диапазоне частот 2кГц/400кГц - 25нТл.

Защита человека от опасного воздействия электромагнитного излучения осуществляется следующими способами:

1. Применение СКЗ

- защита временем;
 - защита расстоянием;
 - снижение интенсивности излучения непосредственно в самом источнике излучения;
 - экранирование источника;
 - защита рабочего места от излучения;
2. Применение средств индивидуальной защиты (СИЗ), которые включают в себя
- Очки и специальная одежда, выполненная из металлизированной ткани (кольчуга). При этом следует отметить, что использование СИЗ возможно при кратковременных работах и является мерой аварийного характера. Ежедневная защита обслуживающего персонала должна обеспечиваться другими средствами.
 - Вместо обычных стекол используют стекла, покрытые тонким слоем золота или диоксида олова (SnO_2).

Повышенное образование электростатических зарядов

Электризация заключается в следующем: нейтральные тела, в нормальном состоянии не проявляющие электрических свойств, при условии отрицательных контактов или взаимодействий становятся электрозаряженными. Опасность возникновения статического электричества проявляется в возможности образования электрической искры и вредном воздействии его на человеческий организм, и не только в случае непосредственного контакта с зарядом, но и за счет действий электрического поля, которое возникает при заряде. При включенном питании компьютера на экране дисплея накапливается статическое электричество. Электрический ток искрового разряда статического электричества мал и не может вызвать поражение человека. Тем не менее, вблизи экрана электризуется пыль и оседает на нем. В результате чего искажается резкость восприятия информации на экране. Кроме того, пыль попадает на лицо работающего и в его дыхательные пути.

Повышенный уровень шума

Предельно допустимый уровень (ПДУ) шума – это уровень фактора, который при ежедневной (кроме выходных дней) работе, но не более 40 часов в неделю в течение всего рабочего стажа, не должен вызывать заболеваний или отклонений в состоянии здоровья, обнаруживаемых современными методами исследований в

процессе работы или в отдаленные сроки жизни настоящего и последующих поколений. Соблюдение ПДУ шума не исключает нарушения здоровья у сверхчувствительных лиц.

Допустимый уровень шума ограничен ГОСТ 12.1.003-83 и СанПиН 2.2.4/2.1.8.10-32-2002. Уровень шума на рабочем месте математиков-программистов и операторов видеоматериалов не должен превышать 50дБА, а в залах обработки информации на вычислительных машинах - 65дБА.

При значениях выше допустимого уровня необходимо предусмотреть СКЗ и СИЗ.

1. СКЗ

- устранение причин шума или существенное его ослабление в источнике образования;
 - изоляция источников шума от окружающей среды средствами звуко- и виброизоляции, звукопоглощения;
 - применение средств, снижающих шум на пути их распространения;

2. СИЗ

- применение спецодежды, спецобуви и защитных средств органов слуха: наушники, беруши, антифоны.

Нервно-психические перегрузки, связанные с напряженностью трудового процесса

Значительное умственное напряжение и другие нагрузки приводят к переутомлению функционального состояния центральной нервной системы, нервно-мышечного аппарата рук. Нерациональное расположение элементов рабочего места вызывает необходимость поддержания вынужденной рабочей позы. Длительный дискомфорт вызывает повышенное позвоночное напряжение мышц и обуславливает развитие общего утомления и снижение работоспособности.

При длительной работе за экраном дисплея появляется выраженное напряжение зрительного аппарата с появлением жалоб на неудовлетворительность работы, головные боли, усталость и болезненное ощущение в глазах, в пояснице, в области шеи, руках.

Режим труда и отдыха работника: при вводе данных, редактировании программ, чтении информации с экрана непрерывная продолжительность работы не должна превышать 4-х часов при 8-часовом рабочем дне. Через каждый час работы необходимо делать перерыв на 5-10 минут, а через два часа на 15 минут.

4.2.2. Обоснование мероприятий по защите персонала предприятия от действия опасных и вредных

Электробезопасность представляет собой систему организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статистического электричества.

Электроустановки классифицируют по напряжению: с номинальным напряжением до 1000 В (помещения без повышенной опасности), до 1000 В с присутствием агрессивной среды (помещения с повышенной опасностью) и свыше 1000 В (помещения особо опасные).

В отношении опасности поражения людей электрическим током различают:

1. Помещения без повышенной опасности, в которых отсутствуют условия, создающие повышенную или особую опасность.
2. Помещения с повышенной опасностью, которые характеризуются наличием в них одного из следующих условий, создающих повышенную опасность: сырость, токопроводящая пыль, токопроводящие полы (металлические, земляные, железобетонные, кирпичные и т.п.), высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям, технологическим аппаратам, с одной стороны, и к металлическим корпусам электрооборудования - с другой.
3. Особо опасные помещения, которые характеризуются наличием оборудования свыше 1000 В и одного из следующих условий, создающих особую опасность: особой сырости, химически активной или органической среды, одновременно двух или более условий повышенной опасности.

Территории размещения наружных электроустановок в отношении опасности поражения людей электрическим током приравниваются к особо опасным помещениям.

Помещение, где была разработана магистерская диссертация, принадлежит к категории помещений без повышенной опасности по степени вероятности поражения электрическим током, вследствие этого к оборудованию предъявляются следующие требования:

- экран монитора должен находиться на расстоянии не менее 50 см от пользователя (расстояния от источника);
- применение приэкранных фильтров, специальных экранов.

Защитное заземление — это преднамеренное электрическое соединение с землей или ее эквивалентом металлических нетоковедущих частей, которые могут оказаться под напряжением.

Сопротивление заземления – основной показатель заземляющего устройства, определяющий его способность выполнять свои функции и определяющий его качество в целом.

Сопротивление заземления зависит от площади электрического контакта заземлителя (заземляющих электродов) с грунтом (“стекание” тока) и удельного электрического сопротивления грунта, в котором смонтирован этот заземлитель (“впитывание” тока). Согласно ПЭУ, изложенным в ГОСТ 12.1.030-81 номинальное сопротивление заземления должно быть не более 4 Ом, ток не более 0,1 А и напряжение 12-36 В.

К основным электрозащитным средствам в электроустановках напряжением до 1000 В относятся:

- изолирующие штанги;
- изолирующие и электроизмерительные клещи;
- диэлектрические перчатки; изолированный инструмент.

Работать со штангой разрешается только специально обученному персоналу в присутствии лица, контролирующего действия работающего. При операциях с изолирующей штангой необходимо пользоваться дополнительными изолирующими защитными средствами – диэлектрическими перчатками и изолирующими основаниями (подставками, ковриками) или диэлектрическими ботами.

Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в отсутствии видимых повреждений аппаратуры и рабочей мебели, в отсутствии повреждений и наличии заземления приэкранного фильтра.

В производственных помещениях, где допустимые нормативные величины микроклимата поддерживать не представляется возможным, необходимо проводить мероприятия по защите работников от возможного перегрева и охлаждения. Это достигается различными средствами: применением систем местного кондиционирования воздуха; использованием индивидуальных средств защиты от повышенной или пониженной температуры; регламентацией периодов работы в неблагоприятном микроклимате. и отдыха в помещении с микроклиматом, нормализующим тепловое состояние; сокращением рабочей смены и др.

Профилактика перегрева работников в нагревающем микроклимате включает следующие мероприятия: нормирование верхней границы внешней термической нагрузки на допустимом уровне применительно к 8-часовой рабочей смене; регламентация продолжительности воздействия нагревающей среды (непрерывно и за рабочую смену) для поддержания среднесменного теплового состояния на оптимальном или допустимом уровне; использование специальных СКЗ и СИЗ, уменьшающих поступление тепла извне к поверхности тела человека и обеспечивающих допустимое тепловое состояние работников.

Основные способы защиты от статического электричества следующие: заземление оборудования, увлажнение окружающего воздуха. Также целесообразно применение полов из антистатического материала.

Защита от шумов – заключение вентиляторов в защитный кожух и установление их внутри корпуса ЭВМ. Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами с максимальными коэффициентами звукопоглощения в области частот 63 - 8000 Гц.

Материалами для звукоизоляции могут являться:

- Пенополистерол
- Поролоновые маты
- Пробковые полотна и плиты

С целью снижения или устранения нервно-психологического, зрительного и мышечного напряжения, предупреждение переутомления необходимо проводить комплекс физических упражнений и сеансы психофизической разгрузки и снятия усталости во время регламентируемых перерывов, и после окончания рабочего дня.

4.3. Экологическая безопасность

4.3.1. Анализ влияния объекта исследования на окружающую среду

Охрана окружающей среды – это комплексная проблема и наиболее активная форма её решения - это сокращение вредных выбросов промышленных предприятий через полный переход к безотходным или малоотходным технологиям производства.

С точки зрения потребления ресурсов компьютер потребляет сравнительно небольшое количество электроэнергии, что положительным образом сказывается на общей экономии потребления электроэнергии в целом.

Основными отходами являются черновики бумаги, картриджи и отработавшие люминесцентные лампы.

4.3.2. Обоснование мероприятий по защите окружающей среды

Основным источником вреда для окружающей среды являются отходы, полученные при проведении работ. Их необходимо утилизировать. Так как израсходованная бумага не содержала никаких закрытых сведений, она была направлена на утилизацию без использования shreddera, а люминесцентные лампы собраны и направлены на утилизацию в соответствующую организацию. Израсходованные картриджи аналогично были переданы производителю для централизованной утилизации в соответствии с требованиями ГОСТ 30775-2001.

4.4. Безопасность в чрезвычайной ситуации

4.4.1. Анализ вероятных ЧС, которые может инициировать объект исследований

Так как данная работы выполняется на компьютере, то наиболее вероятной ЧС является возникновение пожара в помещении. По взрывопожарной и пожарной опасности помещения подразделяются на категории А, Б, В1 - В4, Г и Д, а здания - на категории А, Б, В, Г и Д. По пожарной опасности наружные установки подразделяются на категории А_н, Б_н, В_н, Г_н и Д_н.

Согласно НПБ 105-03 класс или офисное помещение относится к категории В - горючие и трудногорючие жидкости, твердые горючие и трудногорючие вещества и материалы (в том числе пыли и волокна), вещества и материалы, способные при взаимодействии с водой, кислородом воздуха или друг с другом только гореть, при условии, что помещения, в которых они имеются в наличии или обращаются, не относятся к категориям А или Б. По степени огнестойкости данное помещение относится к 1-й степени огнестойкости по СНиП 2.01.02-85 (выполнено из кирпича,

которое относится к трудно сгораемым материалам). Возникновение пожара при работе с электронной аппаратурой может быть по причинам как электрического, так и неэлектрического характера. Основной причиной возникновения пожара неэлектрического характера в офисном помещении может стать халатное неосторожное обращение с огнем (курение, оставленные без присмотра нагревательные приборы, использование открытого огня). Причины возникновения пожара электрического характера: короткое замыкание, перегрузки по току, искрение и электрические дуги, статическое электричество и т. п.

Однако пожар, это не единственная вероятная ЧС. В Томске преобладает континентально-циклонический климат. Природные явления (землетрясения, наводнения, засухи, ураганы и т. д.) отсутствуют. Возможными ЧС могут быть сильные морозы и диверсия.

Для Сибири в зимнее время года характерны морозы. Достижение критически низких температур приведет к авариям систем теплоснабжения и жизнеобеспечения, приостановке работы, обморожениям и даже жертвам среди населения. Также в летне-осенний период возможны ухудшения погоды (грозы с сопутствующим ураганным ветром). Чрезвычайные ситуации, возникающие в результате диверсий, возникают все чаще. Зачастую такие угрозы оказываются ложными. Но случаются взрывы и в действительности.

4.4.2. Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС

Согласно общим требованиям пожарной безопасности по ГОСТ 12.1.004-91 для устранения причин возникновения пожаров в помещении должны проводиться следующие мероприятия:

1. использование только исправного оборудования;
2. проведение периодических инструктажей по пожарной безопасности;
3. назначение ответственного за пожарную безопасность помещений;
4. издание приказов по вопросам усиления пожарной безопасности;
5. отключение электрооборудования, освещения и электропитания по окончании работ;
6. курение в строго отведенном месте;
7. содержание путей и проходов для эвакуации людей в свободном состоянии.

Для локализации или ликвидации возгорания на начальной стадии используются первичные средства пожаротушения. Первичные средства пожаротушения обычно применяют до прибытия пожарной команды.

Воздушно-пенные огнетушители очагов пожара, без наличия электроэнергии. Углекислотные и порошковые огнетушители предназначены для тушения электроустановок, находящихся под напряжением до 1000 В. Кроме того, порошковые применяют для тушения документов.

Для тушения токоведущих частей и электроустановок применяется переносной порошковый закачной огнетушитель ОП-3. Тушение электроустановок нужно производить на расстоянии не менее 1 метра (имеется в виду расстояние от сопла огнетушителя до токоведущих частей). Зарядку порошковых огнетушителей следует производить один раз в пять лет. При возникновении необходимости ремонта или зарядки, следует обращаться в специализированные фирмы.

В общественных зданиях и сооружениях на каждом этаже должно размещаться не менее двух переносных огнетушителей. Огнетушители следует располагать на видных местах вблизи от выходов из помещений на высоте не более 1,35 м. Размещение первичных средств пожаротушения в коридорах, переходах не должно препятствовать безопасной эвакуации людей. Здание должно соответствовать требованиям пожарной безопасности, а именно, наличие охранно-пожарной сигнализации, плана эвакуации, порошковых огнетушителей с поверенным клеймом, табличек с указанием направления к запасному (эвакуационному) выходу (приложение Б).

В случае переморозки труб должны быть предусмотрены запасные обогреватели. Их количества и мощности должно хватать для того, чтобы работа на производстве не прекратилась. Кроме того, необходимо иметь альтернативные источники тепла, электроэнергии и транспорта.

Для защиты от грозы и ураганного ветра рекомендуется следить за сводками погоды и одеваться соответствующе при холодной погоде с сильными осадками, если же штормовое предупреждение обещает наличие шквального ветра, рекомендуется в таких ситуациях не покидать надежных укрытий и пережить стихийное бедствие.

Для предупреждения вероятности осуществления диверсии предприятие необходимо оборудовать системой видеонаблюдения, круглосуточной охраной, пропускной системой, надежной системой связи, а также исключения распространения информации о системе охраны объекта, расположении помещений и оборудования в помещениях, системах охраны, сигнализаторах, их местах установки и количестве. Должностные лица раз в полгода проводят тренировки по отработке действий на случай экстренной эвакуации.

4.5. Выводы и рекомендации

Проанализировав условия труда на рабочем месте, где была разработана магистерская диссертация, можно сделать вывод, что помещение удовлетворяет необходимым нормам и в случае соблюдения техники безопасности и правил пользования компьютером работа в данном помещении не приведет к ухудшению здоровья работника.

Само помещение и рабочее место в нем удовлетворяет всем нормативным требованиям. Кроме того, действие вредных и опасных факторов сведено к минимуму, т.е. микроклимат, освещение и электробезопасность соответствуют требованиям, предъявленным в соответствующих нормативных документах.

Относительно рассмотренного вопроса об экологической безопасности можно сказать, что деятельность помещения не представляет опасности окружающей среде.

Важно добавить, что монитор компьютера служит источником ЭМП – вредного фактора, который отрицательно влияет на здоровье работника при продолжительной непрерывной работе и приводит к снижению работоспособности. Поэтому во избежание негативного влияния на здоровье необходимо делать перерывы при работе с ЭВМ и проводить специализированные комплексы упражнений для глаз.

5. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1. Потенциальные потребители результатов исследования

Для анализа потребителей результатов исследования необходимо рассмотреть целевой рынок и провести его сегментирование.

Целевой рынок – сегменты рынка, на котором будет продаваться в будущем разработка. В свою очередь, сегмент рынка – это особым образом выделенная часть рынка, группы потребителей, обладающих определенными общими признаками.

Сегментирование – это разделение покупателей на однородные группы, для каждой из которых может потребоваться определенный товар (услуга).

Потенциальные потребители: частные компании, государственные компании, научно-исследовательские институты

		Виды использования программного комплекса	
Виды потребителей		Разработка продукции	Научные исследования
	Научно-исследовательские институты		
	Частные компании		
	Государственные компании		
Фирма А			
Фирма Б			

Рисунок 39 – Карта сегментирования рынка услуг по разработке программного комплекса.

5.2. Анализ конкурентных технических решений

Детальный анализ конкурирующих разработок, существующих на рынке необходимо проводить систематически, поскольку рынки пребывают в постоянном движении. Такой анализ помогает вносить коррективы в научное исследование, чтобы успешнее противостоять своим соперникам. Важно реалистично оценить сильные и слабые стороны разработок конкурентов.

С этой целью может быть использована вся имеющаяся информация о конкурентных разработках:

- технические характеристики разработки;
- конкурентоспособность разработки;
- уровень завершенности научного исследования (наличие макета, прототипа и т.п.);
- бюджет разработки;
- уровень проникновения на рынок;
- финансовое положение конкурентов, тенденции его изменения и т.д.

Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения позволяет провести оценку

сравнительной эффективности научной разработки и определить направления для ее будущего повышения.

Основными конкурентами зарубежные компании по созданию инженерно-технических программных комплексов или отечественные компании, работающие над аналогичными разработками. Составим оценочную карту для сравнения конкурентных технических решений.

Таблица 6 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Бф	Бк1	Бк2	Кф	Кк1	Кк2
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Повышение производительности труда потребителя	0,1	5	5	4	0,5	0,5	0,4
2. Потребность в мощном вычислительном оборудовании	0,07	4	5	4	0,28	0,35	0,28
3. Точность вычислений	0,1	5	5	4	0,5	0,5	0,4
4. Качество интеллектуального интерфейса	0,09	4	5	4	0,36	0,45	0,36
5. Функциональная мощность (предоставляемые возможности)	0,11	5	5	4	0,55	0,55	0,44
Экономические критерии оценки эффективности							
1. Конкурентоспособность продукта	0,1	5	5	4	0,5	0,5	0,4
2. Перспективность продукта	0,12	5	5	4	0,6	0,6	0,48
3. Цена	0,1	4	2	4	0,4	0,2	0,4
4. Послепродажное	0,11	4	4	4	0,44	0,44	0,44

обслуживание							
5. Финансирование научной разработки	0,1	5	5	3	0,5	0,5	0,3
Итого	1	46	45	38	4,63	4,59	3,9

Критерии для сравнения и оценки ресурсоэффективности и ресурсосбережения, приведенные в табл. 1, подбираются, исходя из выбранных объектов сравнения с учетом их технических и экономических особенностей разработки, создания и эксплуатации.

Позиция разработки и конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum V_i \cdot B_i, \quad (71)$$

где K – конкурентоспособность научной разработки или конкурента;

V_i – вес показателя (в долях единицы);

B_i – балл i -го показателя.

Основываясь на знаниях о конкурентах, следует объяснить, что, не смотря на то, что иностранные компании, работающие в данной сфере, имеют большой опыт в разработке пакетов инженерного анализа, а также хорошо финансируемы и обеспечивают свою продукцию качественной поддержкой, имеют один большой недостаток – это высокая стоимость лицензионной продукции, которая может превышать стоимость нашей разработки в 5-7 раз. В нашем случае за меньшую цену предоставляется не менее качественный продукт, который ненамного уступает иностранным производителям, а в некоторых случаях даже превосходит их.

5.3. SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта.

Разработанная для данного исследования матрица SWOT представлена в таблице 7.

Таблица 7 – Матрица SWOT

	Сильные стороны научно-исследовательского проекта	Слабые стороны научно-исследовательского проекта
	1. Широкий инструментарий 2. Ориентированность на отечественных пользователей 3. Высокая точность вычислений	1. Необходимость в мощном вычислительном оборудовании 2. Первоначальные сложности с освоением пакета инженерного анализа
Возможности	Благодаря тому, что разработка ориентирована на отечественных потребителей, это позволит пользоваться разработкой, как частным отечественным фирмам, так и крупным государственным предприятиям.	Из-за ориентированности на отечественных потребителей, отсутствует возможность развития за границей.
1. Многообразие операций для решения различных задач инженерного анализа 2. Возможность использования русскоязычного интерфейса удобного для отечественного пользователя 3. Возможность получения результатов анализа с высокой точностью		
Угрозы	Некоторые зарубежные фирмы имеют более широкий список операций, но их главным недостатком является слишком большая цена. К тому же для разработки государственных заказов отечественная продукция будет предпочтительнее, чтобы данная проблема возникла, на рынке должен появиться крупный отечественный разработчик способный выдержать конкуренцию, но обычно такие разработки	В связи с тем, что это трудный наукоемкий проект, который нуждается в постоянной поддержке и доработке, а также очень сильно зависит от технического обеспечения, сокращение финансирования может сильно повлиять на дальнейшее развитие.
1. Появление конкурента с более широким выбором инструментария при такой же цене продукции 2. Сокращение финансирования на дальнейшую разработку проекта		

	занимают много времени, поэтому в ближайшее время эта угроза не актуальна.	
--	--	--

5.4. Инициация проекта

Группа процессов инициации состоит из процессов, которые выполняются для определения нового проекта или новой фазы существующего.

В рамках процессов инициации определяются изначальные цели и содержание и фиксируются изначальные финансовые ресурсы. Определяются внутренние и внешние заинтересованные стороны проекта, которые будут взаимодействовать и влиять на общий результат научного проекта. Данная информация закрепляется в уставе проекта.

Устав проекта документирует бизнес-потребности, текущее понимание потребностей заказчика проекта, а также новый продукт, услугу или результат, который планируется создать. Устав научного проекта магистерской работы должен иметь следующую структуру.

1. Цели и результат проекта

Под заинтересованными сторонами проекта понимаются лица или организации, которые активно участвуют в проекте или интересы которых могут быть затронуты как положительно, так и отрицательно в ходе исполнения или в результате завершения проекта. Информация по заинтересованным сторонам проекта представлена в таблице 8.

Таблица 8 - Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
ВНИИЭФ РФЯЦ	Проведение исследования геометрических ядер с целью разработки алгоритмов для дополнения инструментария ПК ЛОГОС.
Различные предприятия составляющие государственный сектор экономики, частные отечественные фирмы	Использования данной разработки различными фирмами для улучшения производительности при создании собственной

	продукции.
--	------------

Таблица 9 - Цели и результаты проекта

Цели проекта:	Дополнения инструментария ПК ЛОГОС путем создания алгоритмов построения поверхностей вытягиванием кривой вдоль заданного вектора и вращением кривых вокруг заданной оси.
Ожидаемые результаты проекта:	Получения инструментариев способных создавать поверхности разными способами.
Критерии приемки результата проекта:	Визуализация решения задачи по заданным параметрам.
Требования к результату проекта:	Созданный инструмент должен обладать удобным пользовательским интерфейсом, а полученные в результате построения плоскости должны иметь корректное построение.

2. Организационная структуры проекта

На данном этапе работы необходимо решить следующие вопросы: кто будет входить в рабочую группу данного проекта, определить роль каждого участника в данном проекте, а также прописать функции, выполняемые каждым из участников и их трудозатраты в проекте.

3. Ограничения и допущения проекта

Ограничения проекта – это все факторы, сдерживающие свободу участников команды в работе над данным проектом (таблица 10).

Таблица - 10. Ограничения проекта

Фактор	Ограничения/допущения
Источник финансирования	НИ ТПУ, ОЭФ/ ВНИИЭФ РФЯЦ
Бюджет затрат НТИ	147128,97
Сроки проекта	3 месяца
Дата утверждения плана управления	1.03.2019

проектом	
Дата завершения проекта	1.05.2019

5.5. Определение трудоемкости работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Данные по трудозатратам представлены в таблице 11.

Таблица 11 - Данные по трудозатратам

№ п/п	ФИО участника проекта	Роль в проекте	Функции	Трудозатраты, час
1	Фархутдинов В.Ф.,	Научный руководитель проекта	1. Обеспечение проекта ресурсами со стороны 2. Руководство и контроль 3. Регулярный анализ хода выполнения работ.	36
2	Кузнецов Г.О.	Магистрант	1. Выполнения расчетов по проекту 2. Подготовка отчета о проделанной работе	426
Итого:				462

Календарный план проекта за вычетом выходных и праздничных дней для сотрудников, работающих по шестидневной рабочей неделе, представлен в таблице 12.

Таблица 12 – Календарный план проекта

Код работы	Название	Длительность, дни	Дата начала работы	Дата окончания работы	Состав участников (ФИО исполнителей)
1	Составление и утверждение ТЗ	2	1.03.2019	02.03.2019	Фархутдинов В.Ф. Кузнецов Г.О.

2	Подбор и изучение материалов по теме	9	4.03.2019	14.03.2019	Кузнецов Г.О.
3	Сбор и анализ исходных данных	5	15.03.2019	20.03.2019	Кузнецов Г.О.
4	Выбор метода выполнения работы	7	21.03.2019	28.03.2019	Кузнецов Г.О.
5	Календарное планирование работ по теме	2	29.03.2019	30.03.2019	Фархутдинов В.Ф. Кузнецов Г.О.
6	Применение выбранного метода	17	01.04.2019	19.04.2019	Кузнецов Г.О.
7	Тестирование и анализ результатов работы	6	20.04.2019	26.04.2019	Кузнецов Г.О.
8	Исправление найденных ошибок, доработка модели	6	27.04.2019	07.05.2019	Фархутдинов В.Ф. Кузнецов Г.О.
9	Составление пояснительной записки к магистерской диссертации	12	8.05.2019	21.05.2019	Кузнецов Г.О.
10	Оформление пояснительной записки к магистерской диссертации по ГОСТу	7	22.05.2019	29.05.2019	Кузнецов Г.О.
Итого:		71			

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками,

характеризующимися датами начала и окончания выполнения данных работ (Таблица 13).

Таблица 13 – Календарный план-график выполнения работ

Календарный план-график выполнения работ по теме													
№ работы	Наименование работы	Исполнители	Дни	Продолжительность выполнения работ, дни									
				Март					Апрель		Май		
				2	9	5	7	2	17	6	6	12	7
1	Составление и утверждение ТЗ	М Р	2										
2	Подбор и изучение материалов по теме	М	9										
3	Анализ исходных данных	М	5										
4	Выбор метода выполнения работы	М Р	7										
5	Календарное планирование работ по теме	М	2										
6	Применение выбранного метода	М	17										
7	Тестирование и анализ результатов работы	М	6										
8	Исправление найденных ошибок,	М Р	6										

	доработка модели											
9	Составление пояснительной записки к магистерской диссертации	М	12									
10	Оформление пояснительной записки к магистерской диссертации по ГОСТу	М	7									

Руководитель



Магистрант



5.6. Бюджет научно-технического исследования

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

1. материальные затраты НТИ;
2. затраты на специальное оборудование для научных (экспериментальных) работ;
3. основная заработная плата исполнителей темы;
4. дополнительная заработная плата исполнителей темы;
5. отчисления во внебюджетные фонды (страховые отчисления);
6. затраты научные и производственные командировки;
7. контрагентные расходы;
8. накладные расходы.

В нашем случае расчет основных затрат производится только для научного руководителя, так как исполнитель является студентом на договорной основе, и, следовательно, со стороны ТПУ него нет затрат.

5.7. Расчёт материальных затрат НТИ

Данная статья отражает стоимость всех материалов, используемых при разработке проекта, включая расходы на их приобретение и доставку. Транспортные расходы принимаются в пределах 3-5% от стоимости материалов. В материальные затраты, помимо вышеуказанных, включаются дополнительно затраты на канцелярские принадлежности, диски, картриджи и т.п. Однако их учет ведется в данной статье только в том случае, если в научной организации их не включают в расходы на использование оборудования или накладные расходы.

Расчет затрат на материалы производится по форме, приведенной в таблице 14.

Таблица 14 – Материальные затраты

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты на материалы, руб.
Ноутбук MSI GV62 - 7RD	Шт	1	2605	2605 (амортизация за 3 мес)
Microsoft Windows 10 Professional RU x32/x64	Шт	1	10400	10400
Пакет Microsoft Office 2019 Home and Student RU x32/x64	Шт	1	4600	4600
Канцелярские принадлежности	Шт	1	500	500
Итого				18105

5.8. Основная заработная плата

Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20-30 % от тарифа или оклада.

Статья включает основную заработную плату работников, непосредственно занятых выполнением НИИ, (включая премии, доплаты) и дополнительную заработную плату:

$$Z_{\text{зн}} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (72)$$

где $Z_{\text{осн}}$ – основная заработная плата;

$Z_{\text{доп}}$ – дополнительная заработная плата.

Основная заработная плата ($Z_{\text{осн}}$) руководителя (лаборанта, инженера) от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_p, \quad (73)$$

где $Z_{\text{осн}}$ – основная заработная плата;

T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

$Z_{\text{дн}}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_m \cdot M}{F_d} \quad (74)$$

где Z_m – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года

при отпуске в 48 раб. дней $M=10,4$ месяца, 6-дневная неделя(магистрант);

при отпуске в 24 раб. дня $M=11,2$ месяца, 5-дневная неделя(руководитель);

F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. дней.

Месячный должностной оклад работника:

$$Z_m = Z_{\text{мс}} \cdot k_p \quad (75)$$

где Z_m – месячный должностной оклад работника, руб.;

k_p – районный коэффициент, равный 1,3 г. Томск.

Пример расчета заработной платы для руководителя:

$$Z_m = Z_{\text{мс}} \cdot k_p = 55427 \cdot 1,3 = 72055,1 \text{ руб.}$$

$$Z_{\text{он}} = \frac{72055,1 \cdot 11,2}{251} = 3215,21 \text{руб.}$$

$$Z_{\text{осн}} = Z_{\text{он}} \cdot T_p = 3215,21 \cdot 6 = 19291,25 \text{руб.}$$

Таблица 15 – Расчёт основной заработной платы

Исполнители	Z_{mc}	k_p	Z_m , руб	$Z_{\text{он}}$, руб	T_p , дни	$Z_{\text{осн}}$, руб
Руководитель	55427	1,3	72055,1	3215,21	6	19291,25
Магистрант	12663	1,3	16461,9	778,19	71	55252,12
Итого						74543,37

5.9. Дополнительная заработная плата

Затраты по дополнительной заработной плате исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{дон}} = k_{\text{дон}} \cdot Z_{\text{осн}}, \quad (76)$$

где $k_{\text{дон}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12).

Таблица 16 – Расчет дополнительной заработной платы

Исполнители	Основная ЗП, руб	Дополнительная ЗП, руб
Руководитель (ст.преподаватель)	19291,25	2314,95
Магистрант	55252,12	6630,25
Итого		8945,2

5.10. Отчисления во внебюджетные фонды

Отчисления во внебюджетные фонды являются обязательными по установленным законодательством Российской Федерации нормам органам

государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}), \quad (77)$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Установлен размер страховых взносов равный 27,1%. Отчисления во внебюджетные фонды представлены в таблице 16.

Таблица 17 – Отчисления во внебюджетные фонды

Исполнители	Основная ЗП, руб	Сумма отчислений, руб
Руководитель	19291,25	5227,93
Магистрант	55252,12	14973,32
Итого		20201,25

5.11. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$З_{накл} = (З_{осн} + З_{доп}) \cdot k_{нр}, \quad (78)$$

где $k_{нр}$ – коэффициент, учитывающий накладные расходы (принимается равным 30 %). $З_{накл} = 25046,57 \text{ руб.}$

Затраты на электроэнергию, потребляемую оборудованием (ноутбук), при среднем времени работы 6 час. в день и 71 днях работы, а также при стоимости 1 кВт = 4,5 руб. получим:

$$З_{эл} = \text{мощность} \cdot \text{часы} \cdot \text{дни} \cdot \text{тариф}$$

$$З_{эл} = 0,150 \cdot 6 \cdot 71 \cdot 4,5 = 287,55$$

5.12. Формирование бюджета затрат НТИ

Рассчитанная величина затрат научно-исследовательской работы (темы) является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции. Определение бюджета затрат на научно-исследовательский проект приведен в таблице 18.

Таблица 18 – Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб.
1. Материальные затраты НТИ	18105
2. Затраты по основной заработной плате исполнителей темы	74543,37
3. Затраты по дополнительной заработной плате исполнителей темы	8945,2
4. Отчисления во внебюджетные фонды	20201,25
5. Накладные расходы	25046,57
6. Затраты на электроэнергию	287,55
7. Бюджет затрат НТИ	147128,94

5.13. Реестр рисков проекта

Идентифицированные риски проекта включают в себя возможные неопределенные события, которые могут возникнуть в проекте и вызвать последствия, которые повлекут за собой нежелательные эффекты. Потенциальные риски представлены в таблице 19.

Таблица 19 – Реестр рисков

№	Риск	Потенциальное воздействие	Вероятность наступления (1-5)	Влияние риска (1-5)	Уровень риска	Способы смягчения риска	Условия наступления риска
1	Ошибки в программе	Получение некорректных результатов или же отсутствие таковых вообще	3	5	высокий	Проверка кода программы	Ошибки при разработке, дополнении или обновлении.

5.14. Оценки сравнительной эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

Интегральный финансовый показатель разработки определяется как:

$$I_{финр}^{исп.i} = \frac{\Phi_{pi}}{\Phi_{max}}, \quad (79)$$

где $I_{финр}^{исп.i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги). За максимально возможную стоимость исполнения примем 300000 руб.

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в размах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в размах (значение меньше единицы, но больше нуля).

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i, \quad (80)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

a_i^a, b_i^p – бальная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Расчет интегрального показателя ресурсоэффективности представлен в таблице 20.

Таблица 20 – Расчет интегрального показателя

Критерии	Весовой коэффициент параметра	Исп.1: магистрант	Исп.2: конкурент
Способствует росту производительности труда пользователя	0,2	5	5
Возможность применения любым предприятием	0,15	3	5
Многообразие решаемых задач	0,2	5	5
Простота применения	0,1	3	3
Конкурентоспособность (с другими системами)	0,15	4	5
Цена продукции	0,2	5	3
ИТОГО	1	4,35	4,2

$$I_{p-исп.1} = 5 \cdot 0,2 + 3 \cdot 0,15 + 5 \cdot 0,2 + 3 \cdot 0,1 + 4 \cdot 0,15 + 0,2 \cdot 5 = 4,35$$

$$I_{p-исп.2} = 5 \cdot 0,2 + 5 \cdot 0,15 + 4 \cdot 0,2 + 3 \cdot 0,1 + 5 \cdot 0,15 + 0,2 \cdot 3 = 4,2$$

$$I_{\max} = 4,35$$

Интегральный показатель эффективности вариантов исполнения разработки ($I_{исп.i}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.i} = \frac{I_{p-исп.i}}{I_{финр}}, \quad (81)$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта определяется по формуле:

$$\mathcal{E}_{cp} = \frac{I_{исп.i}}{I_{исп.\max}}, \quad (82)$$

Сравнительная эффективность разработки представлена в табл. 20.

Таблица 21 – Сравнительная эффективность разработки

№ п/п	Показатели	Проект магистранта	Проект конкурента
1	Интегральный финансовый показатель разработки	0,8	1
2	Интегральный показатель ресурсоэффективности разработки	4,35	4.2
3	Интегральный показатель эффективности	5,6	4,2
4	Сравнительная эффективность вариантов исполнения	1,28	0,97

Сравнение значений интегральных показателей эффективности позволяет понять и выбрать более эффективный вариант решения поставленной в Магистрантской работе технической задачи с позиции финансовой и ресурсной эффективности.

5.15. Выводы

В ходе выполнения части работы по финансовому менеджменту, ресурсоэффективности и ресурсосбережению был проведен анализ разрабатываемого исследования.

Во-первых, оценен коммерческий потенциал и перспективность проведения исследования. Результаты полученные в ходе исследования говорят о потенциале и перспективности на уровне выше среднего.

Во-вторых, проведено планирование НИР, а именно: определена структура и календарный план работы, трудоемкость и бюджет НТИ. Результаты соответствуют требованиям к ВКР по срокам и иным параметрам.

В-третьих, определена эффективность исследования в разрезах ресурсной, финансовой, бюджетной, социальной и экономической эффективности.

ЗАКЛЮЧЕНИЕ

Современные комплексы программ инженерного анализа и компьютерного моделирования начали свое развитие с двумерных систем проектирования и черчения. Но кардинальные изменения в технологии работы конструктора произошли с появлением компьютерных систем параметрического твердотельного моделирования. Они позволили фиксировать конструкторскую мысль не в виде плоского чертежа, а в виде трехмерной модели. С каждым годом технические системы совершенствуются, что влечет за собой и совершенствование систем автоматизированного проектирования и инженерного анализа. Над программным комплексом ЛОГОС также проводятся работы по его улучшению, например, расширяется функционал и список доступных инструментариев.

В результате проделанной работы была изучена среда коллективной разработки, правила работы с данными, освоены кроссплатформенные библиотеки, такие как: SDK, VTK, Qt, Cmake. Разработаны алгоритмы «Создание поверхности протягиванием ребра вдоль вектора направления на заданное расстояние» и «Создание поверхности вращением кривой вокруг заданной оси», которые были успешно внедрены в систему и дополнили список инструментариев.

Реализованные операции, в дальнейшем, позволят конечному пользователю создавать отдельные поверхности деталей конструкции при помощи протягивания и вращения заданных кривых, что несомненно облегчит достижения поставленных целей и задач.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Голованов Н.Н. Геометрическое моделирование. – М.: Физматлит, 2016 – 404 с.
- 2 Mario Botch., Leif Kobbelt. Polygon Mesh Processing// A K Peters, Natrick, Massachusetts. 2010.
- 3 Ли К. основы САПР(CAD/CAM/CAE). – СПб.: Питер, 2004. – 506с.
- 4 Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ. – М.: Мир, 2001. – 604с.
- 5 Ласло М. Вычислительная геометрия и компьютерная графика на C++. Пер. с англ. – М.: «Издательство БИНОМ», 1997. – 307с.
- 6 Hoffman C. Fundamental techniques for geometric and solid modeling //Purdue University, 1991. – 65p.
- 7 Котельников А.А. CAD/CAM/CAE – системы.: Курск: Юго-западный Государственный университет, 2014. – 336с.
- 8 Chang Kuang-Hua. Product Performance Evaluation using CAD/CAE. // Academic Press, Elsevier, 2013. – 538p.
- 9 Mastering Cmake. Kitware, Inc, 2008, 464p.
- 10 The VTK User's Guide. Updated for VTK Version 4.4. Kitware, Inc, 2004, 332p.
- 11 Schlee M. Qt 4.5. Профессиональное программирование на C++. – СПб.: БХВ-Петербург, 2010. – 896 с.
- 12 Латышев П.Н. Каталог САПР. Программы и производители. – М.: СОЛОН-ПРЕСС, 2010 – 608 с.
- 13 Российское 3D-ядро [Электронный ресурс]. - Режим доступа: <https://habr.com/ru/post/180455/> - Заглавие с экрана. – (Дата обращения 29.03.2019).
- 14 Open CASCADE Technology [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Open_CASCADE_Technology - Заглавие с экрана. – (Дата обращения 29.03.2019)
- 15 Наука, технологии и коммуникации в современном обществе/ Республиканская научнопрактическая конференция с международным участием «Наука, технологии и коммуникации в современном обществе», 6 февр. 2013 г. [Текст] : материалы / редкол.: А.Н. Макаров [и др.]; под ред. д-ра техн. наук, профессора Л.А. Симоновой. – Набережные Челны : Лаб. операт. полиграфии, 2013. – 466 с.
- 16 Параллельный пакет программ ЛЭГАК-ДК для расчета задач гидрогазодинамики и прочности на неструктурированных сетках в лагранжево-эйлеровых переменных/А.А. Рябов [и др.]/Вестник Нижегородского Университета им. Н.И. Лобачевского – 2011 г - № 4-5 – С. 2472-2473.
- 17 Особенности триангуляционных сеток на поверхностях в аналитическом и фасеточном представлении/А.Н. Лукичев [и др.]/ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородская обл. – 2016 г. – с 1-6.

- 18 Les Piegl, Wayne Tiller, The NURBS Book. – Springer, 1997 – 628 p.
- 19 Versprille K.J. Computer-aided design applications of the rational B-spline approximation from. – Diss. Syracuse University, 1975.
- 20 Болтянский В.Г., Ефремович В.А. Наглядная топология. – М.: Наука, 1982.(Библиотечка «Квант», Вып. 21)
- 21 Страуструп Б. Программирование: принципы и практика использования C++, испр. изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 1248 с.:
- 22 СанПиН 2.2.2/2.4.1.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».
- 23 СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений».
- 24 СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий».
- 25 СанПиН 2.2.4/2.1.8.10-32-2002 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки».
- 26 СанПиН 2.2.2.542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ».
- 27 СанПиН 2.2.4.1191-03 «Электромагнитные поля в производственных условиях».

Приложение А

(справочное)

Development of algorithms for creating geometric objects in the boundary representation for the preparation of initial data of engineering analysis in the software package LOGOS.

Студент

Группа	ФИО	Подпись	Дата
0BM71	Кузнецов Георгий Олегович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Трифонов А.Ю.	Доктор ф-м. наук		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель	Кабрышева О.П.			

Introduction

In the modern industry computer modeling and engineering analysis plays an increasingly important role. Their use allows to reduce the time and cost of development, creation and modernization of various technical devices, from simple designs to aircraft and nuclear powerhouse.

Experience in the development of new mechanisms and the development trend of structures and research methods show that the leading role in providing competitive advantages is played by computer modeling technology of individual system properties and virtual design of an object as a whole. This area is becoming increasingly important in the design process by finding the best design solutions and reducing development time and the cost of testing [15].

Currently, RFNC-VNIIEF is working on the creation and implementation of domestic LOGOS software for integrated simulation modeling, intended for use in enterprises of high-tech industries [16].

Creating a set of engineering analysis programs that allow for the study of structures with regard to various processes requires the solution of a priority task — this is the processing of design models for such modeling. The software package should be equipped with all the necessary tools for processing and editing design models.

An integral part of any software for engineering analysis is the geometric core. A geometric (mathematical) core is a set of functions whose execution provides the import, export, and construction of three-dimensional models. The kernel is created for use in application programs. Access to kernel functions for the end user is opened through a graphical user interface. In addition, the core is sometimes called the “engine” of a geometric modeling system. Just as the car engine determines the “ceiling” of its speed, the mathematical core defines the limit of the functionality of the CAE system using it, in terms of working with geometry.

The task was to develop algorithms for creating geometric objects in the boundary representation for the preparation of initial data of engineering analysis. To complete the task, we have:

- to study the types of representation of geometry in the LOGOS software package; mathematical models of geometric curves and surfaces;
- to study the organization of the data structures of the geometric core and their conformity with the PrePostProcessor data structures;
- to develop algorithms for “creating a surface by pulling along a guide vector at a given distance” and “creating a surface by rotating around given axis”.

General information

Computer-aided design system (CAD) is a system designed to automate the design process, which facilitates the process of creating, modifying and analyzing developed projects. (CAD) systems are usually used in conjunction with computer-aided engineering analysis (CAE) and computer-aided production (CAM), allowing you to solve the problem of choosing the optimal design appearance and technology of its production.

Computer-aided design system (CAD)

CAD (computer-aided design) is a computer-aided design system with various components that allows you to create a variety of engineering, architectural and construction projects.

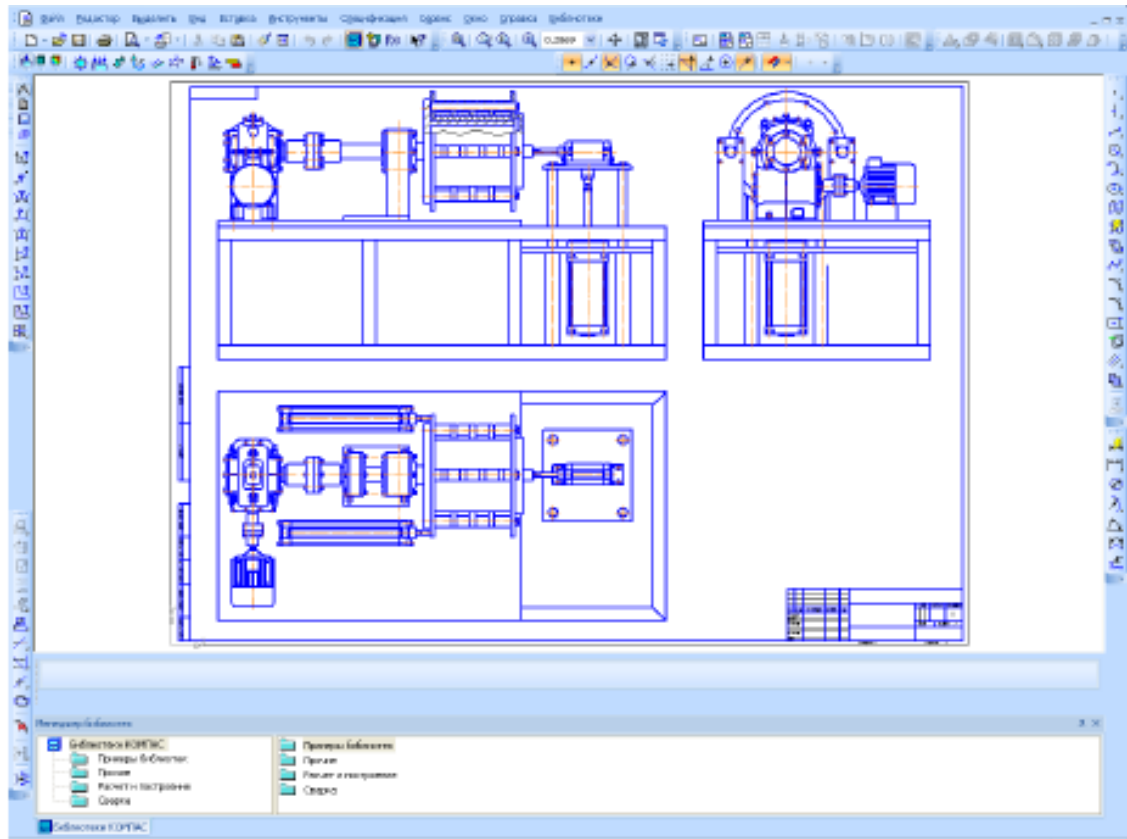


Image 1 - Sample drawing of the design under development in the KOMPAS-3D program.

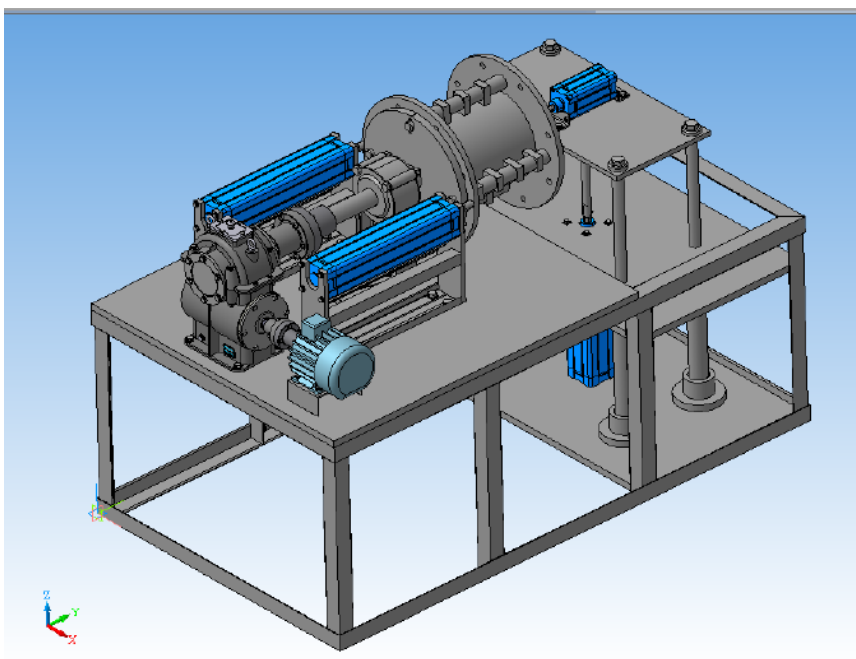


Image 2 - Example of a 3D model of a structure in the KOMPAS-3D program.

The main functions of CAD-systems

CAD systems provide the ability to work with two-dimensional (2D) and three-dimensional (3D) modeling, the ability to design from the simplest parts to complex structures.

Two-dimensional modeling allows to create a model in the form of a flat projection on a virtual sheet of paper. Examples of 2D modeling can be drawings and specifications.

Three-dimensional modeling is designed to create objects in space.

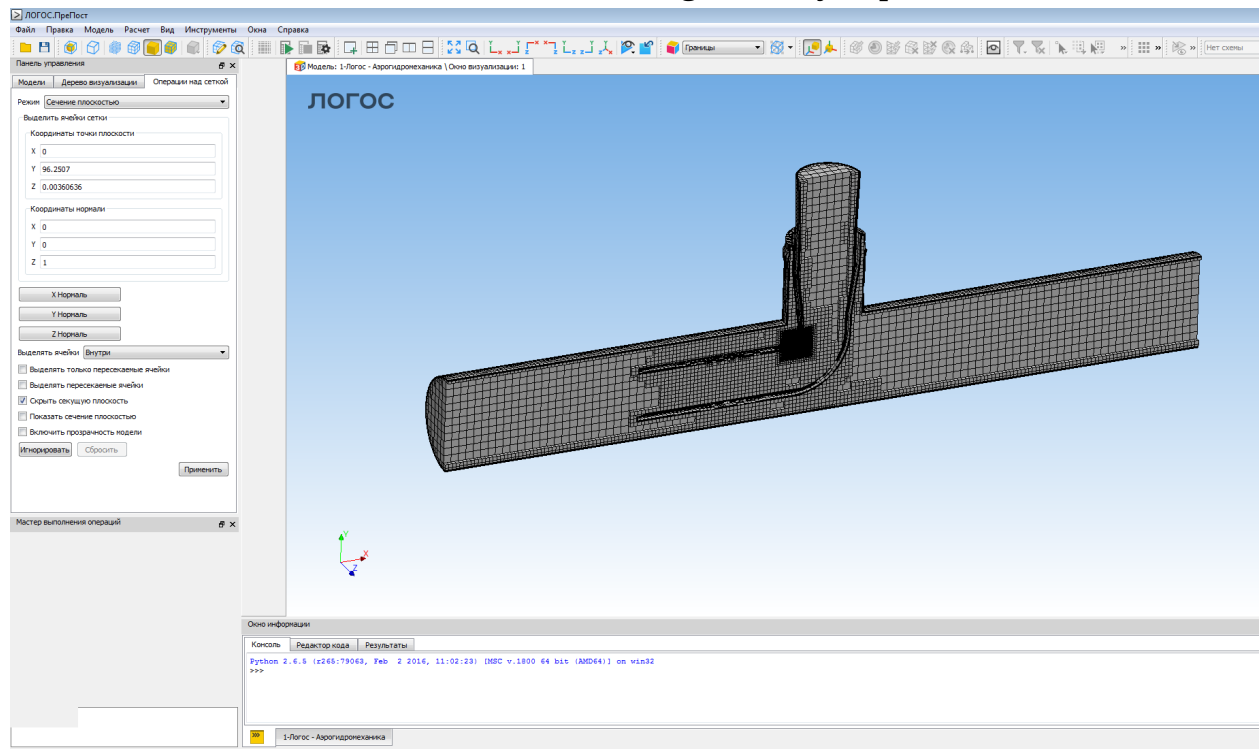
The examples of CAD systems are presented in table 1.

Table 1 - Examples of CAD systems

Foreign	Domestic
NX	T-FLEX
SolidWorks	KOMIAC – 3D
Autodesk	nanoCAD
CATIA	ADEM
FreeCAD	

CAE Automated Engineering Analysis Systems.

CAE (Computer Aided Engineering) is an automated engineering analysis system that allows to conduct theoretical substantiation of the products being developed under conditions close to real ones, without resorting to costly experiments.



Image

3 - Example of a calculation grid in the LOGOS software package

CAE systems can be focused on analysis by simulating various physical processes, such as heat and mass transfer, strength analysis, body kinematics, etc. The result of the analysis may be a change in the design of the products being developed, or for example, the material of manufacture.

Examples of CAE systems are presented in table 2.

Table 2 - Examples of CAE systems.

Foreign	Domestic
ANSYS	ПК ЛОГОС
Abaqus	Fidesys
HyperWorks (HyperMesh)	WIN Machine
NX NASTRAN	FlowVision

All computer-aided design systems have one thing in common - their work depends on the geometric (mathematical) core, which is the basis of the software packages presented in Tables 1 and 2. The main assignments of geometric core and their varieties will be considered in the next chapter.

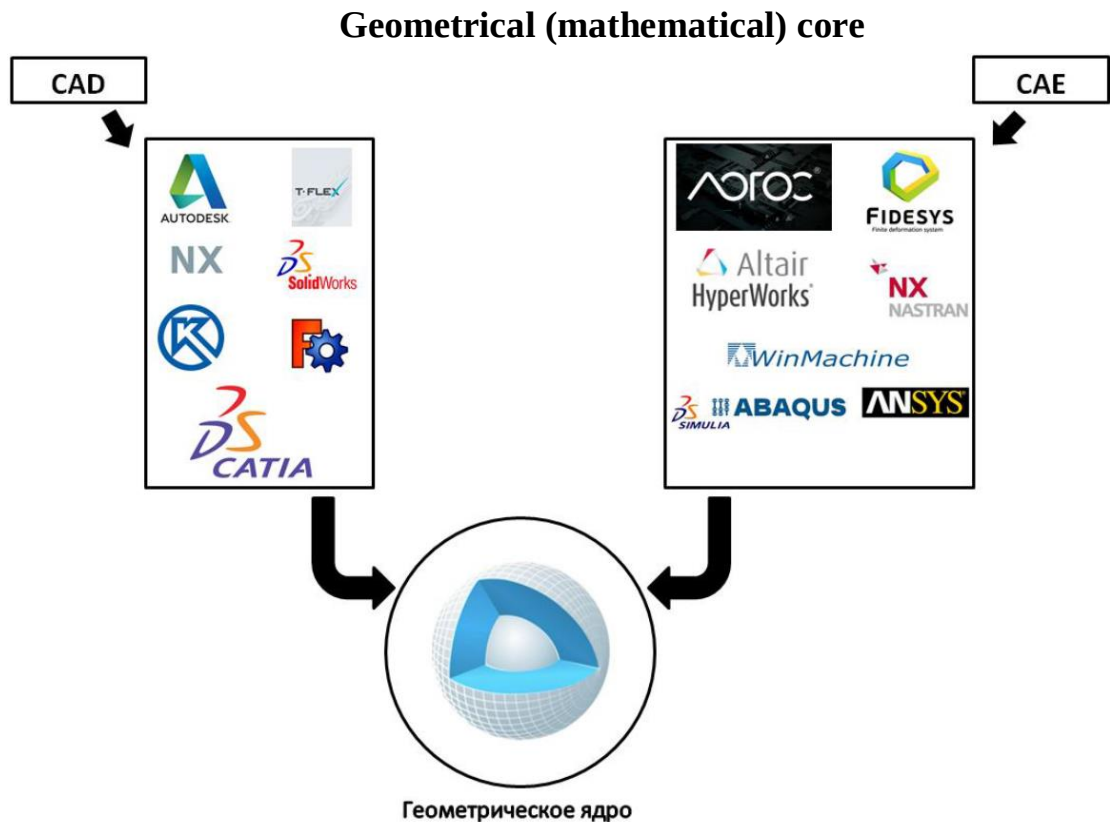


Image 4 - Common modules of CAD / CAE - systems.

The geometric core of three-dimensional modeling is a software component intended for use as a basic tool for developing software systems related to computer modeling of three-dimensional objects [13].

From the programmer's point of view, the geometric core is a library of functions / classes for creating geometric objects (point, curve, surface, solid body), editing them, visualizing the model and exchanging three-dimensional data with other programs.

One of the most difficult tasks in the development of engineering software is the creation of geometric kernels. The solving such a problem requires a combination of high competencies in mathematics, software design, etc.

The main types of geometric kernels for creating CAD / CAE systems

Kernels that exist nowadays are complex high-tech software products that have been developed for 15-30 (or more) years and are actively used in the development of application software. Most famous geometric kernels are:

- **Parasolid**

Parasolid (The developer: Siemens PLM Software) - this is the core of geometric modeling. This product is developed and maintained by Siemens PLM Software. Siemens PLM Software uses Parasolid in its own Solid Edge, Femap, NX systems, and uses the core for commercial purposes, providing licenses to independent software vendors and end users, such as Altair HyperWorks, Abaqus, etc.

- **ACIS**

3D ACIS Modeler (ACIS) is the core of geometric modeling developed by Spatial Corporation (formerly Spatial Technology). This core is a commercial product that is used in their developments by companies such as Autodesk, Bricsys, and others.

One of the important developments is 3D Spatial InterOp - a development toolkit software used to translate 3D data in various systems.

- **RGK**

RGK (Russian Geometric Kernel) - this product was developed in the framework of the project “Development of a domestic licensed 3D core” by order of the Ministry of Industry and Trade of the Russian Federation [13].

The advantage of creating such a system for the state is that this product will be non-commercial (all project costs are spent on developing the system, and not on the purchase of expensive licenses). At the moment, the core needs to be improved.

- **C3D Modeler (C3D Labs, Russia)**

In 1995, ASCON developed its own three-dimensional modeling core to create CAD KOMPAS-3D. This is done in order not to buy expensive licensed products of foreign manufacturers in the field of creating geometric core. The program KOMPAS-3D was released for commercial use in 2000. In 2012, C3D became a commercial product (previously used only as part of its own developments). Licenses of this core are used to create proprietary software products by companies such as: STC APM, Nanosoft, SolarTech, Eremiks, etc.

- **Open CASCADE Technology (OCCT) (OPEN CASCADE S.A.S, France)**

This core was created in 1993 by Matra Datavision and had the name CAS.CADE (Computer Aided Software for Computer Aided Design and Engineering). Initially, CAS.CADE was considered as a set of software tools for internal use, which allowed software developers to apply object-oriented technologies [14]. In 1999, the source code CAS.CADE was published on the Internet and received the name Open CASCADE. This was done in order to improve the code of this kernel could all independent developers. Software based on the Open CASCADE core: SALOME, FreeCAD

PC LOGOS based on 3DTV

In the LOGOS software package, the geometric core of 3DTV is used. In addition to the usual functionality for working with geometry, this mathematical core provides tools that complete diagnostics of design models and include all the necessary set of tools for quick and effective correction of geometric models. The process of diagnostics and correction is highly automated and can be applied as separate bodies, assemblies of various degrees of complexity. The correction is always carried out within the framework of the accuracy of the model, which ensures the original integrity of the representation and eliminates deformations [12].

The geometric core of 3DTV is focused on working with geometric data, both in the boundary representation (B-rep) and in faceted representation. Types of geometric representations will be discussed in the next chapter.

Types of geometric representations

The most general approach to the description of bodies is to represent the body as a set of shells limiting its volume, faces and edges are parametrically specified. Each shell is built from a set of surfaces of arbitrary shape that are joined to each other and contain complete information about their boundaries and connections with their neighbors. Such a description of the bodies is called the boundary representation. It makes it possible to perform many operations on the bodies, while maintaining a single way of their internal structure. Representation of bodies using borders allows you to model objects of arbitrary shape and complexity.

All the listed approaches in the description of bodies use topological objects and satisfy the conditions of connectedness, orientability and closedness.

B-rep oriented algorithms usually include the following steps:

- generation of three-dimensional vertices;
- creation of edges by the obtained coordinates of the vertices;
- creation of faces from edges;
- formation of a three-dimensional object from faces [1].

CSG

Constructive solid-state geometry (Constructive Solid Geometry or CSG) operates with primitives, which, as a rule, include a rectangular prism, triangular prism, sphere, cylinder, cone and torus. You can perform various operations on primitives and bodies derived from them (primarily Boolean operations). The surfaces used by the constructive solid-state geometry (spherical, cylindrical, conical, the surface of the torus and the plane) divide the space into two parts and for them it is possible to indicate from which side of the surface the internal volume of the body is located.

Faceted

Faceted geometry describes the surface of the body using a set of triangles so that the deviation of the resulting surface from the original is not more than a specified value. As a criterion for estimating deviations, various approaches are used:

- Absolute - the distance between the surfaces;
- Relative - deviation from the surface per unit length / area, volume loss, etc.

Advantages of faceted geometry:

- The simplest format of representation and storage - for a complete description of the surface, an array of triangles with the coordinates of the vertices is sufficient;
- Algorithms for processing this geometry: the projection of a point on the surface, finding the shortest path along the surface, the intersection of surfaces, etc.

Disadvantages:

- Loss of accuracy: irrevocable roughening of curved surfaces;
- In most cases - the loss or lack of information about the characteristic elements of the geometry: edges, fragments, and therefore there is a need to restore it.

Despite the shortcomings, the faceted representation of geometry is indispensable for solving many practical problems:

- Restoration of the surfaces of bodies according to the data obtained as a result of scanning a real object;
- Construction of a shell for a point cloud [2] [17].

Most CAD / CAE systems use boundary representation because Compared to the representation in the form of constructive solid-state geometry (CSG), which uses only primitive objects and Boolean operations to combine them, the boundary representation is more flexible and has a rich set of operations.

Conclusion

Modern complexes of engineering analysis and computer modeling programs began their development from two-dimensional design and drawing systems. But fundamental changes in the technology of the designer's work took place with the advent of computer systems of parametric solid modeling. They allowed to fix the design idea not in the form of a flat drawing, but in the form of a three-dimensional model. Every year the technical systems are improved, which entails the improvement of computer-aided design and engineering analysis. Above the LOGOS software complex, work is also underway to improve it, for example, the functionality and the list of available tools are expanded.

As a result of the work done, the collective development environment was studied, the rules for working with data, and cross-platform libraries were mastered, such as: SDK, VTK, Qt, Cmake. The algorithms "Creating a surface by pulling an edge along a direction vector at a given distance" and "Creating a surface by rotating a curve around a given axis" that have been developed, were successfully introduced into the system and added to the list of tools.

The implemented operations will further allow the end user to create separate surfaces of the construction details by pulling and rotating the specified curves, which will undoubtedly facilitate the achievement of the goals and objectives.

ПРИЛОЖЕНИЕ В. Код программы на C++ «Создание поверхности протягиванием»

```
using upp::widgets::VBoxLayout;
using upp::widgets::HBoxLayout;

class WidgetDeactivator
{
    QString m_activeWidget;
public:
    WidgetDeactivator( const QString& name ) : m_activeWidget( name ) {};
    void operator () ( uppPickingWidget* widget )
    {
        if( widget->objectName() != m_activeWidget )
            widget->Deactivate();
    }
};

static void MeshBrep( ik::IBrep* brep, vtkPolyData* pd );
static void MeshPolylines( const std::vector<std::vector<D3>>& polylines,
    vtkPolyData* poly );

uppCreateSurfaceExtrusionDialog::uppCreateSurfaceExtrusionDialog( QWidget *parent ) :
    m_scaleMin( 0.001 ),
    m_scaleMax( 5. ),
    m_isLoopPlanar( false ),
    m_distanceModifiedbyUser( false )
{
    setWindowTitle( tr( "Создание поверхности протягиванием" ) );

    memset( &m_cachedOpPrms, 0, sizeof( m_cachedOpPrms ) );

    lppApp->GetSession()->GetSelection()->Clear();
    m_edgesWidget = new uppPickingWidget( this );
    m_edgesWidget->setObjectName( "edges_widget" );
    targetMask objectMask;
    lppApp->GetTargetManager()->GetTargetMask( UPP_DOT_GEOMETRY_EDGE, objectMask );
    m_edgesWidget->SetTargetMask( objectMask );
    QColor selectionColor( 255, 255, 0, 255 );
    m_edgesWidget->SetSelectionColor( selectionColor );
    m_edgesWidget->setSizePolicy( QSizePolicy::Expanding, QSizePolicy::Fixed );
    m_edgesWidget->setMinimumWidth( 180 );
    m_edgesWidget->setFixedHeight( 48 );

    {////////ПЕБРА//////////////////////////////////////
        CollapsingBoxWidget* clpbxEdges = new CollapsingBoxWidget( this );
        clpbxEdges->setCaption( tr( "Ребро" ) );
        clpbxEdges->setUnrollingFlag( true );

        QSizePolicy sizePolicy( QSizePolicy::Expanding, QSizePolicy::Fixed );
        sizePolicy.setHorizontalStretch( 0 );
        sizePolicy.setVerticalStretch( 0 );
        sizePolicy.setHeightForWidth( m_edgesWidget->sizePolicy().hasHeightForWidth()
    );

    clpbxEdges->setSizePolicy( sizePolicy );

    VBoxLayout* vloutObjects = new VBoxLayout( this );
```

```

        vloutObjects->setObjectName( QString::fromUtf8( "vloutObjects" ) );
        vloutObjects->addWidget( m_edgesWidget );
        clpbxEdges->setInternalFrameLayout( vloutObjects );

        layout()->addWidget( clpbxEdges );
}//////////

{//////////НАПРАВЛЕНИЕ//////////

    CollapsingBoxWidget* clpbxDirection = new CollapsingBoxWidget( this );
    clpbxDirection->setCaption( tr( "Направление" ) );
    clpbxDirection->setUnrollingFlag( true );

    QVBoxLayout* vloutDirection = new QVBoxLayout( this );
    vloutDirection->setObjectName( QString::fromUtf8( "vloutVector" ) );

    m_directionWidget = new upp::widgets::DirectionWidget( this, false );
    m_directionWidget->setObjectName( "direction_widget" );

    QGridLayout* gloutDirection = new QGridLayout( this );
    gloutDirection->setObjectName( QString::fromUtf8( "gloutVector" ) );
    gloutDirection->addWidget( m_directionWidget );
    vloutDirection->addLayout( gloutDirection );
    clpbxDirection->setInternalFrameLayout( vloutDirection );

    layout()->addWidget( clpbxDirection );
}//////////

{//////////РАССТОЯНИЕ//////////

    CollapsingBoxWidget* clpbxL = new CollapsingBoxWidget( this );
    clpbxL->setCaption( tr( "Расстояние" ) );
    clpbxL->setContentsMargins( 0, 0, 0, 0 );
    clpbxL->setUnrollingFlag( true );

    uppQDoubleValidator* validator = new uppQDoubleValidator( this );

    QVBoxLayout* vloutL = new QVBoxLayout( this );
    vloutL->setContentsMargins( 0, 0, 0, 0 );
    m_leDistance = new QLineEdit( this );
    m_leDistance->setValidator( validator );
    m_leDistance->setText( "0" );
    QGridLayout* gloutL = new QGridLayout( this );
    gloutL->addWidget( m_leDistance );
    vloutL->addLayout( gloutL );
    clpbxL->setInternalFrameLayout( vloutL );

    layout()->addWidget( clpbxL );
}//////////

{//////////МАСШТАБ//////////

    CollapsingBoxWidget* clpbxL = new CollapsingBoxWidget( this );
    clpbxL->setCaption( tr( "Масштабный коэффициент" ) );
    clpbxL->setContentsMargins( 0, 0, 0, 0 );
    clpbxL->setUnrollingFlag( true );

    uppQDoubleValidator* validator = new uppQDoubleValidator( this );
    validator->SetLimits( m_scaleMin, m_scaleMax );

```

```

VBoxLayout* vloutL = new VBoxLayout( this );
vloutL->setContentsMargins( 0, 0, 0, 0 );
m_leScale = new QLineEdit( this );
m_leScale->setValidator( validator );
m_leScale->setText( "0" );
GridLayout* gloutL = new GridLayout( this );
gloutL->addWidget( m_leScale );
vloutL->addLayout( gloutL );
clpbxL->setInternalFrameLayout( vloutL );

layout()->addWidget( clpbxL );
}////////////////////////////////////

{////ПРОСМОТР////////////////////////////////////
CollapsingBoxWidget* clpbxL = new CollapsingBoxWidget( this );
clpbxL->setCaption( tr( "Отображение" ) );
clpbxL->setContentsMargins( 0, 0, 0, 0 );
clpbxL->setUnrollingFlag( true );

HBoxLayout* hlayout = new HBoxLayout( this );
hlayout->setContentsMargins( 0, 0, 0, 0 );
m_previewSwitcher = new QCheckBox( this );
m_previewSwitcher->setChecked( true );
hlayout->addWidget( m_previewSwitcher );
hlayout->addWidget( new QLabel( tr( "Просмотр результата" ) ) );
hlayout->addSpacerItem( new QSpacerItem( 40, 20, QSizePolicy::Expanding,
QSizePolicy::Minimum ) );
clpbxL->setInternalFrameLayout( hlayout );

layout()->addWidget( clpbxL );
}////////////////////////////////////

m_widgets.push_back( m_edgesWidget );
m_widgets.push_back( m_directionWidget );

QSpacerItem* verticalSpacer = new QSpacerItem( 20, 40, QSizePolicy::Minimum,
QSizePolicy::Expanding );
layout()->addItem( verticalSpacer );

connect( m_edgesWidget, SIGNAL( SelectionModified() ), SLOT( OnEdgePicked() ) );
connect( m_edgesWidget, SIGNAL( ListWidgetActivated( const QString& ) ), SLOT(
OnWidgetActivated( const QString& ) ) );
connect( m_directionWidget, SIGNAL( DirectionChanged() ), SLOT(
OnDirectionChanged() ) );
connect( m_directionWidget, SIGNAL( DirectionRemoved() ), SLOT(
OnDirectionChanged() ) );
connect( m_directionWidget, SIGNAL( ListWidgetActivated( const QString& ) ),
SLOT( OnWidgetActivated( const QString& ) ) );
connect( m_leDistance, SIGNAL( editingFinished() ), SLOT(
OnDistanceEditingFinished() ) );
connect( m_leScale, SIGNAL( editingFinished() ), SLOT( OnScaleEditingFinished() )
);
connect( m_previewSwitcher, SIGNAL( stateChanged( int ) ), SLOT(
OnPreviewSwitcherStateChanged() ) );

{
    vtkSmartPointer<vtkPolyData> polydata = vtkSmartPointer<vtkPolyData>::New();

```

```

        vtkSmartPointer<vtkPolyDataMapper> mapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
        mapper->SetInput( polydata );

        m_transformedEdgesActor = vtkSmartPointer<vtkActor>::New();
        m_transformedEdgesActor->SetMapper( mapper );
        m_transformedEdgesActor->GetProperty()->SetColor( 0, .6, 0 );
        m_transformedEdgesActor->GetProperty()->SetLineWidth( 2. );
        m_transformedEdgesActor->GetProperty()->LightingOff();
    }

    {
        vtkSmartPointer<vtkPolyData> polydata = vtkSmartPointer<vtkPolyData>::New();

        vtkSmartPointer<vtkPolyDataMapper> mapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
        mapper->SetInput( polydata );

        m_brepPreviewActor = vtkSmartPointer<vtkActor>::New();
        m_brepPreviewActor->SetMapper( mapper );
        m_brepPreviewActor->GetProperty()->SetColor( .8, .8, .8 );
        m_brepPreviewActor->GetProperty()->SetOpacity( .2 );
        m_brepPreviewActor->GetProperty()->SetAmbientColor( 0.23125, 0.23125, 0.23125
);
        m_brepPreviewActor->GetProperty()->SetAmbient( 0.1 );
        m_brepPreviewActor->GetProperty()->SetSpecularColor( 0.773911, 0.773911,
0.773911 );
        m_brepPreviewActor->GetProperty()->SetSpecular( 1. );
        m_brepPreviewActor->GetProperty()->SetSpecularPower( 128 );
    }

    memset( &m_cachedOpPrms, 0, sizeof( m_cachedOpPrms ) );

    m_leScale->setText( QString( "%1" ).arg( 1 ) );
    m_leDistance->setText( QString( "%1" ).arg( 1 ) );
}

uppCreateSurfaceExtrusionDialog::~uppCreateSurfaceExtrusionDialog()
{
}

bool uppCreateSurfaceExtrusionDialog::LoadData()
{
    return true;
}

bool uppCreateSurfaceExtrusionDialog::SaveData()
{
    return true;
}

void uppCreateSurfaceExtrusionDialog::BeforeActivate()
{
    m_edgesWidget->Activate();
    m_directionWidget->Deactivate();

    uppIVisualizer* visualizer = lppApp->GetSession()->GetCurrentVisualizer();
    if (visualizer )
    {
        uppISelection* selection = lppApp->GetSession()->GetSelection();

```

```

        if ( selection )
        {
            selection->Clear();
        }
        visualizer->Update();
    }

    visualizer->GetRenderer()->AddActor( m_transformedEdgesActor );
    visualizer->GetRenderer()->AddActor( m_brepPreviewActor );

    m_distanceModifiedbyUser = false;
}

void uppCreateSurfaceExtrusionDialog::BeforeDeactivate()
{
    if ( uppIVisualizer* visualizer = lppApp->GetSession()->GetCurrentVisualizer() )
    {
        visualizer->GetRenderer()->RemoveActor( m_transformedEdgesActor );
        visualizer->GetRenderer()->RemoveActor( m_brepPreviewActor );
    }
}

bool uppCreateSurfaceExtrusionDialog::Execute()
{
    uppIPartGeometry* geometry = lppApp->GetSession()->GetCurrentPart()-
>GetIGeometry();
    uppLogger& logger = uppLogger::getLogger( "uppCreateSurfaceBySectionsDialog" );

    if ( !m_directionWidget->IsDone() )
    {
        logger.error() << "Необходимо задать направление" << logger.end();
        return false;
    }

    double dist = m_leDistance->text().toDouble();
    if ( fabs(dist) < geometry->GetTolerance()*100. )
    {
        logger.error() << "Расстояние меньше " << geometry->GetTolerance()*100. <<
logger.end();
        return false;
    }

    double scale = m_leScale->text().toDouble();
    if ( scale < m_scaleMin || scale > m_scaleMax )
    {
        logger.error() << "Масштабный коэффициент должен быть в диапазоне [" <<
m_scaleMin << "," << m_scaleMax << "]" << logger.end();
        return false;
    }

    D3 dir = m_directionWidget->GetDirection();
    dir.normalize();
    dir *= dist;
    CMatrix4d trans;
    trans.makeIdentity();
    trans.scale( scale, scale, scale );
    D3 mc = upp::ops::MassCenter( m_closedPlaneLoop );
    D3 newPoint = trans * mc;
    trans.translate( mc - newPoint );
    trans.translate( dir );

```

```

        std::string errorStr;
        std::shared_ptr<undo::tv3::GeoState> geoState = undo::tv3::GeoState::Generate(
geometry );
        bool result = upp::ops::SurfaceCreateByExtrusion( geometry, m_closedPlaneLoop,
trans, "", errorStr );
        if ( result )
        {
            undo::tv3::SaveGeometryUndoData( geoState, windowTitle() );
            m_edgesWidget->Clear();
            lppApp->GetSession()->GetCurrentViewport()->GetIVisualizer()->Update();
        }
        else
        {
            logger.error() << errorStr.c_str() << logger.end();
        }

        return true;
    }

void uppCreateSurfaceExtrusionDialog::OnEdgePicked()
{
    m_closedPlaneLoop.clear();
    m_isLoopPlanar = false;

    uppIPartGeometry* geometry = lppApp->GetSession()->GetCurrentPart()-
>GetIGeometry();
    uppISelection *selection = m_edgesWidget->GetSelection();

    std::vector<uppIGeometryEdge*> selectedEdges;
    if ( selection )
    {
        uppDataTreeNode* object = NULL;
        uppIGeometryEdge* edge = NULL;
        selection->InitTraversal();
        while ( object = selection->GetNextElement() )
        {
            if ( edge = dynamic_cast<uppIGeometryEdge*>( object->GetINode() ) )
            {
                selectedEdges.push_back( edge );
            }
        }
    }

    if ( !selectedEdges.empty() )
    {
        std::vector<std::vector<uppIGeometryEdge*>> contours;
        std::vector<bool> closedness;
        upp::ops::GetEdgeChains( selectedEdges, geometry->GetScene()->tolerance(),
contours, closedness );

        if ( contours.size() == 1 )
        {
            {
                m_closedPlaneLoop = contours.front();
                m_isLoopPlanar = upp::ops::IsPlanar( m_closedPlaneLoop, geometry-
>GetScene()->tolerance() );
            }
        }
    }
}

```

```

    }

    if ( !m_distanceModifiedbyUser && !m_closedPlaneLoop.empty() && m_isLoopPlanar )
    {
        CBox3d bbox;
        for ( size_t i = 0; i < m_closedPlaneLoop.size(); i++ )
        {
            CBox3d _box;
            m_closedPlaneLoop[i]->BBox( _box );
            bbox.extendBy( _box );
        }
        m_leDistance->setText( QString( "%1" ).arg( bbox.diagonal() ) );
    }

    UpdatePreview();
    lppApp->GetSession()->GetCurrentVisualizer()->Update();
}

void uppCreateSurfaceExtrusionDialog::UpdatePreview()
{
    vtkPolyData* poly = vtkPolyData::SafeDownCast( m_transformedEdgesActor-
>GetMapper()->GetInputDataObject(0,0) );
    if ( !poly )
        return;

    poly->Initialize();

    vtkPolyData* mesh = vtkPolyData::SafeDownCast( m_brepPreviewActor->GetMapper()-
>GetInputDataObject( 0, 0 ) );
    if ( !mesh )
        return;

    mesh->Initialize();

    if ( !m_previewSwitcher->isChecked() )
        return;

    if ( !m_directionWidget->IsDone() )
        return;

    if ( !m_isLoopPlanar )
        return;

    uppIPartGeometry* geometry = lppApp->GetSession()->GetCurrentPart()-
>GetIGeometry();
    if ( !geometry )
        return;

    GetCurrentOperationParams( m_cachedOpPrms );

    D3 dir = m_directionWidget->GetDirection();
    double dist = m_leDistance->text().toDouble();
    double scale = m_leScale->text().toDouble();

    dir.normalize();
    dir *= dist;
    CMatrix4d trans;
    trans.makeIdentity();

```

```

trans.scale( scale, scale, scale );

D3 mc = upp::ops::MassCenter( m_closedPlaneLoop );
D3 newPoint = trans * mc;
trans.translate( mc - newPoint );
trans.translate( dir );

std::vector<std::vector<D3>> polylines;
upp::ops::TransformedTessellation( m_closedPlaneLoop, geometry->GetScene()-
>tolerance(), trans, polylines );

if ( !polylines.empty() )
{
    MeshPolylines( polylines, poly );
}

if ( ik::IBrep* brep = upp::ops::SurfaceCreateByExtrusion( m_closedPlaneLoop,
trans, false, "Ice preview" ) )
{
    MeshBrep( brep, mesh );
    geometry->GetScene()->deleteEntity( brep );
}

uppIVisualizer* visualizer = lppApp->GetSession()->GetCurrentVisualizer();
visualizer->GetRenderer()->ResetCameraClippingRange();
}

void uppCreateSurfaceExtrusionDialog::OnDistanceEditingFinished()
{
    m_distanceModifiedbyUser = true;

    OpParameters prms;
    GetCurrentOperationParams( prms );
    if ( prms == m_cachedOpPrms )
        return;

    UpdatePreview();
    lppApp->GetSession()->GetCurrentVisualizer()->Update();
}

void uppCreateSurfaceExtrusionDialog::OnScaleEditingFinished()
{
    double scale = m_leScale->text().toDouble();
    if ( scale < m_scaleMin ) scale = m_scaleMin;
    if ( scale > m_scaleMax ) scale = m_scaleMax;
    m_leScale->blockSignals( true );
    m_leScale->setText( QString( "%1" ).arg( scale ) );
    m_leScale->blockSignals( false );

    OpParameters prms;
    GetCurrentOperationParams( prms );
    if ( prms == m_cachedOpPrms )
        return;

    UpdatePreview();
    lppApp->GetSession()->GetCurrentVisualizer()->Update();
}

void uppCreateSurfaceExtrusionDialog::OnDirectionChanged()

```

```

{
    OpParameters prms;
    GetCurrentOperationParams( prms );
    if ( prms == m_cachedOpPrms )
        return;

    UpdatePreview();
    lppApp->GetSession()->GetCurrentVisualizer()->Update();
}

void uppCreateSurfaceExtrusionDialog::OnPreviewSwitcherStateChanged()
{
    vtkPolyData* pd = vtkPolyData::SafeDownCast( m_brepPreviewActor->GetMapper()-
>GetInputDataObject( 0, 0 ) );
    if ( pd && pd->GetNumberOfCells() == 0 )
    {
        UpdatePreview();
    }
    m_brepPreviewActor->SetVisibility( m_previewSwitcher->isChecked() );
    m_transformedEdgesActor->SetVisibility( m_previewSwitcher->isChecked() );
    lppApp->GetSession()->GetCurrentVisualizer()->Update();
}

void uppCreateSurfaceExtrusionDialog::OnWidgetActivated( const QString& name )
{
    if ( !m_widgets.empty() )
        std::for_each( m_widgets.begin(), m_widgets.end(), WidgetDeactivator( name )
);
}

void uppCreateSurfaceExtrusionDialog::GetCurrentOperationParams( OpParameters& prms )
{
    D3 dir = m_directionWidget->GetDirection();
    double dist = m_leDistance->text().toDouble();
    double scale = m_leScale->text().toDouble();
    prms.directionX = dir.x;
    prms.directionY = dir.y;
    prms.directionZ = dir.z;
    prms.distance = dist;
    prms.scaleCoeff = scale;
}

void MeshPolylines( const std::vector<std::vector<D3>>& polylines, vtkPolyData* poly
)
{
    size_t ptsCount = 0, cellsCount = polylines.size();
    for ( int i = 0; i < polylines.size(); i++ )
    {
        ptsCount += polylines[i].size();
    }

    if ( !poly->GetPoints() )
    {
        vtkSmartPointer<vtkPoints> pts = vtkSmartPointer<vtkPoints>::New();
        poly->SetPoints( pts );
    }

    poly->GetPoints()->SetNumberOfPoints( ptsCount );

    poly->Allocate( cellsCount );
}

```

```

size_t offset = 0;
for ( int i = 0; i < polylines.size(); i++ )
{
    vtkSmartPointer<vtkIdList> ids = vtkSmartPointer<vtkIdList>::New();
    ids->SetNumberOfIds( polylines[i].size() );
    for ( int j = 0; j < polylines[i].size(); j++ )
    {
        D3 pt = polylines[i][j];
        poly->GetPoints()->SetPoint( offset + j, pt.x, pt.y, pt.z );
        ids->SetId( j, offset + j );
    }
    offset += polylines[i].size();
    poly->InsertNextCell( VTK_POLY_LINE, ids );
}

poly->Squeeze();
}

void MeshBrep( ik::IBrep* brep, vtkPolyData* pd )
{
    int totalPoints = 0;
    int totalCells = 0;
    for ( int i = 0; i < brep->nFace(); i++ )
    {
        totalPoints += brep->face( i )->meshVx().size();
        totalCells += brep->face( i )->meshTr().size();
    }

    pd->Initialize();

    if ( !pd->GetPoints() )
    {
        vtkSmartPointer<vtkPoints> pts = vtkSmartPointer<vtkPoints>::New();
        pd->SetPoints( pts );
    }

    pd->GetPoints()->SetDataTypeToFloat();
    pd->GetPoints()->SetNumberOfPoints( totalPoints );

    vtkSmartPointer<vtkFloatArray> normals = vtkSmartPointer<vtkFloatArray>::New();
    normals->SetNumberOfComponents( 3 );
    normals->SetNumberOfTuples( totalPoints );
    pd->GetPointData()->SetNormals( normals );

    pd->Allocate( totalCells );

    int offset = 0;
    for ( int i = 0; i < brep->nFace(); i++ )
    {
        ik::IFace* face = brep->face( i );

        ik::IShell* shell = face->shell();
        bool reverted = ( shell == NULL ) ? face->isReverted() : mk::xor2( face-
>isReverted(), shell->isReverted() );
        float sign = reverted ? -1 : 1;

        for ( int j = 0; j < face->meshVx().size(); j++ )

```

```

{
    D3 pt = face->meshVx()[j];
    pd->GetPoints()->SetPoint( offset + j, pt.ptr() );
    D3 nrm = face->meshNormal()[j] * sign;
    pd->GetPointData()->GetNormals()->SetTuple( offset + j, nrm.ptr() );
}

for ( int j = 0; j < face->meshTr().size(); j++ )
{
    I3 ids = face->meshTr()[j];
    vtkIdType offsetedsIds[3] = { ids.x + offset, ids.y + offset, ids.z +
offset };

    pd->InsertNextCell( VTK_TRIANGLE, 3, offsetedsIds );
}

offset += face->meshVx().size();
}

pd->Squeeze();
}

```

ПРИЛОЖЕНИЕ С. Код программы на С++. Класс, отвечающий за создание контура

```
static ik::IEdge* CreateEdge(D3 p1, D3 p2, ik::IBrep* brep)
{
}
namespace upp
{
    namespace ops
    {
        UPPGEOMOPS_EXPORT ik::IBrep* SolidCreateByExtrusion(
std::vector<uppIGeometryEdge*>& chain, const CMatrix4d& trans, bool formSolid, const
char* label )
        {
            if ( chain.empty() )
                return NULL;
            ik::IScene* scene = chain.front()->GetIEdge()->scene();
            TVector<ik::IEdge*> srcEdges( chain.size() );
            for ( size_t i = 0; i < chain.size(); i++ )
            {
                srcEdges[i] = chain[i]->GetIEdge();
            }
            ik::IBrep* edgesBrep( NULL );
            scene->classFactory()->createBrep( edgesBrep, scene->rootBlock() );
            bool opResult = true;

            TVector<ik::IEdge*> transformedEdges;
            opResult = uppGeometryUtils::CreateEdgesTransformed( scene, srcEdges,
trans, transformedEdges, edgesBrep );
            if ( !opResult )
            {
                // удаляем временный контейнер для ребер
                scene->deleteEntity( edgesBrep );
                return NULL;
            }
            assert( transformedEdges.size() == srcEdges.size() );

            D3 sp1 = srcEdges.front()->startVertex()->point();
            D3 sp2 = transformedEdges.front()->startVertex()->point();
            D3 ep1 = srcEdges.back()->endVertex()->point();
            D3 ep2 = transformedEdges.back()->endVertex()->point();

            ik::IEdge* e0 = CreateEdge( sp1, sp2, edgesBrep );
            ik::IEdge* e1 = CreateEdge( ep1, ep2, edgesBrep );
            TVector<ik::IEdge*> faceEdges;
            faceEdges.add() = e0;
            faceEdges.add() = e1;
            for ( size_t i = 0; i < srcEdges.size(); i++ )
                faceEdges.add() = srcEdges[ i ];
            for ( size_t i = 0; i < transformedEdges.size(); i++ )
                faceEdges.add() = transformedEdges[ i ];

            {
                ik::I13Seg* segment( NULL );
                ik::IEdge* edge( NULL );
                ik::IScene* scene = edgesBrep->scene();
                ik::IClassFactory* classFactory = scene->classFactory();
                classFactory->createSegment3D(segment, sp1.ptr(), sp2.ptr());
            }
        }
    }
}
```

```

        classFactory->createEdge(edge, edgesBrep, segment);
        classFactory->createSegment3D(segment, ep1.ptr(), ep2.ptr());
        classFactory->createEdge(edge, edgesBrep, segment);

    }

    ik::IBlockLoc brepLoc;
    ik::IBrep* brep = NULL;
    int i;

    TVectorC<ik::IBlockLoc> edgesLoc(faceEdges.size());
    TVector<ik::IEdge*> edges(faceEdges.size());

    for (i = 0; i < faceEdges.size(); i++)
    {
        TAc<ik::IPart*> aPart;
        ik::IBlock* block = ik::IBlock::cast(faceEdges[i]->parent()->parent());
        edgesLoc[i].reset(block, aPart);
    }
    std::auto_ptr<uppCreateGeometryFaceOperation> operation =
    std::auto_ptr<uppCreateGeometryFaceOperation>(uppCreateGeometryFaceOperation::New());
    operation->SetEdges(faceEdges.ac(), &(edgesLoc.ac()));
    operation->SetBrep(edgesBrep, &brepLoc);
    operation->SetCreateMode(uppCreateGeometryFaceOperation::UPP_REGULAR);
    operation->SetApproximationTolerance(scene->tolerance());

    if (operation->Execute() == Operation::OPER_OK);
    return edgesBrep;
}

UPPGEOMOPS_EXPORT bool SolidCreateByExtrusion( uppIPartGeometry* pg,
std::vector<uppIGeometryEdge*>& chain, const CMatrix4d& trans, const char* label,
std::string& errorStr )
{
    bool opResult = true;

    ik::IBrep* solid = SolidCreateByExtrusion( chain, trans, true, label );

    if( surface )
        pg->Update();

    return (surface != NULL);
}
}

```

ПРИЛОЖЕНИЕ D. Код программы на С++ «Создание поверхности вращением кривой вокруг заданной оси»

```
using upp::widgets::VBoxLayout;
using upp::widgets::HBoxLayout;

uppCreateSolidRotationByCurveDialog::uppCreateSolidRotationByCurveDialog( QWidget
*parent ) :
{
    setWindowTitle( tr( "Создание поверхности вращением кривой вокруг заданной оси"
) );

    memset( &m_cachedOpPrms, 0, sizeof( m_cachedOpPrms ) );

    lppApp->GetSession()->GetSelection()->Clear();
    m_edgesWidget = new uppPickingWidget( this );
    m_edgesWidget2 = new uppPickingWidget( this );
    m_edgesWidget->setObjectName( "edges_widget" );
    m_edgesWidget2->setObjectName( "asix_widgest" );
    targetMask objectMask;
    lppApp->GetTargetManager()->GetTargetMask( UPP_DOT_GEOMETRY_EDGE, objectMask );
    m_edgesWidget->SetTargetMask( objectMask );
    QColor selectionColor( 255, 255, 0, 255 );
    m_edgesWidget->SetSelectionColor( selectionColor );
    m_edgesWidget->setSizePolicy( QSizePolicy::Expanding, QSizePolicy::Fixed );
    m_edgesWidget->setMinimumWidth( 180 );
    m_edgesWidget->setFixedHeight( 48 );
    m_edgesWidget->setFocus();

    m_edgesWidget2->SetTargetMask( objectMask );
    m_edgesWidget2->SetSelectionColor( selectionColor );
    m_edgesWidget2->setSizePolicy( QSizePolicy::Expanding, QSizePolicy::Fixed );
    m_edgesWidget2->setMinimumWidth( 180 );
    m_edgesWidget2->setFixedHeight( 48 );

    {////////ПЕБРА//////////////////////////////////////
        CollapsingBoxWidget* clpbxEdges = new CollapsingBoxWidget( this );
        clpbxEdges->setCaption( tr( "Контып" ) );
        clpbxEdges->setUnrollingFlag( true );

        QSizePolicy sizePolicy( QSizePolicy::Expanding, QSizePolicy::Fixed );
        sizePolicy.setHorizontalStretch( 0 );
        sizePolicy.setVerticalStretch( 0 );
        sizePolicy.setHeightForWidth(m_edgesWidget>sizePolicy().hasHeightForWidth());
        clpbxEdges->setSizePolicy( sizePolicy );
        VBoxLayout* vloutObjects = new VBoxLayout( this );
        vloutObjects->setObjectName( QString::fromUtf8( "vloutObjects" ) );
        vloutObjects->addWidget( m_edgesWidget );
        clpbxEdges->setInternalFrameLayout( vloutObjects );

        layout()->addWidget( clpbxEdges );
    }//////////

    {////////ОСЬ//////////////////////////////////////

        CollapsingBoxWidget* clpbxDirection = new CollapsingBoxWidget( this );
        clpbxDirection->setCaption( tr( "Ось" ) );
        clpbxDirection->setUnrollingFlag( true );
```

```

VBoxLayout* vloutDirection = new VBoxLayout( this );
vloutDirection->setObjectName( QString::fromUtf8( "vloutVector" ) );

m_directionWidget = new upp::widgets::DirectionWidget( this, false );
m_directionWidget->setObjectName( "direction_widget" );

QGridLayout* gloutDirection = new QGridLayout( this );
gloutDirection->setObjectName( QString::fromUtf8( "gloutVector" ) );
gloutDirection->addWidget( m_directionWidget );
vloutDirection->addLayout( gloutDirection );
clpbxDirection->setInternalFrameLayout( vloutDirection );

layout()->addWidget( clpbxDirection );
}////////////////////////////////////

uppCreateSolidRotationByCurveDialog::~uppCreateSolidRotationByCurveDialog()
{

}

bool uppCreateSolidRotationByCurveDialog::Execute()
{
    uppIPartGeometry* geometry = lppApp->GetSession()->GetCurrentPart()-
>GetIGeometry();
    uppLogger& logger = uppLogger::getLogger( "uppCreateSolidRotationByCurveDialog"
);
    QVector<uppIGeometryEdge*> edges;

    if ( GetEdges( edges ) == 0 )
    {
        logger.error() << "Не задан контур для вращения" << logger.end();
        return false;
    }

    if ( !m_directionWidget->IsDone() )
    {
        logger.error() << "Необходимо задать направление" << logger.end();
        return false;
    }

    ik::IScene *scene = geometry->GetScene();
    ik::IClassFactory * classFactory = scene->classFactory();
    const D3 bottomPoint = m_directionWidget->GetBasePoint();
    const D1 dist = m_directionWidget->GetDirection().norm();
    const D3 topPoint( m_directionWidget->GetBasePoint() + m_directionWidget-
>GetDirection() * dist );

    ik::I13Seg* axisSeg( NULL );
    std::vector< ik::I23Revolution* > surfaces( edges.size(), NULL );
    ik::IBrep *brep = NULL;
    classFactory->createBrep( brep, scene->rootBlock() );

    // 2 Построение
    classFactory->createSegment3D( axisSeg, m_directionWidget->GetBasePoint().ptr() ,
topPoint.ptr() );

    bool faceIsCreated = false;
    TVector<ik::IFaceParametric*> faces;
    for ( int iEdge = 0, nEdge = edges.size(); iEdge < nEdge; ++iEdge )

```

```

{
    ik::I13Seg* curve = NULL;

    // 2.1.1 Кривая для граней
    D3 p0( edges[ iEdge ]->GetIEdge()->startVertex()->point().x, edges[ iEdge
]->GetIEdge()->startVertex()->point().y, edges[ iEdge ]->GetIEdge()->startVertex()-
>point().z );
    D3 p1( edges[ iEdge ]->GetIEdge()->endVertex()->point().x, edges[ iEdge
]->GetIEdge()->endVertex()->point().y, edges[ iEdge ]->GetIEdge()->endVertex()-
>point().z );
    classFactory->createSegment3D( curve, p0.ptr(), p1.ptr() );
    for ( int j = 0; j < 1; j++ ) // угол
    {
        classFactory->createSurfaceOfRevolution( surfaces[ iEdge ], axisSeg,
curve, j * mk::cPI * 0.5, (j + 1) * mk::cPI * 2. );
        classFactory->createFace( faces.add(), brep, surfaces[ iEdge ], false );
    }
    geometry->Update();
    lppApp->GetSession()->GetCurrentViewport()->GetIVisualizer()->Update();

    return true;
}

bool uppCreateSolidRotationByCurveDialog::SaveData()
{
    return true;
}

bool uppCreateSolidRotationByCurveDialog::LoadData()
{
    return true;
}

void uppCreateSolidRotationByCurveDialog::BeforeActivate()
{
}

void uppCreateSolidRotationByCurveDialog::BeforeDeactivate()
{
    if ( uppIVisualizer* visualizer = lppApp->GetSession()->GetCurrentVisualizer() )
    {
        visualizer->GetRenderer()->RemoveActor( m_transformedEdgesActor );
        visualizer->GetRenderer()->RemoveActor( m_brepPreviewActor );
    }
}

void uppCreateSolidRotationByCurveDialog::OnWidgetActivated( const QString& name )
{
}

int uppCreateSolidRotationByCurveDialog::GetEdges( QVector<uppIGeometryEdge*>& edges)
const
{
    edges.clear();
    if ( m_edgesWidget )
    {
        m_edgesWidget->GetSelection()->InitTraversal();
        uppDataTreeNode* object = NULL;
        uppIGeometryEdge* edge = NULL;
    }
}

```

```

        while ( object = m_edgesWidget->GetSelection()->GetNextElement() )
        {
            if( edge = dynamic_cast<uppIGeometryEdge*>( object->GetINode() ) )
                edges.append( edge );
        }

        return edges.size();
    }
}

bool uppCreateSolidRotationByCurveDialog::GetAxis( uppIGeometryEdge** axis ) const
{
    *axis = NULL;
    if ( m_edgesWidget2 )
    {
        m_edgesWidget2->GetSelection()->InitTraversal();
        uppDataTreeNode* object = NULL;
        uppIGeometryEdge* edge = NULL;
        if ( object = m_edgesWidget2->GetSelection()->GetNextElement() )
            *axis = dynamic_cast<uppIGeometryEdge*>( object->GetINode() );
    }
    return *axis != NULL;
}

```