

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТУРИЗМА И  
СЕРВИСА»  
(ФГОУВПО «РГУТИС»)

**Косенко И. И., Кузнецова Л. В.,  
Николаев А. В., Кузнецов Л. Ю.**

**МЕТОДЫ ГЕОМЕТРИЧЕСКОГО  
МОДЕЛИРОВАНИЯ И  
ВИРТУАЛЬНОГО  
ПРОТОТИПИРОВАНИЯ.  
ПЕРСПЕКТИВНЫЕ  
ТЕХНОЛОГИИ**

*Монография*

**Москва 2009**

УДК 004.9:33:658:564

ББК 32.973.2-018.2

О-75

**Рецензенты:**

доктор физико-математических наук,

профессор **Степанов С. Я.**;

доктор технических наук, профессор **Куклин В. Ж.**

**Косенко И. И., Кузнецова Л. В., Николаев А. В., Кузнецов Л. Ю.** Методы геометрического моделирования и виртуального прототипирования. Перспективные технологии: монография. ФГОУВПО «РГУТиС» — М., 2009.

Рассмотрены задачи проектирования (трехмерного геометрического моделирования) деталей машин: валов, зубчатых колес, других элементов редукторов. Приведены методики виртуального прототипирования в CAD/CAM/CAE-системе CATIA V5 на основе твердотельного моделирования изделий сложной геометрической формы на примере зубчатых передач и методика использования виртуального прототипирования в системе CATIA V5 в процессе конструкторско-технологической подготовки производства червячных передач. Анализируется также технология объектно-ориентированного моделирования, реализованная в CAE-системе Dymola, и основанная на применении языка Modelica. Несмотря на лаконичность изложения, представленный материал вполне достаточен для того, чтобы полно представлять набор инструментальных возможностей, получаемых разработчиком виртуальных прототипов технических систем различного назначения.

*Рекомендовано Научно-техническим советом  
Федерального государственного образовательного учреждения  
высшего профессионального образования  
«Российский государственный университет туризма и сервиса»  
(ФГОУВПО «РГУТиС»).*

© Косенко И. И., Кузнецова Л. В., 2009

© ФГОУВПО «РГУТиС», 2009

# Оглавление

|   |    |
|---|----|
| Введение .....  | 6  |
| Глава 1. Основы конструирования в едином информационном<br>пространстве на примере валов и осей.....    | 10 |
| 1.1. Общие сведения .....   | 11 |
| 1.2. Классификация валов .....  | 11 |
| 1.3. Материалы и обработка валов и осей .....   | 15 |
| 1.4. Расчеты на прочность и жесткость .....   | 17 |
| 1.4.1. Порядок расчета осей на статическую прочность.....   | 17 |
| 1.4.2. Ориентировочный расчет на кручение .....   | 19 |
| 1.4.3. Расчет на сложное сопротивление (кручение и изгиб).....  | 21 |
| 1.4.4. Расчет валов (осей) на выносливость (усталостную<br>прочность) .....                             | 26 |
| 1.4.5. Расчет валов на жесткость .....  | 28 |
| 1.4.6. Расчет на виброустойчивость .....  | 29 |
| 1.5. Моделирование вала в системе CATIA V5.....   | 34 |
| 1.6. Классификация подшипников качения .....  | 35 |
| 1.7. Конструкция опор валов.....  | 39 |
| 1.8. Крепление подшипников на валу .....  | 43 |
| 1.9. Методика расчета осей в ANSYS.....   | 45 |
| 1.9.1. Расчет на чистый изгиб (для оси) .....   | 45 |
| 1.9.2. Построение модели .....  | 45 |
| 1.9.3. Проведение расчета .....   | 48 |
| Глава 2. Разработка параметрической модели зубчатой передачи  | 51 |
| 2.1. Параметры и конструкции зубчатых передач.....  | 52 |
| 2.2. Особенности расчета косозубых цилиндрических передач .....   | 57 |
| 2.3. Создание параметрической модели цилиндрической<br>прямозубой передачи в CAD–системе CATIA V5 ..... | 58 |
| 2.4. Создание параметрической модели цилиндрической<br>косозубой передачи в CAD–системе CATIA V5.....   | 72 |
| Глава 3. Червячные передачи .....   | 78 |
| 3.5. Общие сведения .....   | 78 |
| 3.6. Геометрические параметры и способы изготовления<br>червячных передач.....                          | 79 |
| 3.6.1. Расчет геометрии зацепления .....  | 80 |

|  |  |
|--|--|
| 3.6.2. Пример расчета параметров червячного зацепления.....  | 81   |
| 3.6.3. Геометрические показатели качества зацепления.....  | 82   |
| 3.6.4. Способы изготовления.....   | 82   |
| 3.7. Выбор материала и особенности расчета червяка и червячного<br>колеса .....                              | 84   |
| 3.8. Конструкции червячных редукторов .....  | 85   |
| 3.9. Методика создания модели червячного вала в среде системы<br>трехмерного моделирования CATIA V5 .....    | 88   |
| 3.10. Методика создания модели червячного колеса в среде<br>системы трехмерного моделирования CATIA V5 ..... | 97   |
| 3.11. Методика создания сборки червячной пары в среде системы<br>трехмерного моделирования CATIA V5 .....    | 105  |
| Глава 4. Методика расчета червячной передачи .....   | 106  |
| 4.1. Алгоритмическая схема.....  | 106  |
| 4.2. Методика построения геометрической модели<br>цилиндрической червячной передачи .....                    | 107  |
| 4.2.1. Методика построения геометрической модели<br>цилиндрической червячной передачи .....                  | <b>Ошибка! Закладка не<br/>определена.</b> |
| 4.2.2. Построение геометрической модели передачи .....   | 110  |
| 4.3. Определение свойств материалов.....   | 111  |
| 4.4. Установка атрибутов объемных тел .....  | 112  |
| 4.5. Генерирование сетки конечных элементов.....   | 112  |
| 4.6. Определение контактирующих поверхностей.....  | 112  |
| 4.7. Установка граничных и силовых условий .....   | 112  |
| 4.8. Установка параметров решения.....   | 113  |
| 4.9. Анализ результатов .....  | 113  |
| 4.10. Выполнение расчета в ANSYS .....   | 113  |
| Глава 5. Modelica — язык объектно-ориентированного<br>моделирования динамики технических систем .....        | 124  |
| 5.1. Основные понятия.....   | 124  |
| 5.2. Базовые элементы языка .....  | 125  |
| 5.3. Классы.....   | 126  |
| 5.4. Соединения.....   | 130  |
| 5.5. Частичные модели и наследование .....   | 133  |
| 5.6. Параметризация классов .....  | 136  |
| 5.7. Массивы .....   | 140  |



|   |     |
|---|-----|
| 5.8. Циклы.....   | 145 |
| 5.9. Императивное кодирование.....                            | 147 |
| 5.10. Гибридные модели .....                                  | 149 |
| 5.11. Дискретные события и модели с дискретным временем ....  | 151 |
| 5.12. Библиотеки классов.....                                 | 157 |
| 5.13. Трехмерная анимация моделей.....                        | 159 |
| Глава 6. Dymola — визуальный компилятор с языка Modelica..... | 163 |
| 6.1. Возможности Dymola.....                                  | 163 |
| 6.2. Создание проекта.....                                    | 167 |
| 6.3. Отладка проекта .....                                    | 179 |
| 6.4. Развитие проекта .....                                   | 183 |
| 6.5. Пример маятника.....                                     | 186 |
| Библиография.....   | 191 |

# Введение

В промышленности и в системе образования — в учебном процессе и научных исследованиях — важную роль играют различные средства моделирования. Моделирование технических систем особенно трудно ввиду их сложности. Эта сложность имеет многоплановую природу. Ее можно характеризовать не только структурно и параметрически. Особую проблему представляет гетерогенная природа физических законов, используемых в технических системах различного назначения. При моделировании конкретного аппарата или машины можно столкнуться одновременно с механическими, электротехническими, электронными, гидравлическими, пневматическими, тепловыми, химическими процессами, моделями процессов управления, систем массового обслуживания.

Гетерогенная или скорее «мультифизическая» природа присуща, например, задачам оценки времени жизни машин, их узлов и агрегатов. Здесь помимо сложных механических процессов нагружения и деформирования следует рассматривать модели процессов преобразования энергии в тепловую форму и её передачи, инициированные этими процессами химические взаимодействия на поверхности материалов и в приповерхностном слое, кинетические модели накопления микрповреждений. Все эти явления можно описать при помощи систем дифференциально-алгебраических уравнений различного уровня сложности, описывающих механическую, топахимическую и трибомеханическую эволюцию в рамках единой модели.

С другой стороны, в процессе активной унификации понятийного аппарата и инструментальных средств значительный прогресс наметился в направлении построения и применения языка объектно-ориентированного моделирования Modelica, целиком сконструированного на базе объектно-ориентированной парадигмы (<http://www.modelica.org>). В основе соответствующего подхода лежит методика предварительного анализа абстракций, максимально адекватно описывающих предметную область. В процессе реализации системы моделирования из этих абстракций впоследствии, на этапе проектирования, возникают классы, обеспечивающие последующее компонентное построение конкретной модели.

К области приложений языка Modelica относится широкий спектр задач из сферы производства и обслуживания автомобильного, железнодорожного, авиационного транспорта; химического, сельскохозяйственного производства; муниципального хозяйства, экологии; производства и передачи электроэнергии; систем анало-

гового и дискетного управления техническими объектами и технологическими процессами и многих других видов деятельности, где в той или иной форме можно создать описание процессов в виде динамических моделей высокой степени сложности.

Одной из главных задач здесь является технология создания адекватных виртуальных прототипов технических систем или технологических процессов. Инновационная сущность виртуального прототипирования состоит в том, что оно позволяет от 80 до 90% и более всех требующихся при создании нового продукта испытаний провести не на реальном образце, а на его виртуальном аналоге (прототипе), т. е. на соответствующей компьютерной модели. Время выполнения проектных работ и их стоимость сокращаются на порядок. Подобные технологии создания новых изделий активно используются компаниями ведущих мировых промышленных центров.

Виртуальное прототипирование применяется на всех этапах жизненного цикла изделия: анализа, проектирования, разработки, производства, эксплуатации и утилизации. Например, системы управления жизненным циклом, PLM-системы, могут использовать виртуальные прототипы во время эксплуатации изделия для обеспечения оптимальных режимов его работы, диагностики и др. Подводя итог сказанному выше, отметим безусловную актуальность обсуждения важных проблем применения одного из наиболее продвинутых ресурсов современных CAE-систем — языка Modelica. В дополнение к этому не менее актуальна задача построения интегрированных компьютерных моделей, обеспечивающих реализацию новых методов оценки процессов старения в изделиях для более точного прогнозирования времени жизни машин, их узлов и агрегатов.

Данное обстоятельство наилучшим образом соответствует функциональным возможностям современных универсальных средств компьютерного моделирования технических систем. Эти средства чаще всего основываются на единых математических формализмах представления динамики, получивших значительное развитие в самом конце прошлого века таких, как теория бондграфов, позволяющая обеспечить унифицированное представление динамической модели технической системы при помощи графов перетоков энергии между её подсистемами. Такое программное обеспечение обычно имеет визуальный интерфейс, обеспечивающий создание моделей достаточно высокой структурной сложности.

Применение на практике технологий виртуального прототипирования на примерах разработок ведущих производителей программных продуктов для организации совместной деятельности научного сообщества, преподавателей, инженеров позволяет обеспечить должный уровень научных изысканий и одновременно повысить качество учебного процесса в области технических и технологических дисциплин. Изучение и применение программных комплексов высокого уровня таких, как CAD/CAE/CAM/PLM-система CATIA от Dassault Systèmes позволяет создавать трехмерные solid-модели технических систем, которые затем могут быть рассчитаны и исследованы без создания натуральных образцов. Появляется возможность вариативного проектирования прототипов технических систем и моделирования их поведения в виртуальной динамической лаборатории Dymola от Dynasim AB, ныне отделения Dassault Systèmes. Более того, появившийся не так давно и уже активно используемый в промышленности, новый продукт CATIA V6, помимо традиционных CAD-задач и новых сетевых и мультимедийных технологий, полностью интегрировал в себя средства визуального компилятора Dymola.

Содержание предлагаемой вниманию читателя книги построено на рассмотрении нескольких примеров инженерной направленности. Эти примеры могут быть использованы в учебном процессе в рамках курсов «Объектно-ориентированное моделирование», «Математическое моделирование», «Системы автоматизированного проектирования», «Конструирование», «Детали машин», «Компьютерная геометрия и графика», «Графическое моделирование», «Инженерная графика» и др. Кроме того, рассматриваемый материал можно использовать для формирования специальных курсов для магистров и аспирантов, а также курсов повышения квалификации в системе дополнительного профессионального образования.

Весь, представленный в монографии материал естественным образом можно разделить на две части. В первой части — это главы с первой по четвертую — рассмотрены задачи проектирования (трехмерного геометрического моделирования) деталей машин: валов, зубчатых колес, других элементов редукторов. Здесь приведены методики виртуального прототипирования в системе CATIA V5 на основе твердотельного моделирования изделий сложной геометрической формы на примере зубчатых передач и методика использования виртуального прототипирования в системе CATIA V5 в процессе конструкторско-технологической подготовки производства червячных передач.

Все примеры построения геометрии выполнены с использованием известной CAD/CAM/CAE-системы CATIA V5. Анализ примеров может быть полезен для понимания принципов работы «геометрии» CATIA при проектировании геометрии узлов и деталей машин. Силовой расчет рассмотренных узлов дополняется конечно-элементным анализом и расчетом напряжений и деформаций, выполненным при помощи еще одного «тяжелого» программного продукта — ANSYS.

Вторая часть, состоящая из глав пятой и шестой, целиком посвящена технологии объектно-ориентированного моделирования, реализованной в CAE-системе Dymola, и основанная на применении языка Modelica. Несмотря на лаконичность изложения, представленный материал вполне достаточен для того, чтобы полно представлять набор инструментальных возможностей, получаемых разработчиком виртуальных прототипов технических систем различного назначения. В центре внимания находится парадигма объектно-ориентированного подхода и особенности её применения при реализации динамических моделей.

Главы с 1-й по 4-ю монографии написаны Л. В. Кузнецовой, А. В. Николаевым, Л. Ю. Кузнецовым, а главы 5, 6 — И. И. Косенко.

# Глава 1. Основы конструирования в едином информационном пространстве на примере валов и осей

Современные возможности вычислительной техники и программного обеспечения позволяют по-новому подойти к процессу проектирования изделий. Для обеспечения роста конкурентоспособности продукции и эффективности производства необходимо внедрение сквозной **информационной поддержки** на всех стадиях жизненного цикла **изделия** (ИПИ). В полной мере данная концепция реализуется при помощи PDM-технологии. PDM-технология призвана повысить эффективность управления информацией об изделии за счет представления данных в электронном виде и организации оперативного доступа к этим данным.

Оперативный доступ к данным об изделии обеспечивается интеграцией всех данных об изделии в логически единую модель. В свою очередь, совокупность информационных процессов, управляемых PDM-системой, является электронным документооборотом — основой при построении **единого информационного пространства** для всех участников жизненного цикла (ЖЦ) изделия.

На начальном этапе проектирования изделия наполнением единого информационного пространства является нормативно-технический и справочный материал, на основе которого производится создание автоматической системы расчета параметров изделия. По расчетным параметрам в CAD-системе строится трехмерная модель изделия, работа с которой обеспечивает наглядность и гибкость при его проектировании. Готовая модель изделия анализируется в CAE-системе, которая позволяет производить различные виды расчетов при заданных условиях (материал изделия, температура, величина нагрузки и пр.) По результатам анализа изделия вносятся необходимые изменения в его конструкцию или характеристики. Далее по готовой модели в САМ-системе создается управляющая программа для станков с ЧПУ.

Таким образом, использование безбумажного принципа проектирования изделия позволяет существенно снизить затраты на всех этапах ЖЦ изделия и избежать ошибок и несоответствий при изготовлении, сборке и дальнейшей эксплуатации изделия, обес-

печивая тем самым повышение качества продукции при одновременном сокращении сроков выхода изделия на рынок. В полной мере данный подход реализуется и при проектировании механических конструкций, рассматриваемых ниже.

## **1.1. Общие сведения**

Осью называется деталь, предназначенная только для поддержания установленных на ней деталей. В отличие от вала ось не передает вращающего момента и работает только на изгиб. В машинах оси могут быть неподвижными или же могут вращаться вместе с сидящими на них деталями (подвижные оси).

Оси предназначены для соединений в механизмах общего назначения. Оси изготавливают гладкие и с буртиком. Допускается изготовление с углублением на торце или сквозным отверстием, концентричным наружной цилиндрической поверхности.

Валом называется деталь (как правило, гладкой или ступенчатой цилиндрической формы), предназначенная для поддержания установленных на ней шкивов, зубчатых колес, звездочек, катков и т. д., и для передачи вращающего момента.

При работе вал испытывает изгиб и кручение, а в отдельных случаях помимо изгиба и кручения валы могут испытывать деформацию растяжения (сжатия).

## **1.2. Классификация валов**

По условиям работы различаются валы:

а) коренные, или основные, передающие или воспринимающие работу двигателя; коренной вал может быть прямым, кривошипным или коленчатым;

б) передаточные, являющиеся одновременно ведомыми и ведущими;

в) трансмиссионные, получающие работу от двигателя и передающие её отдельным рабочим машинам;

г) контрприводные, имеющие назначение менять число оборотов ведомых валов относительно ведущих и в случае необходимости — направление вращения.

По конструкции различаются валы:

а) гладкие, имеющие на всём своём протяжении сечения одинаковых форм и размера;

б) фасонные с различными поперечными сечениями;

в) фланцевые оканчивающиеся соединительным фланцем.

По типу сечения различаются валы:

- а) сплошные;
- б) полые (облегчённые валы), обычно кольцевого сечения;
- в) составные из нескольких частей, жёстко соединяемых между собой.

Особый класс представляют гибкие валы, допускающие передачу вращения под любым углом и состоящие из предохранительного рукава (оболочки) и сердцевины из прямой проволоки, окружённой несколькими соосными слоями витой проволоки.

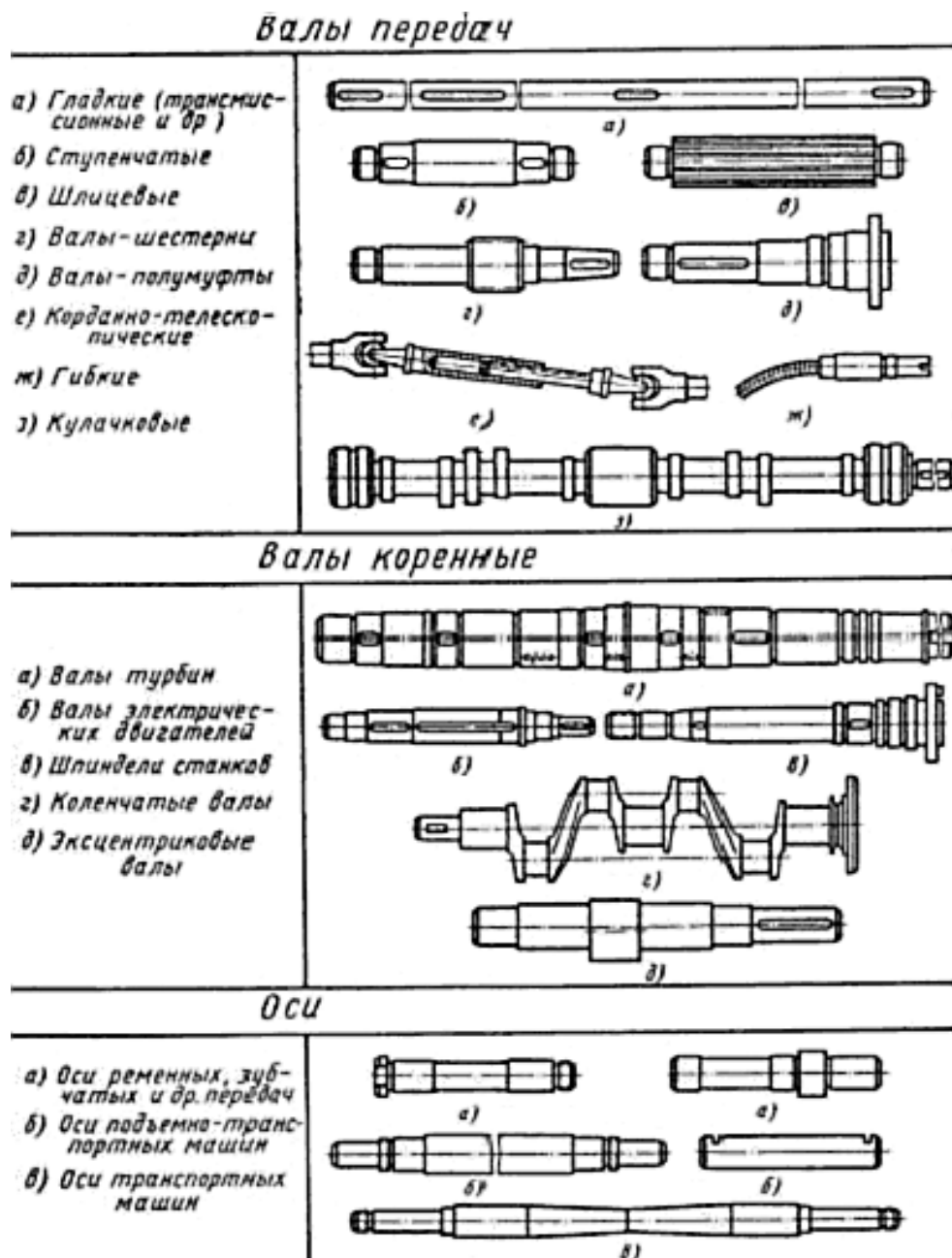


Рис. 1.2.1. Классификация валов.

Форма вала (или оси) определяется его напряжённостью при принятых габаритах, положением и способом крепления деталей, с



ним связанных, способом регулировки его положения, посадками, типом и размерами подшипников и условиями обработки и сборки. Форма вала должна быть согласована с указанными факторами, из которых часть выбирается в процессе конструирования, а часть заранее задана.

В целях уменьшения веса и габаритов сильно нагруженные валы ответственных машин изготавливаются обычно фасонными и часто полыми.

Вал, рассчитанный с учетом действующих на него моментов так, что сечения его равнопрочны, имеет форму, близкую к форме тела равного сопротивления. Между отдельными участками вала сопряжения делаются плавными в виде галтелей возможно больших радиусов или конических переходов. Такой вал изображен на Рис. 1.2.2. Пунктиром отмечена форма тела равного сопротивления.

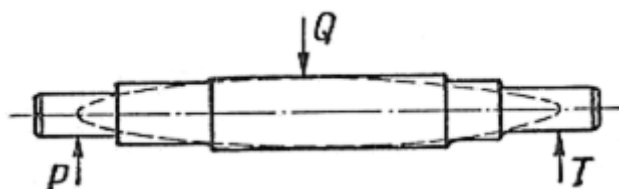


Рис. 1.2.2. Вал.

Пустотелые валы и оси выгодны в отношении использования материала. Пустотелые валы вследствие их малого веса и относительно большого момента сопротивления особенно пригодны для передач с большими расстояниями между отдельными опорами. Иногда пустотелые валы применяют в тех случаях, когда они получают усилия от проходящего сквозь них сплошного вала.

Переходные участки (галтели) между ступенями валов и осей выполняются для снижения концентрации напряжений и увеличения долговечности.

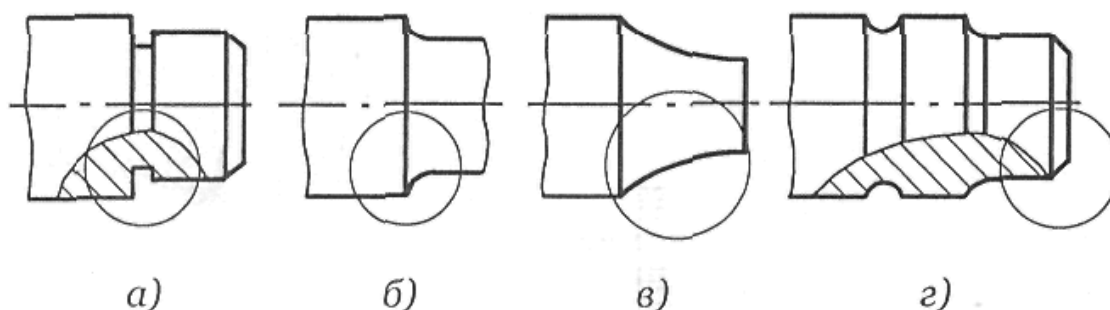


Рис. 1.2.3. Вал.

Конструктивные разновидности переходных участков между ступенями валов и осей: канавка со скруглением для выхода шлифовального круга (Рис. 1.2.3, а); галтель постоянного радиуса (Рис.

1.2.3, б); галтель переменного радиуса (Рис. 1.2.3, в). Торцы валов и осей делаются с фасками (Рис. 1.2.3, г). Для осевого фиксирования вала и деталей на валу используются уступы самого вала (Рис. 1.2.4, а), заплечики (Рис. 1.2.4, б), конические участки вала (Рис. 1.2.4, в) и стопорные кольца (Рис. 1.2.4, г).

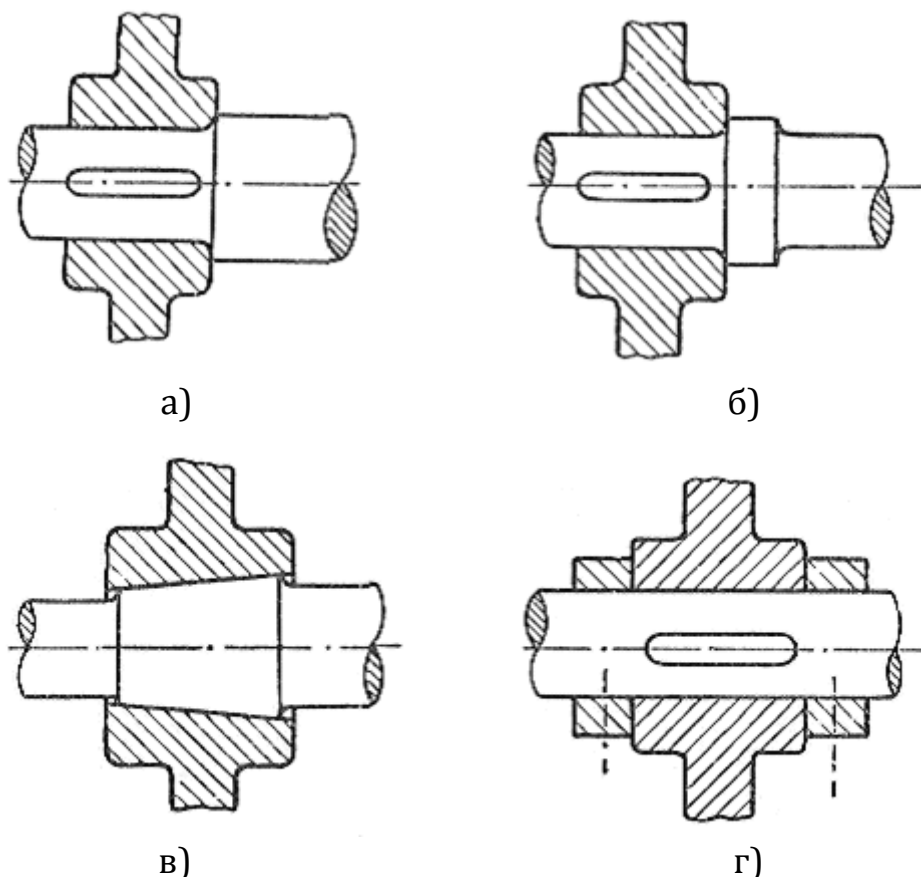


Рис. 1.2.4. Осевое фиксирование вала и деталей на валу.

Заплечики, составляющие одно целое с валом, применяются там, где они не мешают сборке цельных деталей (зубчатых колёс, шкивов). Однако они дороги, так как приходится стачивать много материала или высаживать их. На гладкие валы при больших продольных усилиях заплечики надеваются в горячем состоянии, а при продольных усилиях средней величины заплечики лучше заменять стопорными кольцами.

Концы валов могут быть цилиндрические и конические. Для цилиндрических валов исполнения 1 используют сегментные шпонки (ГОСТ 24071-80) для вала диаметром  $d$  до 14мм; призматические обыкновенные (ГОСТ 23360-78) для вала диаметром  $d$  свыше 12мм; тангенциальные нормальные (ГОСТ 24069-80).

Для валов исполнения 2 используются шпонки призматические обыкновенные (ГОСТ 23360-78) для вала диаметром  $d$  до

30мм; призматические высокие (ГОСТ 10748-79) и тангенциальные усиленные (ГОСТ 24070-80) для вала диаметром свыше 30мм.

Для редукторов и мотор-редукторов допускается в соответствии с ГОСТ 24266-94 исполнение концов валов с двумя шпоночными пазами, расположенными под углом 120°.

На предел усталости вала влияют конфигурация вала и сидящих на нём деталей, способы их крепления и величины натягов. Давление на вал в результате натяга от сидящей на валу детали снижает предел усталости. Неблагоприятно влияет на прочность вала резкое возрастание давлений. Целесообразны такие формы деталей, при которых удельные давления нарастают постепенно.

### **1.3. Материалы и обработка валов и осей**

Для осей и валов применяются разнообразные марки углеродистых и легированных сталей.

Валы и оси в зависимости от предъявляемых к ним требований могут подвергаться термической и термохимической обработке (цементация, азотирование).

Марка стали и термообработка обеспечивают достаточную прочность вала (оси) и достаточное сопротивление истиранию. На выбор марки стали и её термообработки влияют условия работы вала, его напряженность при заданных габаритах, скорость вращения, назначение и конструкция вала и его опор.

Для неответственных малонагруженных конструкций валов и осей применяются углеродистые стали без термической обработки. Ответственные тяжело нагруженные валы изготавливаются из легированной стали 40ХНМА, 25ХГТ и др. Без термической обработки применяются стали 35 и 40, Ст5, Ст6, 40Х, 40ХН, 30ХНЗА, с термической обработкой — стали 45, 50 и др.

Особенно часто применяется сталь 45. Для изготовления тяжело нагруженных, фасонных и шлицевых валов часто применяются легированные стали.

Применение дорогостоящих легированных сталей вызывается требованиями ограничения веса и габаритов вала и повышением износоустойчивости его элементов.

Термохимическая обработка применяется в случаях, когда необходимо повышение износоустойчивости шеек вала. Помимо этого цементирование, а особенно азотирование уменьшают влияние концентрации напряжений на предел усталости.

В отдельных случаях валы и оси изготавливаются из стального литья, чугуна и дерева.

Таблица 1.3.1

## Характеристика основных марок сталей, применяемых для изготовления валов

| Марка стали  | Диаметр заготовки, мм                           | Твердость, НВ, не менее |     | $\sigma_B$ , МПа | $\sigma_T$ , МПа         | $\tau_T$ , МПа           | $\sigma_{-1}$ , МПа   | $\tau_{-1}$ , МПа     | $\psi_\sigma$    | $\psi_\tau$      | ГОСТ    | Качество поверхности в зависимости от назначения |
|--|---|-------------------------|-----|------------------|--------------------------|--------------------------|-----------------------|-----------------------|------------------|------------------|---------|--|
| Ст. 5 пс3<br>Ст. 5 сп3<br>ВСт. 5пс2<br>ВСт. 5сп2   | До 20<br>Свыше 20 до 40<br>» 40 до 100<br>» 100 | —<br>—<br>—             |     | 500...640        | 290<br>280<br>270<br>260 | 170<br>170<br>160<br>160 | $(0,4...0,5)\sigma_B$ | $(0,4...0,5)\sigma_B$ | 0<br>0<br>0<br>0 | 0<br>0<br>0<br>0 | 380-71  | Три группы качества                              |
| Сталь 45<br>(45-2)<br>(45-3)   | Свыше 5   | 229*                    | 197 | 610              | 360                      | 210                      | 270                   | 150                   | 0,1              | 0,05             | 1050-74 | То же  |
| 40Х<br>40ХН  | Свыше 5   | 217<br>229              |     | 1000             | 800                      | 480                      | 450                   | 250                   | 0,15             | 0,1              | 4543-71 | Две группы качества                              |
| 12ХН3А   | Любого диаметра                                 | 217                     |     | 950              | 700                      | 420                      | 420                   | 240                   | 0,15             | 0,1              |         |  |
| 18ХГТ  | То же   | 217                     |     | 1000             | 900                      | 540                      | 450                   | 250                   | 0,15             | 0,1              |         |  |
| 30ХГТ  | "   | 229                     |     | 1500             | 1300                     | 780                      | 670                   | 380                   | 0,15             | 0,1              |         |  |
| Примечание.* Без термообработки: ГОСТ 380-71 — сталь углеродистая обыкновенного качества; ГОСТ 1-50-74 — сталь углеродистая качественная конструкционная; ГОСТ 4543-71 — сталь легированная конструкционная. |   |                         |     |                  |                          |                          |                       |                       |                  |                  |         |  |

## 1.4. Расчеты на прочность и жесткость

Валы и оси в основном испытывают циклически меняющиеся напряжения. Отсюда следует, что основным критерием работоспособности валов и осей является усталостная прочность. Усталостная прочность (выносливость) валов и осей оценивается коэффициентом запаса прочности.

Неподвижные оси при действии постоянных нагрузок рассчитываются только на статическую прочность. Подвижные быстроходные оси и валы рассчитываются на выносливость. Тихоходные валы и оси, нагруженные переменной нагрузкой, рассчитываются на статическую прочность и выносливость.

Основными расчетными силовыми факторами для осей и валов являются изгибающие  $M_{\text{и}}$  и крутящие  $M_{\text{к}}$  (только для валов) моменты. Влияние растягивающих и сжимающих сил незначительно и, поэтому, как правило, в расчетах не учитывается.

Спроектированные валы и оси с учетом обеспечения статической или усталостной прочности иногда выходят из строя вследствие недостаточной их жесткости или из-за вибрации. Кроме того, малая жесткость нарушает нормальную работу зубчатых передач и подшипников. Валы и оси дополнительно рассчитываются на жесткость и колебания. Жесткость валов и осей оценивается величиной прогиба в местах установки деталей или углом закручивания сечений; колебания — критической угловой скоростью.

Ось не передает вращающего момента, а воспринимает только поперечные нагрузки, рассчитывают только на изгиб.

### 1.4.1. Порядок расчета осей на статическую прочность

1) По конструкции узла (Рис. 1.4.1, а) составляется расчетная схема (Рис. 1.4.1, б), определяются силы, действующие на ось, строятся эпюры изгибающих моментов;

2) Диаметр оси определяется по формуле:

а) для цилиндрических сплошных осей (Рис. 1.4.2, а):

$$d = \sqrt[3]{\frac{M_{\text{и}}}{0,1[\sigma_{\text{из}}]}}$$

или  $M_{\text{и}} = 0,1d^3[\sigma_{\text{из}}]$ ;

б) для цилиндрических полых осей (Рис. 1.4.2, б):

$$M_{\text{и}} = 0,1 \frac{d^4 - d_0^4}{d} [\sigma_{\text{из}}],$$

где  $M_{из}$  — изгибающий момент Н·мм,  $[\sigma_{из}]$  — допускаемое напряжение на изгиб МПа и  $d, d_0$  — соответственно наружный и внутренний диаметры полый оси.

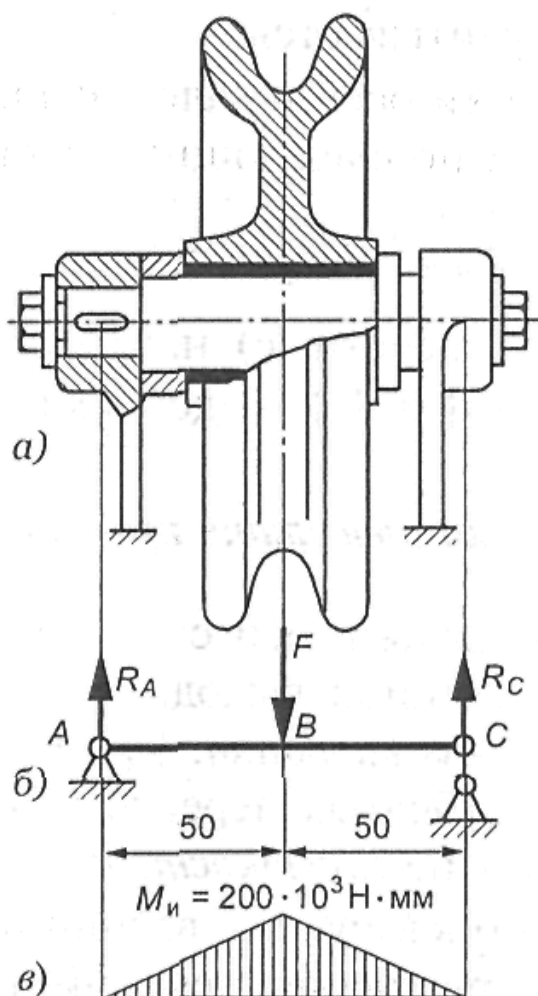


Рис. 1.4.1. Расчетная схема оси: а) конструкция; б) расчетная схема; в) эпюра изгибающих моментов.

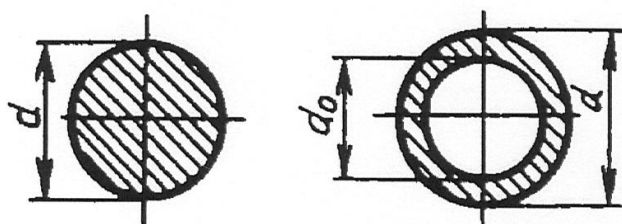


Рис. 1.4.2. а) цилиндрическая сплошная ось (вал); б) цилиндрическая полая ось.

### 3) Выбор $[\sigma_{из}]$ :

Во вращающихся осях напряжение изгиба изменяется по симметричному циклу: для них принимается, что  $[\sigma_{из}] = [\sigma_{-1из}] = (0,5 \div 0,6)[\sigma_{из}]$ , в неподвижных  $[\sigma_{из}] = [\sigma_{0из}]$ , во вращающихся осях из Ст. 5  $[\sigma_{из}] = 50 \div 80$  МПа, для не вращающихся  $[\sigma_{из}] = 100 \div 160$  МПа

(меньшие значения рекомендуется принимать при наличии концентраторов напряжений).

4) Полученное значение диаметра оси  $d$  округляется до ближайшего большего стандартного размера: 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 30, 32, 34, 36, 38, 40, 42, 45, 48, 50, 52, 55, 60, 63, 65, 70, 75, 80, 85, 90, 95, 100. Если ось в расчетном сечении имеет шпоночную канавку, то ее диаметр увеличивают на 10%.

5) Проводится проверочный расчет оси на статическую прочность:

$$\tau_{из} = \frac{M_{из}}{0,1d^3} \leq [\sigma_{из}],$$

где  $[\sigma_{из}]$  — расчетное напряжение изгиба в опасном сечении.

Порядок расчета валов:

- ориентировочный расчет на кручение;
- расчет на сложное сопротивление (кручение и изгиб);
- проверка запаса прочности по выносливости в наиболее опасных сечениях;
- проверка жесткости вала;
- проверка на виброустойчивость.

Приближенный расчет по пунктам 1 и 2 выполняется как проектный, на основе которого ориентировочно устанавливаются диаметры характерных сечений вала с последующим уточнением коэффициентов запаса прочности по выносливости.

### 1.4.2. Ориентировочный расчет на кручение

При этом расчете обычно определяется диаметр выходного конца вала или диаметр вала под подшипником (под опорой), который испытывает только кручение.

Исходя из условия прочности получаем величину

$$d = \sqrt[3]{\frac{M_{кр}}{0,2[\tau]_{кр}}}$$

и проверочный расчет

$$\tau_{кр} = \frac{M_{кр}}{0,2d^3} \leq [\tau]_{кр},$$

где  $d$  — расчетный диаметр вала,  $M_{кр}$  — крутящий момент в опасном сечении вала;  $\tau_{кр}$  и  $[\tau]_{кр}$  — расчетное и допустимое напряжения кручения в опасном сечении вала.

За расчетный момент принимается наибольший длительно действующий момент. Расчетный вращающий момент на валу:

$$T = \frac{T_d \eta}{i}, T = 974000 \frac{P_d \eta}{n},$$

где  $T_d$  — момент на валу двигателя, Н·см;  $\eta$  — КПД участка кинематической цепи от двигателя до рассчитываемого вала;  $i = n/n_0$  — передаточное отношение от двигателя до вала;  $n$  — расчетная частота вращения вала, мин<sup>-1</sup>;  $n_0$  — частота вращения вала двигателя, мин<sup>-1</sup>;  $P_d$  — мощность на валу двигателя.

Для определения вращающего момента по мощности и частоте вращения можно пользоваться номограммой. Величины допускаемых вращающих моментов  $T$ , Н·м, вычисляются по формуле  $T = 10^{-3} K d^3$ , где  $d$  — диаметр конца вала, мм;  $K = \frac{\pi}{16} [\tau]$  в МПа.

Таблица 1.4.1

Значения коэффициента  $K$  в зависимости от характера нагрузки, прочности и твердости материала вала

| Предел прочности $\sigma_b$ , МПа | Твердость вала, НВ              | Значение коэффициента $K$ , МПа, при |      |     |  |     |                   |     |
|-----------------------------------|---------------------------------|--------------------------------------|------|-----|--|-----|-------------------|-----|
|                                   |                                 | чистом кручении                      |      |     | кручении плюс изгиб от радиальной нагрузки |     |                   |     |
|                                   |                                 |                                      |      |     | $F \leq 250\sqrt{T}$                       |     | $F > 250\sqrt{T}$ |     |
|                                   |                                 | a                                    | b    | c   | a  | b   | a                 | b   |
| От 500 до 850<br>Св. 850 до 1200  | От 145 до 250<br>Св. 250 до 350 | 8                                    | 5,6  | 4   | 5,6  | 4   | 2,8               | 2   |
|                                   |                                 | 11,2                                 | 8    | 5,6 | 8  | 5,6 | 4                 | 2,8 |
| Св. 1200                          | Св. 350                         | 16<br>22,4                           | 11,2 | 8   | 11,2                                       | 8   | 5,6               | 4   |

Допускаемые вращающие моменты, передаваемые концами валов, определяются по таблице по диаметру и коэффициенту  $K$ .

Таблица 1.4.2

Допускаемые напряжения кручения, соответствующие значениям коэффициента  $K$

|                  |     |     |     |     |     |      |      |      |
|------------------|-----|-----|-----|-----|-----|------|------|------|
| $K$ , МПа        | 2,0 | 2,8 | 4,0 | 5,6 | 8,0 | 11,2 | 16,0 | 22,4 |
| $[\tau]_k$ , МПа | 10  | 14  | 20  | 28  | 40  | 56   | 80   | 112  |



### 1.4.3. Расчет на сложное сопротивление (кручение и изгиб)

Участок вала между опорами (под шестерней, колесом и т. п.) рассчитывается на совместное действие кручения и изгиба по приведенному моменту  $M_{\text{пр}} \cdot \sqrt{M_{\text{и}}^2 + M_{\text{к}}^2} = M_{\text{пр}}$  — приведенный (результрующий) момент в опасном сечении, Н·мм,  $M_{\text{и}}$  и  $M_{\text{к}}$  — соответственно изгибающий и крутящий моменты.

Тогда диаметр имеет вид:  $d = \sqrt[3]{\frac{M_{\text{пр}}}{0,1[\sigma_{\text{из}}]}}$ , а проверочный расчет

— вид:  $\sigma_{\text{пр}} = \frac{M_{\text{пр}}}{0,1d^3} \leq [\sigma]_{\text{пр}}$ , где  $\sigma_{\text{пр}}$  — приведенное напряжение расчетного сечения вала.

Порядок расчета:

- 1) По чертежу узла составляется расчетная схема (Рис. 1.4.3, а).
- 2) Определяются действующие на вал силы; если они действуют не в одной плоскости, то их необходимо разложить по двум взаимно перпендикулярным плоскостям. При угле между плоскостями менее  $30^\circ$  все силы можно рассматривать как действующие в одной плоскости.

Основные нагрузки на валы создают силы, действующие в зубчатом и червячном зацеплениях. При определении направления сил следует учитывать, что на ведомом колесе окружная сила является движущей и направлена в сторону вращения. На ведущей шестерне (или червяке) окружная сила является реакцией со стороны ведомого колеса и направлена к стороне, противоположную вращению. Радиальное усилие на цилиндрических колесах внешнего зацепления направлено к центру колеса (к оси червяки). Направление осевого усилия зависит от направления спирали и направления вращения. Как и полное давление, действующее на зуб, оно направлено внутрь зуба.

В схеме (Рис. 1.4.3, а)  $M_{\text{к}}$  — крутящий момент, возникающий в поперечных сечениях вала;  $F_{\text{в}}$  и  $F_{\text{г}}$  — силы, действующие на вал в вертикальной и в горизонтальной плоскостях.

- 3) Определяются опорные реакции:

- в вертикальной плоскости  $R_{\text{Ав}} = R_{\text{Св}} = \frac{F_{\text{в}}}{2}$ ;
- в горизонтальной плоскости  $R_{\text{Аг}} = R_{\text{Сг}} = \frac{F_{\text{г}}}{2}$ .

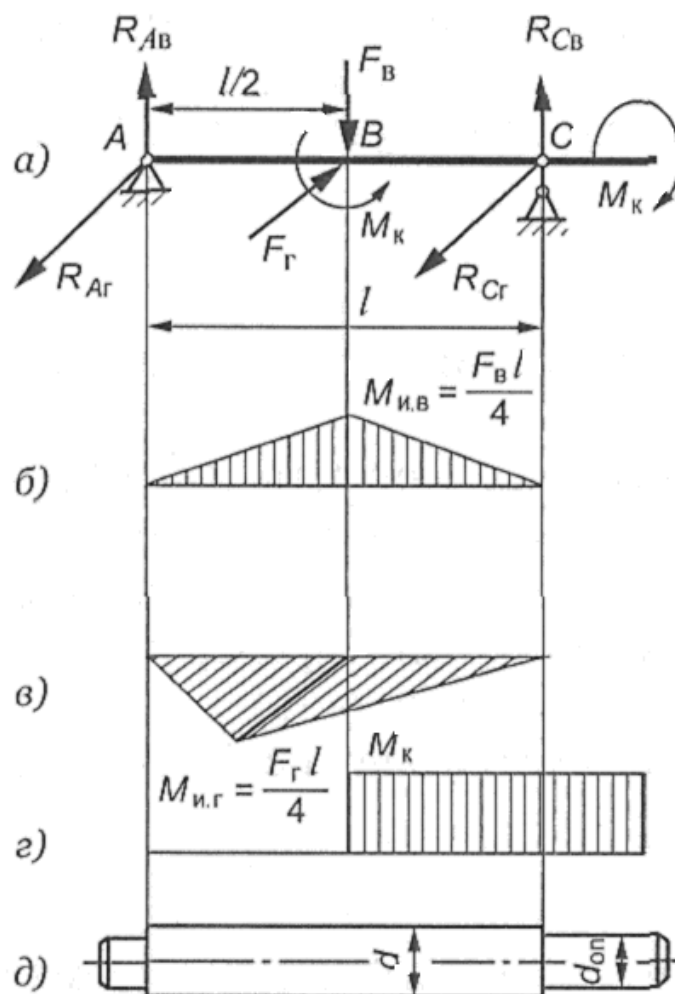


Рис. 1.4.3. Расчетная схема валов: а) схема нагружения; б) эпюра изгибающего момента в вертикальной плоскости; в) эпюра изгибающего момента в горизонтальной плоскости; г) эпюра крутящего момента; д) эскиз вала.

При расчете вал принимается за балку, лежащую на шарнирных опорах. Эта расчетная схема точно соответствует действительному положению только для валов на подшипниках качения, установленных по одному или по два в опоре; при двух подшипниках должна быть обеспечена самоустанавливаемость опоры; например, установкой конических роликоподшипников вершинами роликов в разные стороны.

Для других опор такую расчетную схему можно применять как приближенную. При длинных самоустанавливающихся подшипниках скольжения, расположенных по концам вала, равнодействующую реакции подшипника следует предполагать приложенной к точке, отстоящей от его кромки со стороны пролета на  $1/3-1/4$  длины подшипника.

При расчете валов, вращающихся в длинных подшипниках скольжения ( $l/d = 3$ ), расчетная схема приближается к схеме балки с заделанными концами.

В случае применения в опоре одного шарикового или роликового радиального подшипника расчетная точка опоры располагается посередине ширины подшипника (Рис. 1.4.4, а). При одинарном радиально-упорном подшипнике (Рис. 1.4.4, б, в) радиальная реакция считается приложенной к валу в точке пересечения его геометрической оси и прямой, проведенной через центр шарика или середину ролика под углом  $(90^\circ - \alpha)$  к оси подшипника, где  $\alpha$  — угол контакта, указанный в каталогах подшипников.

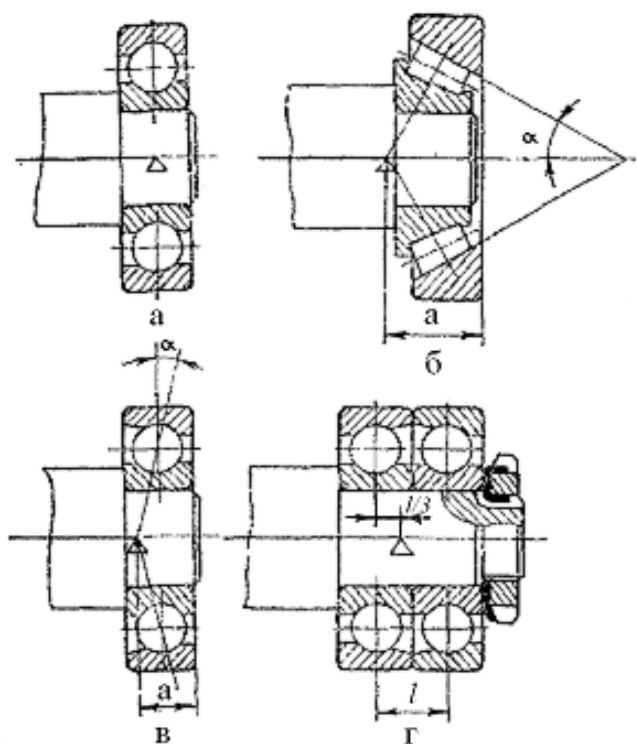


Рис. 1.4.4. Расположение расчетных точек опор.

Расстояние между точкой приложения реакции и широким торцом наружного кольца подшипника (Рис. 1.4.4) приближенно может быть найдено по следующим формулам:

- для однорядных шариковых радиально-упорных подшипников:  

$$a = 0,5[B + 0,5(d + D)\operatorname{tg} \alpha];$$
- для однорядных конических роликовых подшипников:  

$$a = \frac{T}{2} + \frac{(d + D)}{6}e, \text{ где } e = 1,5 \operatorname{tg} \alpha.$$

Аналогичным образом ведется расчет и в случае, если в опоре установлены двойной подшипник или два одинаковых подшипника, но затяжка подшипника такова, что одновременно работает только одни ряд тел качения.

Если затяжка такова, что одновременно работают оба ряда тел качения и осевая нагрузка должна учитываться при расчете, условно местом приложения реакции считается точка, отстоящая от середины подшипника, который воспринимает осевую нагрузку, на  $1/3$  расстояния между рядами тел качения (рис. 1.4.4, г). Приблизительно допускается совмещение расчетного положения опоры с серединой подшипника, воспринимающего осевую нагрузку. Если осевая нагрузка при расчете подшипников учитываться не должна, то расчетная точка опоры размещается между подшипниками.

Действующие на вал расчетные силы и моменты считаются сосредоточенными и расположенными посередине длины воспринимающих их элементов. Погрешность расчета идет при этом в запас прочности.

На рис. 1.4.5, 1.4.6 приведены формулы для определения реакций опор и изгибающих моментов двухопорных валов с характерными случаями нагружения.

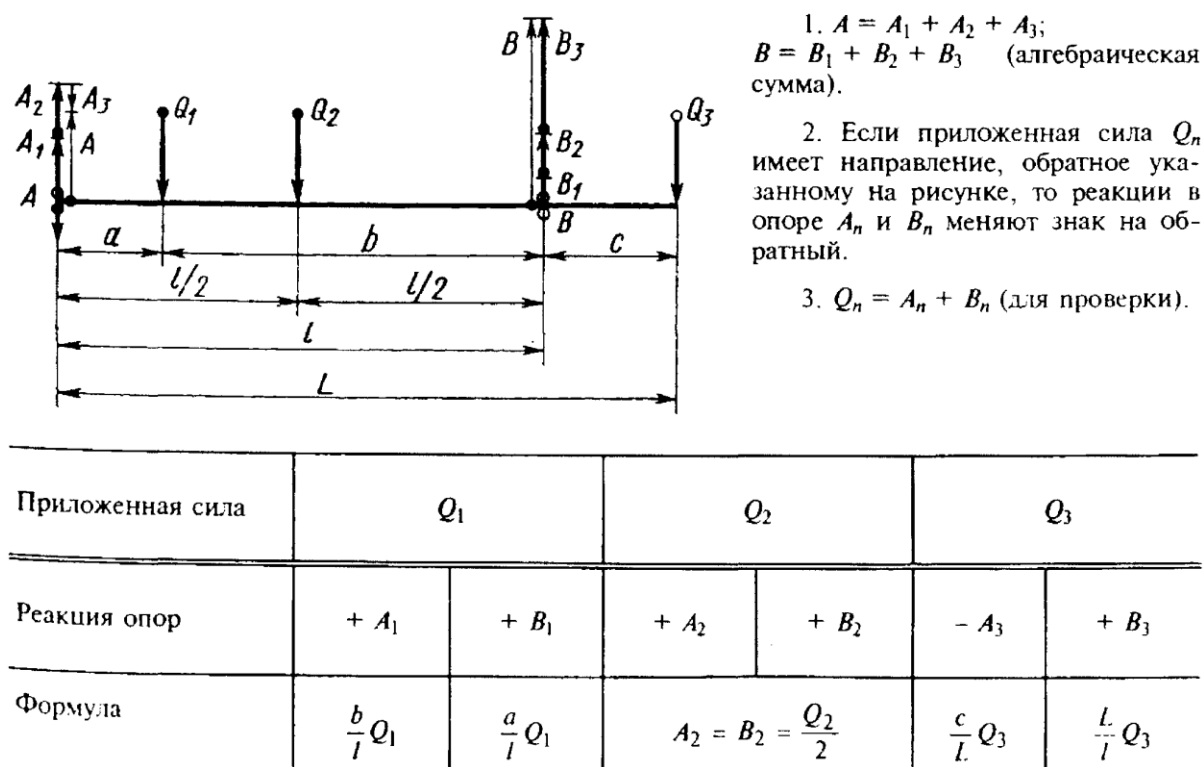


Рис. 1.4.5. Определение реакций опор.

4) Изгибающие моменты  $M_{и}$  и их эпюры:

- в вертикальной плоскости — в сечениях А и С:  $M_{и.в.} = 0$ ;
- в сечении В:  $M_{и.в.} = \frac{(R_{AB}l)}{2} = \frac{F_B l}{4}$  (Рис. 1.4.3, б); в горизонтальной плоскости — в сечении А и С:  $M_{и.г.} = 0$ ;

– в сечении B:  $M_{и.г.} = \frac{(R_{A\Gamma}l)}{2} = \frac{F_{\Gamma}l}{4}$  (Рис. 1.4.3, в);

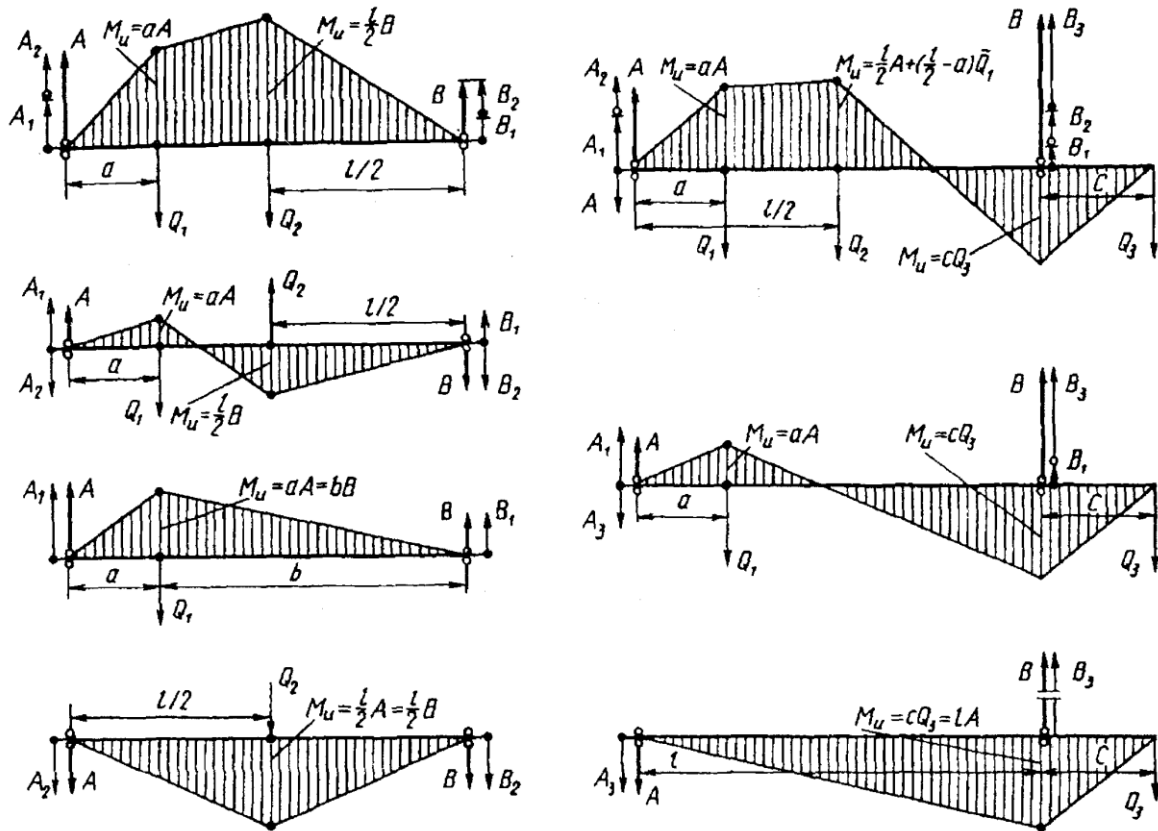


Рис. 1.4.6. Определение реакций опор и изгибающих моментов.

5) Суммарный изгибающий момент в сечении B:

$$M_{\text{сум.и}} = \sqrt{M_{и.в}^2 + M_{и.г}^2} = \sqrt{\left(\frac{F_{\text{в}}l}{4}\right)^2 + \left(\frac{F_{\Gamma}l}{4}\right)^2}.$$

Определяется крутящий момент и строится эпюра (Рис. 1.4.3, г):

$$M_{\text{к}} = T_{\text{к}}, \sqrt{M_{\text{сум.и}}^2 + M_{\text{к}}^2} = M_{\text{пр}}$$

7) Определяется диаметр вала между опорами по формуле:

$$d = 3 \sqrt[3]{\frac{M_{\text{пр}}}{0,1[\sigma_{\text{из}}]}}.$$

Полученное значение  $d$  округляется до ближайшего большего стандартного.

8) Определяются диаметры под подшипники.

Окончательные диаметры валов должны быть согласованы с диаметрами подшипников. Из двух размеров: вал или подшипник выбирается больший.

#### 1.4.4. Расчет валов (осей) на выносливость (усталостную прочность)

После предварительных расчетов по пунктам 1 и 2 и конструктивного оформления валов (осей) фасонных конструкций, имеющих ряд ступеней, отверстий, канавок кольцевых и шпоночных и т. п. В ответственных случаях производится уточненный (проверочный) расчет валов (осей) на усталостную прочность по формуле  $s \geq [s]$ , где  $s$  и  $[s]$  — фактический и допускаемый коэффициент запаса прочности для опасного сечения.

При расчете на усталостную прочность необходимо установить характер цикла изменения напряжений. В большинстве случаев действительный цикл нагрузки машин в эксплуатационных условиях установить трудно. При расчете валов (осей) на усталостную прочность принимают, что напряжения изгиба изменяются по симметричному циклу (Рис. 1.4.7, а), а напряжения кручения — по пульсирующему циклу (Рис. 1.4.7, б).

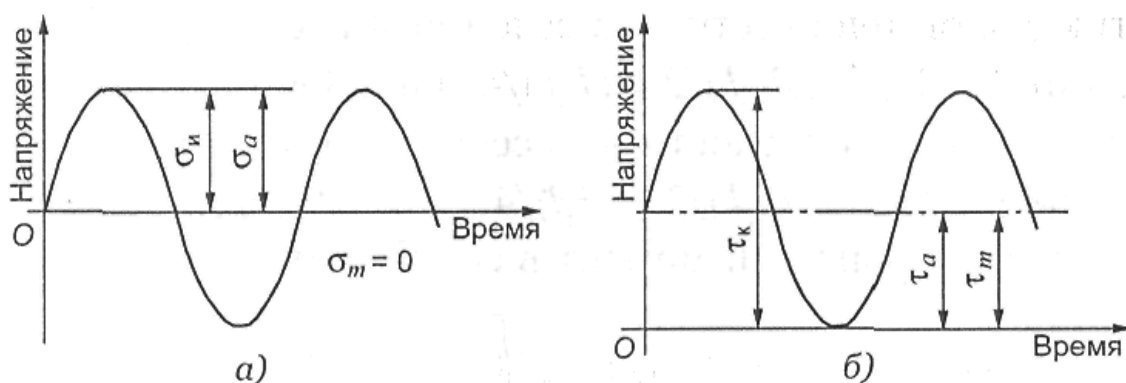


Рис. 1.4.7. Циклы изменений напряжений в сечениях вала.

Расчетный коэффициент запаса прочности определяют по формуле:

$$s = \frac{s_\sigma s_\tau}{\sqrt{s_\sigma^2 + s_\tau^2}},$$

где  $s_\sigma$  и  $s_\tau$  — коэффициенты запаса прочности при учете изгиба (или по нормальным напряжениям) и при учете кручения (или по касательным напряжениям).

Коэффициенты  $s_\sigma$  и  $s_\tau$  определяются по формулам:

$$s_\sigma = \frac{\sigma_{-1}}{(K_\sigma / K_d)\sigma_a + \Psi_\sigma \sigma_m}, \quad s_\tau = \frac{\tau_{-1}}{(K_\tau / K_d)\tau_a + \Psi_\tau \tau_m}.$$

где  $\sigma_{-1}, \tau_{-1}$  — пределы выносливости при изгибе и кручении для симметричного цикла;  $K_\sigma$  и  $K_\tau$  — эффективные коэффициенты концентрации напряжений при изгибе и кручении (Таблица 1.4.3);  $K_d$  — коэффициент влияния абсолютных размеров поперечного сечения

(Таблица 1.4.4);  $\Psi_\sigma$  и  $\Psi_\tau$  — коэффициенты чувствительности материала к асимметрии цикла.

Таблица 1.4.3

Значения коэффициентов  $K_\sigma$  и  $K_\tau$

| Концентратор напряжения                           | $K_{\sigma}$                 |             | $K_{\tau}$ |             |
|---|------------------------------|-------------|------------|-------------|
|   | Для стали с $\sigma_e$ , МПа |             |            |             |
|   | до<br>700                    | св.<br>1000 | до<br>700  | св.<br>1000 |
| Галтель:  |                              |             |            |             |
| $h/r = 1$ и $h/d = 0,02$                          | 1,49                         | 1,60        | 1,37       | 1,39        |
| $r/d = 0,05$                                      | 1,69                         | 1,83        | 1,46       | 1,51        |
| $r/d = 0,10$                                      | 1,55                         | 1,72        | 1,42       | 1,46        |
| Шпоночная канавка, выполненная<br>торцевой фрезой | 1,89                         | 2,26        | 1,71       | 2,22        |
| Прессовая посадка при $p \geq 20$ , МПа           | 2,4                          | 3,6         | 1,8        | 2,5         |
| Резьба  | 2,2                          | 2,61        | 1,0        | 1,0         |

Таблица 1.4.4

Значения коэффициента  $K_d$

| Напряженное состояние и материал | Диаметр вала $d$ , мм |      |      |      |      |
|----------------------------------|-----------------------|------|------|------|------|
|                                  | 30                    | 40   | 50   | 70   | 100  |
| Изгиб для углеродистой стали     | 0,88                  | 0,85 | 0,81 | 0,76 | 0,71 |
| Изгиб для легированной стали     | 0,77                  | 0,73 | 0,70 | 0,67 | 0,62 |
| Кручение для всех сталей         |                       |      |      |      |      |

$$\sigma_a = \sigma_i = \frac{M_i}{W_{\text{и.нетто}}}, \quad \tau_a = \tau_i = \frac{M_k}{2W_{\text{к.нетто}}},$$

где  $W_{\text{и.нетто}}$  и  $W_{\text{к.нетто}}$  — моменты сопротивления изгибу и кручению нетто сечения, определяемые по таблице 1.4.5.

Постоянные составляющие циклов нагружения для принятых условий (Рис. 1.4.7):

$$\sigma_m = 0, \quad \sigma_a = \sigma_i = \frac{M_i}{W_{\text{и.нетто}}}, \quad \tau_a = \tau_i = \frac{M_k}{2W_{\text{к.нетто}}}.$$

Значения  $\Psi_\sigma$  и  $\Psi_\tau$  выбираются по Таблице 1.3.1.

Порядок расчета валов и осей на усталостную прочность:

- После проведения расчета по пунктам 1 и 2, для каждого опасного сечения определяются расчетные коэффициенты запаса прочности и оценивается выносливость;

- При кратковременных перегрузках наиболее нагруженные сечения вала проверяются на статическую прочность (по теории энергии формоизменения):  $\sigma_{\text{пр}} = \sqrt{\sigma_{\text{и}}^2 + 3\tau_{\text{к}}^2} \leq [\sigma_{\text{из}}]$ .

Таблица 1.4.5

Формулы для расчета  $W_{\text{и.нетто}}$  и  $W_{\text{к.нетто}}$ 

| Сечение вала (оси)   | $W_{\text{и.нетто}}$                                      | $W_{\text{к.нетто}}$                                      |
|--|---|---|
| Круглое, сплошное диаметром $d$                                      | $\frac{\pi d^3}{32} \approx 0,1d^3$                       | $\frac{\pi d^3}{16} \approx 0,1d^3$                       |
| Полое, при внешнем диаметре $d$ и внутреннем $d_0$                   | $\frac{\pi d^3}{32} \left[ 1 - \frac{d_0^4}{d^4} \right]$ | $\frac{\pi d^3}{16} \left[ 1 - \frac{d_0^4}{d^4} \right]$ |
| Сплошное, с одной шпоночной канавкой ( $b$ и $t$ ) — размеры канавки | $\frac{\pi d^3}{32} - \frac{bt(d-t)^2}{d}$                | $\frac{\pi d^3}{16} - \frac{bt(d-t)^2}{d}$                |

### 1.4.5. Расчет валов на жесткость

Расчет валов на жесткость сводится к определению прогибов  $f$ , углов наклона оси вала  $\theta$ , которые не должны превышать допустимые значения.

Таблица 1.4.6

Допускаемые прогибы вала  $f$  и углы наклона  $\theta$ 

| Конструкция                                | Параметр                     |
|--|------------------------------|
| Максимальный прогиб, мм:                   |                              |
| Валов общего назначения при длине вала $l$ | $[f] = (0,0002 \div 0,003)l$ |
| В месте установления зубчатых колес        | $[f] = (0,01 \div 0,03)m^*$  |
| Валов червяков                             | $[f] = (0,005 \div 0,01)m^*$ |
| Угол наклона $\theta$ , рад:               |                              |
| радиальный шарикоподшипник                 | $\theta = 0,01$              |
| подшипник скольжения                       | $\theta = 0,001$             |
| сферический шарикоподшипник                | $\theta = 0,05$              |
| * $m$ — модуль                             |                              |

Для проверки вала на жесткость по углу закручивания при  $[\varphi] \approx (4,4 \div 8,8)10^{-3}$  рад ( $\sim 0,25 \div 0,5^\circ$ ) на 1м длины вала пользуются формулой  $d = (10,5 \div 12,5) \sqrt[4]{\frac{P}{n}}$ , где  $d$  — в см,  $P$  — в кВт,  $n$  — в мин<sup>-1</sup>.



Запасы прочности при расчёте валов и осей должны быть повышены (допускаемые напряжения снижены) в случаях:

1) если действующие нагрузки и возникающие в отдельных частях вала напряжения не могут быть точно рассчитаны (например, в случаях: а) статически неопределимого вала, опоры которого имеют осадки, не поддающиеся расчёту; б) наличия вибраций, не поддающихся расчёту; в) такой формы вала, при которой неизвестны эффективные коэффициенты концентрации напряжений и т. д.);

2) если вал (или ось) изготавливается из неоднородного материала, механические качества которого плохо известны;

3) если вал (или ось), работающий в условиях высоких температур, может быть подвергнут действию коррозии;

4) если вал (или ось) имеет большие абсолютные размеры, при которых сильнее сказываются технологические факторы и внутренние напряжения;

5) если вал (или ось) имеет ответственное значение и разрушение его может привести к тяжёлым последствиям (например; оси железнодорожного подвижного состава).

#### **1.4.6. Расчет на виброустойчивость**

Валы быстроходных машин, находящиеся под действием периодических внешних сил, могут получить значительные напряжения от возникающих в них колебаний.

Критическое число — количество оборотов вала, при котором движения валов становятся динамически неустойчивыми и в них возникают значительные поперечные колебания. Конструктивные размеры вала и масс деталей, на нём сидящих, должны выбираться такими, чтобы угловая частота собственных колебаний вала  $\omega$  отличалась от угловой скорости вращения вала  $\Omega$ .

При работе в «докритической зоне» должно осуществляться условие  $\omega \geq 1,3\Omega$ . При работе в «закритической зоне» —  $1,4\omega_I < \Omega < 0,7\omega_{II}$ , где  $\omega_I$  и  $\omega_{II}$  — угловые частоты I и II формы собственных колебаний.

При крутильных колебаниях таким же путём надо избегать состояния, при котором собственная частота колебаний совпадает с частотой действующих на вал гармоник внешних сил.

#### **Пример расчета.**

Дано:

- мощность на валу зубчатого колеса  $P = 20 \text{ кВт}$ ;
- угловая скорость вращения вала  $\omega = 152 \text{ рад/с}$ ;
- диаметр делительной окружности  $d = 210 \text{ мм}$ ;

- сила давления цепи на вал  $F = 2200\text{Н}$  (направление  $\alpha = 30^\circ$  к горизонту)
- материал вала — сталь 45 ( $\sigma_T = 290\text{ МПа}$ ,  $\sigma_{-1} = 250\text{ МПа}$ ,  $[\sigma]_и = 60\text{ МПа}$ ,  $[\tau]_к = 25\text{ МПа}$ );
- нагрузка на вал — нереверсивная постоянная.

Найти: диаметры выходного вала  $d_1, d_2, d_3, d_4$  прямозубчатого редуктора (Рис. 1.4.8).

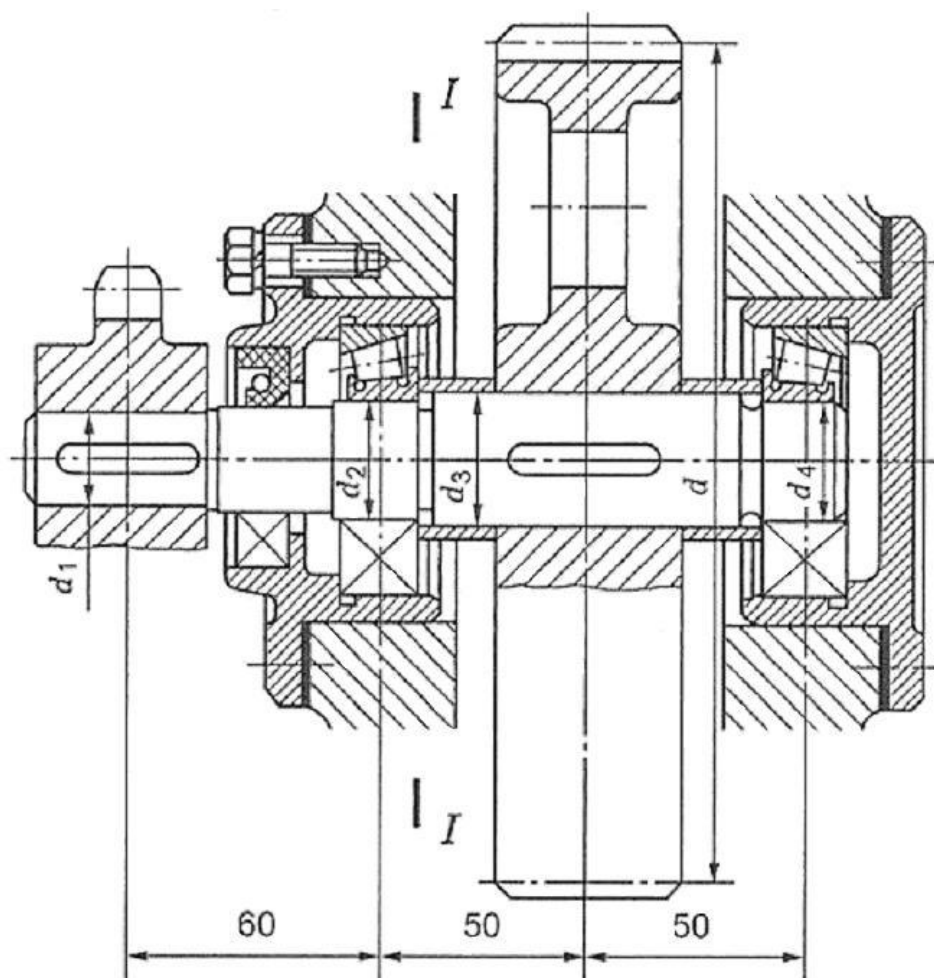


Рис. 1.4.8. Тихоходный вал редуктора.

Решение:

Составляем расчетную схему вала (Рис. 1.4.9, а) в соответствии с конструкцией редуктора (Рис. 1.4.8). Определяем силы, действующие на вал.

Окружная сила на колесе  $F_t = 2T/d \approx 1250\text{Н}$ , где  $T = 10^3 P / \omega_1$  Н·мм. Радиальная сила на колесе  $F_r = F_t \tan \alpha = 464\text{Н}$ .

Силу давления  $\mathbf{F}$  цепи на вал раскладываем на составляющие в вертикальной и горизонтальной плоскостях:  $F_v = F \sin \theta = 1100\text{Н}$ ,  $F_r = F \cos \theta = 1870\text{Н}$ .

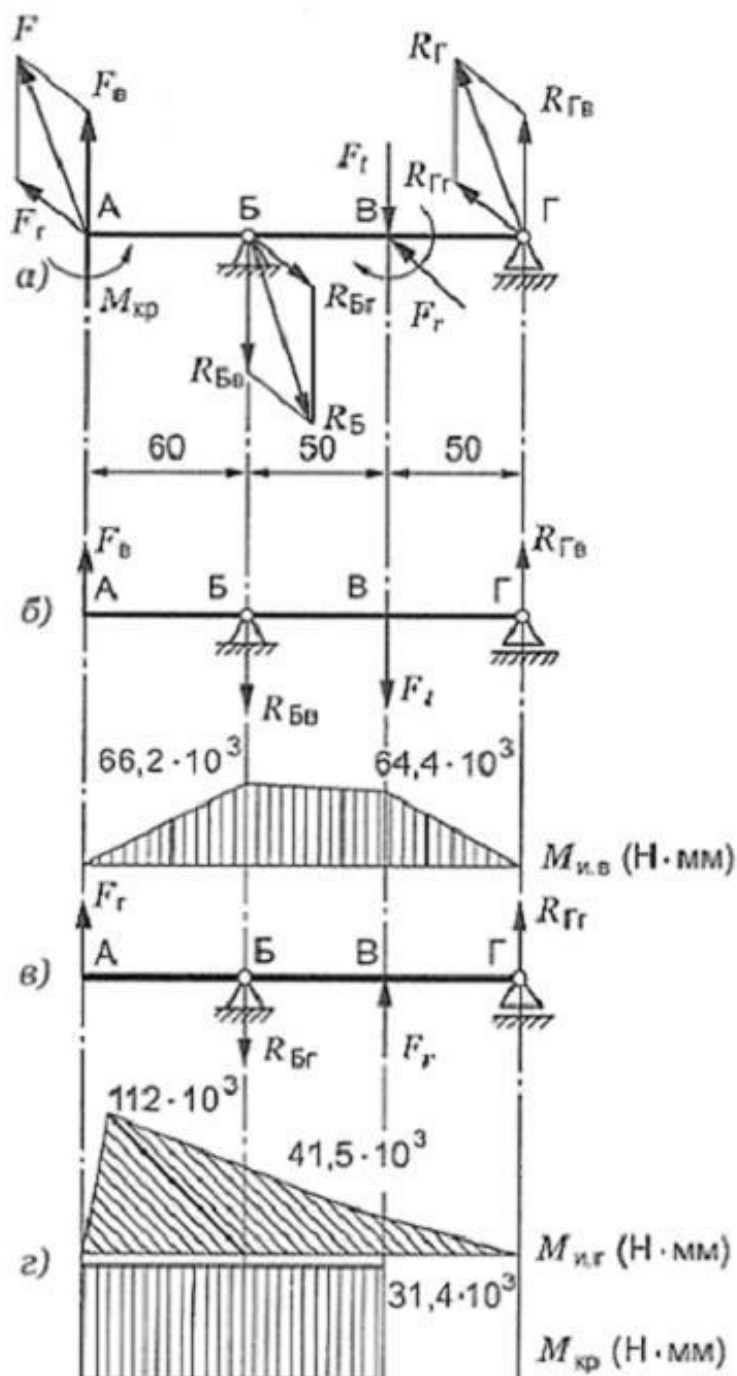


Рис. 1.4.9. Расчетная схема тихоходного вала редуктора.

Строим эпюры изгибающих моментов. В вертикальной плоскости (Рис. 1.4.9, б):

а) определяем опорные реакции:

$$\Sigma M_B = 0, -F_v \cdot 60 - F_t \cdot 50 + R_{Gv} \cdot 100 = 0, R_{Gv} = 1285 \text{ Н};$$

$$\Sigma M_G = 0, -F_v \cdot 160 + F_t \cdot 50 + R_{Bv} \cdot 100 = 0, R_{Bv} = 1135 \text{ Н};$$

б) проверяем правильность определения реакций:

$$\Sigma Y = 1100 - 1135 - 1250 + 1285 = 0 \text{ — реакции найдены правильно};$$

в) строим эпюру изгибающих моментов  $M_{и.в.}$ , для чего определяем их значения в характерных сечениях вала:

- в сечении А:  $M_{и.в.} = 0$ ;
- в сечении Б:  $M_{и.в.} = F_B \cdot 60 = 66,2 \cdot 10^3 \text{ Н} \cdot \text{мм}$ ;
- в сечении В:  $M_{и.в.} = R_{ГВ} \cdot 50 = 64,4 \cdot 10^3 \text{ Н} \cdot \text{мм}$ ;
- в сечении Г:  $M_{и.в.} = 0$ ;

В горизонтальной плоскости (Рис. 1.4.9, в):

а) определяем опорные реакции:

$$\Sigma M_B = 0, -F_G \cdot 60 + F_r \cdot 50 + R_{ГГ} \cdot 100 = 0, R_{ГГ} = 890 \text{ Н};$$

$$\Sigma M_G = 0, -F_G \cdot 160 - F_r \cdot 50 + R_{БГ} \cdot 100 = 0, R_{БГ} = 3224 \text{ Н};$$

б) проверяем правильность определения реакций:

$$\Sigma Z = 1870 - 3224 + 464 + 890 = 0 \text{ — реакции найдены правильно};$$

в) строим эпюру изгибающих моментов  $M_{и.г.}$ , для чего определяем их значения в характерных сечениях вала:

- в сечении А:  $M_{и.г.} = 0$ ;
- в сечении Б:  $M_{и.г.} = F_G \cdot 60 = 112,2 \cdot 10^3 \text{ Н} \cdot \text{мм}$ ;
- в сечении В:  $M_{и.г.} = F_{ГГ} \cdot 50 = 44,5 \cdot 10^3 \text{ Н} \cdot \text{мм}$ ;
- в сечении Г:  $M_{и.г.} = 0$ ;

г) определяем суммарный изгибающий момент в сечении Б:

$$M_{\text{сум.и1}} = \sqrt{M_{и.в.}^2 + M_{и.г.}^2} = 130 \cdot 10^3 \text{ Н} \cdot \text{мм};$$

- в сечении В:  $M_{\text{сум.и2}} = \sqrt{M_{и.в.}^2 + M_{и.г.}^2} = 78 \cdot 10^3 \text{ Н} \cdot \text{мм}$ .

Строим эпюру крутящих моментов (Рис. 1.4.9, г). Передача вращающего момента происходит вдоль вала — от середины ступицы колеса до середины ступицы звездочки (Рис. 1.4.9):  $T_K = 131,4 \cdot 10^3 \text{ Н} \cdot \text{мм}$ .

Определяем эквивалентный момент (при расчете по теории максимальных касательных напряжений):

- в сечении Б:  $M_{\text{экв1}} = \sqrt{M_{\text{сум.и1}}^2 + M_K^2} = 185 \cdot 10^3 \text{ Н} \cdot \text{мм}$ ;
- в сечении В:  $M_{\text{экв1}} = \sqrt{M_{\text{сум.и2}}^2 + M_K^2} = 153 \cdot 10^3 \text{ Н} \cdot \text{мм}$ ;

Определяем диаметры вала  $d_1, d_2, d_3, d_4$  (Рис. 1.4.8). Из расчетной схемы (Рис. 1.4.9) диаметры опасных сечений:

- в сечении А:  $d_1 = 3 \sqrt{\frac{M_K}{0,2[\tau]_K}} = 29,6 \text{ мм}$ ; принимаем  $d_1 = 30 \text{ мм}$ ;

- в сечении Б:  $d_2 = 3 \sqrt{\frac{M_{\text{экв1}}}{0,1[\sigma]_и}} = 31,7 \text{ мм}$ ; принимаем  $d_2 = 35 \text{ мм}$ , так

как подшипники качения, по стандарту, не имеют диаметра 32 мм.

Диаметр вала под правым подшипником качения для унификации принимаем  $d_4 = d_2 = 35 \text{ мм}$ .

В сечении В  $d_3 = \sqrt[3]{\frac{M_{\text{эKB2}}}{0,1[\sigma]_{\text{и}}}} = 29,4 \text{ мм}$ . Из конструктивных сообра-

жений диаметр вала под зубчатым колесом принимаем  $d_3 = 40 \text{ мм}$ .

Определим запас усталостной прочности в наиболее нагруженном сечении I-I (Рис. 1.4.8). В этом сечении концентрация напряжений обусловлена канавкой с галтелью и посадкой внутреннего кольца подшипника с натягом:

а) моменты в сечении I-I:

- суммарный изгибающий момент принимаем равным приведенному изгибающему моменту в сечении Б:  $M_{\text{сум.и1}} = 130 \cdot 10^3 \text{ Н·мм}$ ;
- крутящий момент  $M_{\text{к}} = 131,4 \cdot 10^3 \text{ Н·мм}$ ;

б) номинальные напряжения в сечении I-I:

- напряжение изгиба:  $\sigma_{\text{и}} = \frac{M_{\text{и}}}{W_{\text{и}}} = \frac{M_{\text{сум.и1}}}{0,1d_2^3} = \frac{130 \cdot 10^3}{0,1 \cdot 35^3} = 30 \text{ МПа}$ ;
- в сечении I-I продольные силы отсутствуют, поэтому нормальные напряжения в этом сечении изменяются по симметричному циклу с амплитудой  $\sigma_a = \sigma_u = 30 \text{ МПа}$ ,  $\sigma_m = 0$ ;
- напряжение кручения:  $\tau_{\text{и}} = \frac{M_{\text{к}}}{W_{\text{к}}} = \frac{M_{\text{к}}}{0,2d_2^3} = \frac{131,4 \cdot 10^3}{0,2 \cdot 35^3} = 15,2 \text{ МПа}$ .

При нулевом цикле амплитуда изменения касательных напряжений  $\tau_a = \frac{\tau_{\text{к}}}{2} = 7,6 \text{ МПа}$ .

В сечении I-I два концентратора напряжений — канавка с галтелью и посадка с натягом. Принимаем концентрацию напряжений от посадки внутреннего кольца подшипника, для которой  $K_{\sigma} = 2,4$  и  $K_{\tau} = 1,8$ ,  $d_d = 0,88$ .

Определяем коэффициенты запаса прочности вала в сечении I-I по нормальным и касательным напряжениям.

$$s_{\sigma} = \frac{\sigma_{-1}}{(K_{\sigma}/K_d)\sigma_a + \Psi_{\sigma}\sigma_m} = 3,06; \quad s_{\tau} = \frac{\tau_{-1}}{(K_{\tau}/K_d)\tau_a + \Psi_{\tau}\tau_m} = 8,5;$$

где  $\tau_{-1} = 0,58$ ,  $\sigma_{-1} = 145 \text{ МПа}$ ,  $\Psi_{\tau} = 0,05$ .

Расчетный коэффициент запаса прочности в сечении I-I:

$$s = \frac{s_{\sigma}s_{\tau}}{\sqrt{s_{\sigma}^2 + s_{\tau}^2}} = 2,68 > [s] = 2,5. \text{ Так что усталостная прочность вала в}$$

сечении I-I обеспечена.

## 1.5. Моделирование вала в системе CATIA V5

Войдите в систему и создайте новый Part (“File → New ...” затем из списка выберите Part). Присвойте ему имя Shaft (Рис. 1.5.1).

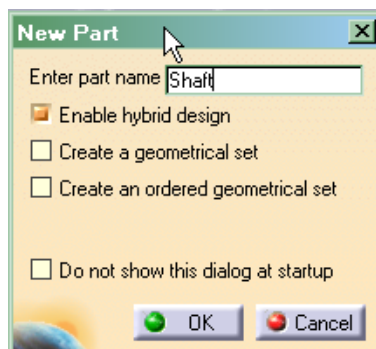







Рис. 1.5.1. Окно создания новой твердотельной модели.

Создайте эскиз на плоскости XY, с помощью команды “Sketch” . Задайте ось вращения вала, с помощью команды “Axis” . С помощью команды “Profile” , создайте половину контура сечения вала. Затем создайте фаски, используя команду “Chamfer”  и скругления с помощью команды “Corner”  (Рис. 1.5.2).

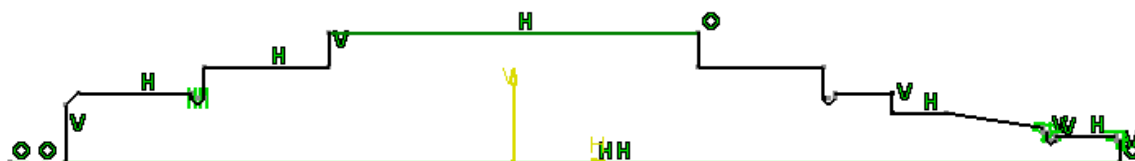



Рис. 1.5.2. Эскиз вала.

Выйдите из режима редактирования эскиза. С помощью команды “Shaft” , создайте тело вращения (Рис. 1.5.3, 1.5.4).

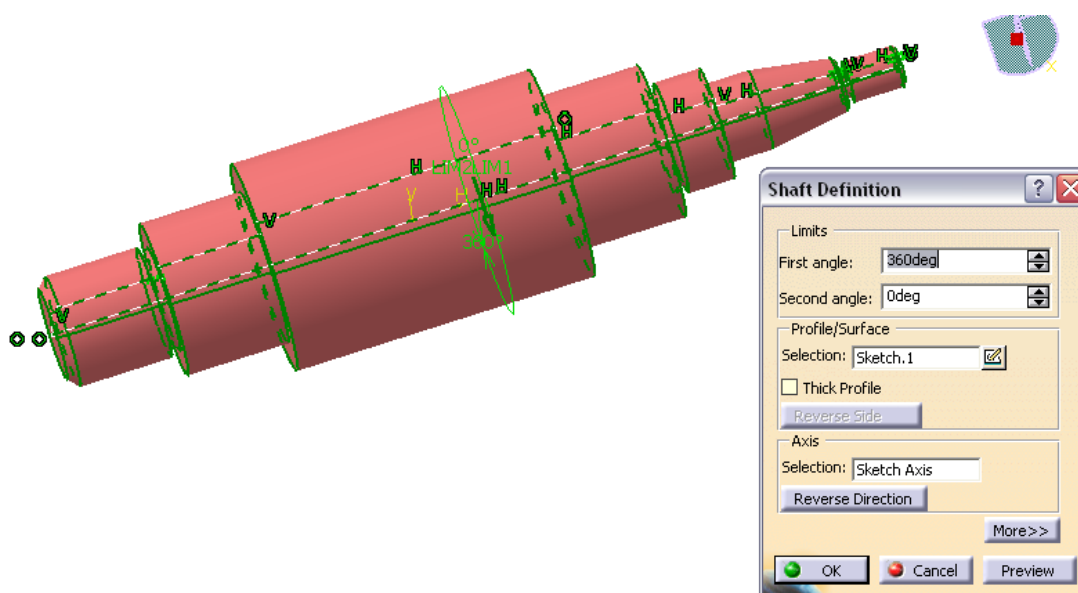


Рис. 1.5.3. Создание модели вала вращением вокруг оси.

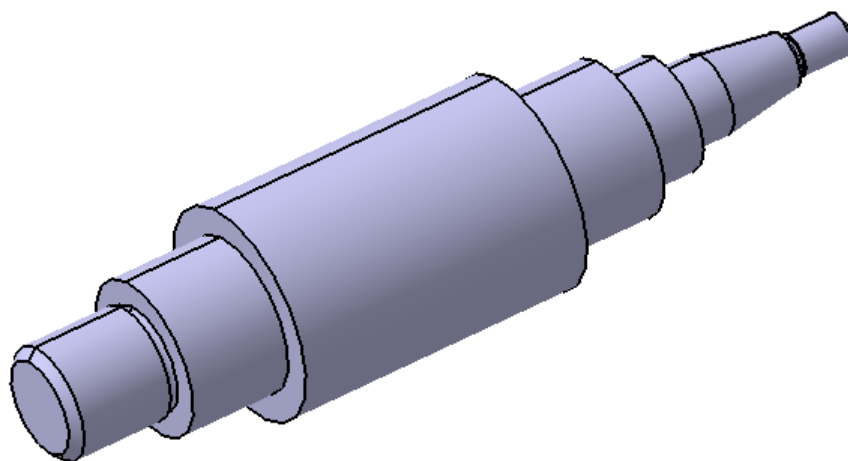


Рис. 1.5.4. Вал, смоделированный в системе CATIA V5.

## 1.6. Классификация подшипников качения

Подшипники качения — это стандартные, выпускаемые промышленностью узлы, служащие опорами валов и вращающихся осей. Подшипники качения классифицируют по следующим основным признакам (Рис. 1.6.1, 1.6.2):

- направлению (радиальному или осевому) воспринимаемой нагрузки относительно оси вала (радиальные, радиально-упорные, упорно-радиальные и упорные);
- форме тел качения (шариковые и роликовые);
- числу рядов тел качения (однорядные, двухрядные, четырехрядные и многорядные);
- способности самоустановки (самоустанавливающиеся и несамоустанавливающиеся).

Дадим здесь краткую характеристику основных типов подшипников. Конструктивно любой из подшипников качения, как узел, состоит из следующих деталей (Рис. 1.6.3): тел качения (шариков или роликов), внутреннего (надеваемого на вал) и внешнего (вставляемого в корпус, например, редуктора), колец и сепаратора, разделяющего и направляющего тела качения при их движении.

Соотношение габаритных размеров подшипников определяет их серию: сверхлегкую, особо легкую, легкую, легкую широкую, среднюю, среднюю широкую и тяжелую. Выпускаются преимущественно подшипники легкой и средней серий.

Точность подшипников качения определяется точностью их основных размеров и точностью вращения.


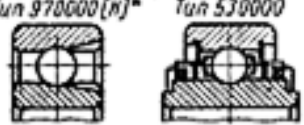



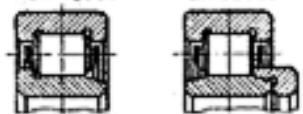

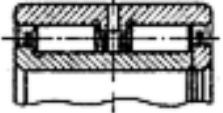
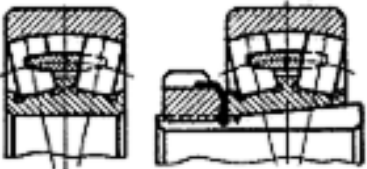
| Подшипники |                   | Радиальные   |   |
|------------|-------------------|--|---|
|            |                   | Воспринимают только радиальные нагрузки. Не обеспечивают осевую фиксацию вала  | Воспринимают радиальные и ограниченные осевые нагрузки. Обеспечивают осевую фиксацию вала   |
| Шариковые  | Несамостоятельная |  | Тип 0000    Тип 80000    Тип 50000<br><br>Тип 970000 (H)*    Тип 530000<br> |
|            |                   |  |   |
|            | Самостоятельная   |  | Тип 1000    Тип 11000<br>   |
| Роликовые  | Несамостоятельная | Тип 2000    Тип 32000    Тип 74000<br><br>Тип 4000 (H)    Тип 5000<br> | Тип 42000*    Тип 52000*<br><br>Тип 92000*    Тип 62000*<br>            |
|            |                   | Тип 3182000    Тип 784000 (H)<br><br>Конусность 1:12   | Тип 794700 (H)*<br>  |
|            | Самостоятельная   |  | Тип 3000    Тип 13000<br>  |

Рис. 1.6.1. Классификация подшипников качения (радиальные).

Основными материалами для колец и тел качения подшипников являются шарикоподшипниковые высокоуглеродистые хромистые стали ШХ15 и ШХ15СГ, широко применяются также цементуемые легированные стали 18ХГТ и 20Х2Н4А. Твердость колец и роликов обы-



чно 60...65 HRC, шариков 62...65 HRC. Сепараторы массовых подшипников изготавливают из мягкой углеродистой стали методом штамповки.

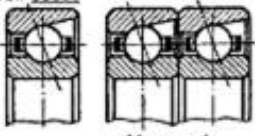
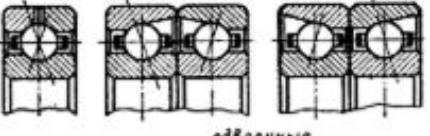
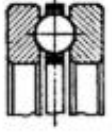




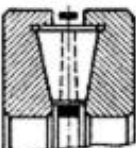

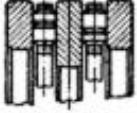
| Радиально-упорные   |   |  | Упорные   |   |
|---|---|--|---|---|
| Воспринимают как радиальную, так и осевую нагрузку одного направления               |   | Воспринимают как значительные радиальные, так и осевые нагрузки в обоих направлениях | Воспринимают только осевые нагрузки одного направления                                | Воспринимают только осевые нагрузки в обоих направлениях                              |
| Тип 36000<br>Тип 46000<br>Тип 86000   | Тип 436000<br>Тип 446000  | Тип 116000 (H)<br>Тип 336000<br>Тип 236000<br>Тип 246000                             | Тип 8000  |   |
|    |   |     |    |   |
|   |   | Тип 86000 (H)<br>Тип 56000<br>Тип 76000  |   | Тип 38000   |
|   |   |     |   |    |
|   |   |  |   |   |
| Тип 7000<br>Тип 67000   |   |  | Тип 19000   |   |
|  |  |  |  |   |
|   |   | Тип 97000<br>Тип 47000 (H)   |   | Тип 89000 (H)   |
|   |   |   |   |  |

Рис. 1.6.2. Классификация подшипников качения (упорные и радиально-упорные).

1) Шарикоподшипники радиальные однорядные.

Являются наиболее распространенными. Допускают сравнительно большую угловую скорость, особенно с сепараторами из цветных металлов или из пластмасс. Предназначены для восприятия радиальных нагрузок, могут воспринимать и значительные осевые нагрузки в двух направлениях (допустимая осевая нагрузка не более 70% от неиспользованной радиальной грузоподъемности подшипника), особенно при увеличенных внутренних зазорах.

2) Шариковые радиальные двухрядные сферические подшипники.

Предназначены воспринимать радиальные и небольшие осевые нагрузки (порядка 20% от неиспользованной радиальной). Применяются в тех случаях, когда перекося колец подшипников может составлять 2-3°).

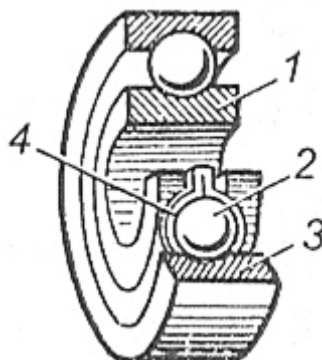


Рис. 1.6.3. Конструкция подшипника: 1 – внутреннее кольцо; 2 – тело качения; 3 – наружное кольцо; 4 – сепаратор

3) Роликоподшипники с короткими цилиндрическими роликами.

Обладают увеличенной грузоподъемностью по сравнению с аналогичными по габаритным размерам шарикоподшипниками, хорошо выдерживают ударные нагрузки. Однако они совершенно не воспринимают осевых нагрузок и не допускают перекося вала (ролики начинают работать кромками и подшипники быстро выходят из строя).

4) Шарикоподшипники радиально-упорные.

Способны воспринимать комбинированные радиально-осевые нагрузки. Их осевая грузоподъемность растет с увеличением угла контакта  $\alpha$ . Осевую нагрузку могут воспринимать только в одном направлении, поэтому для фиксации вала в обе стороны их обычно устанавливают по два подшипника на вал или по два в опору. Требуют регулировки осевых зазоров.

5) Конические роликоподшипники (тип 7000).

Рекомендуют устанавливать при средних и низких угловых скоростях (до 15 м/с). Могут воспринимать радиальные и односторонние осевые нагрузки. Для восприятия двухсторонних осевых нагрузок применяются в паре. Способны воспринимать и ударные нагрузки. Способность воспринимать осевые нагрузки растет с ростом угла конусности  $\alpha$  наружного кольца, при этом радиальная грузоподъемность уменьшается. Радиальная грузоподъемность в среднем почти в 2 раза выше, чем у радиальных однорядных подшипников. Конические роликоподшипники разъемные, что позволяет производить раздель-

ный монтаж наружных и внутренних (с комплектом роликов) колец. Требуют регулировки осевых зазоров, чувствительны к перекосам.

б) Упорные подшипники (тип 8000).

Могут воспринимать лишь осевые нагрузки: одинарные — только в одном направлении, сдвоенные — в двух направлениях. Чувствительны к перекосам. Предельные частоты вращения ограничены, поэтому при повышенных скоростях, и особенно на горизонтальных валах, их применять не следует.

## 1.7. Конструкция опор валов

Подшипники качения должны быть установлены на валу и в корпусе так, чтобы обеспечивалось необходимое радиальное и осевое фиксирование вала.

По способности фиксировать осевое положение вала опоры подразделяют на плавающие и фиксирующие. Плавающая опора (Рис. 1.7.1, а) допускает осевое перемещение вала в любом направлении. Фиксирующая опора (Рис. 1.7.1, б) фиксирует вал, т. е. предотвращает его осевое перемещение в одном направлении. Опора (Рис. 1.7.1, в) фиксирует вал в обоих направлениях.

Осевые нагрузки  $t$  могут воспринимать только фиксирующие опоры (Рис. 1.7.1 б, в).

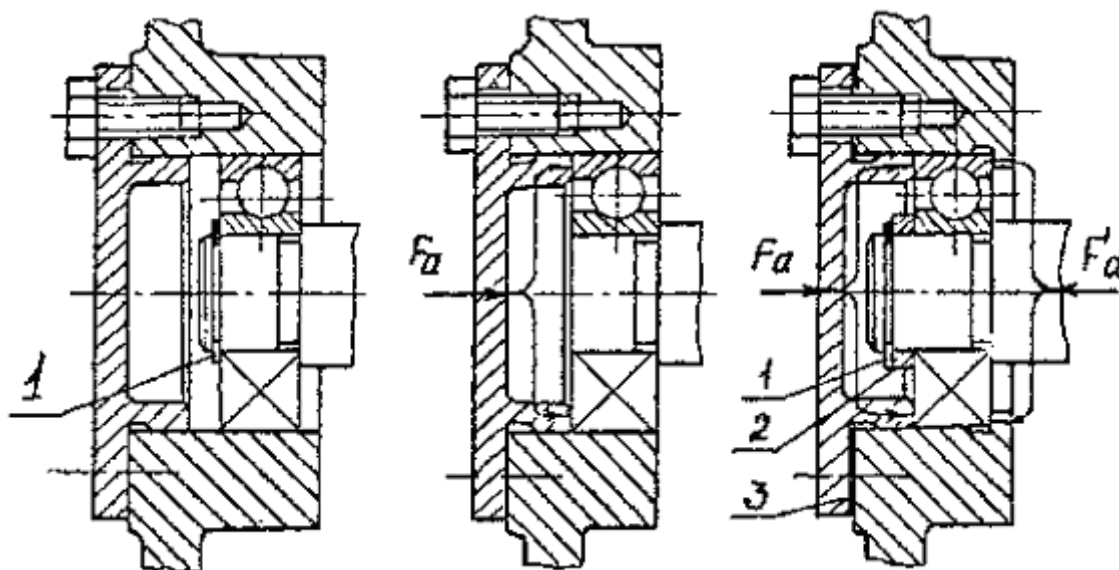


Рис. 1.7.1. Виды опор валов: а) плавающая опора, б) фиксирующая в одном направлении, в) фиксирующая в двух направлениях. 1 — плоское пружинное кольцо; 2 — кольцо-компенсатор; 3 — набор прокладок.

$F_a$  и  $F'_a$  — варианты осевой нагрузки, которую способна воспринимать фиксирующая опора.

При выборе типа подшипников для плавающих и фиксирующих опор следует учитывать их конструктивные особенности. Так, в плавающих опорах применяются радиальные шариковые или роликовые (Рис. 1.6.1) подшипники.

В зависимости от конструкции подшипниковых узлов возможны различные сочетания плавающих и фиксирующих опор, т. е. различные схемы установки валов.

Схема 1. Обе опоры плавающие (Рис. 1.7.2).

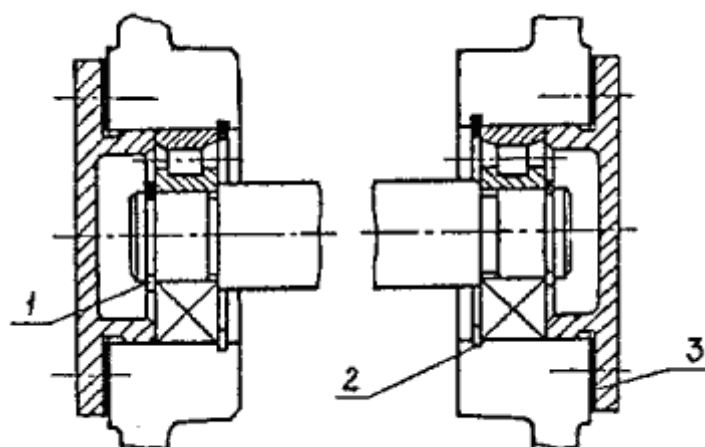


Рис. 1.7.2. Установка вала на двух плавающих опорах. 2 — плоское пружинное кольцо; 3 — набор прокладок.

Данная схема применяется в тех случаях, когда осевая фиксация вала осуществляется какими-либо другими элементами конструкции. Например, в шевронных зубчатых передачах опоры одного из валов должны обеспечивать его самоустановку относительно другого вала, зафиксированного от осевых перемещений.

Это вызвано тем, что при изготовлении шевронных зубчатых колес неизбежны погрешности углового положения зуба одного полушеврона относительно зуба другого полушеврона. По этой причине первоначально в зацепление входят зубья только одного полушеврона. Возникающая при этом осевая сила смещает колесо вместе с валом вдоль оси плавающих опор до того положения, когда в зацепление войдут зубья обоих полушевронов. Осевая фиксация вала осуществляется не опорами, а шевронными зубьями колеса. Здесь лишняя связь в виде фиксирующей опоры привела бы к статической неопределимости системы.

Схема 2. Одна из опор фиксирующая, вторая — плавающая.

Фиксирующая опора фиксирует вал в обоих направлениях, и может содержать как один (Рис. 1.7.3), так и два подшипника. В качестве плавающей целесообразно выполнять опору, нагруженную меньшей радиальной нагрузкой.

В таком виде система является статически определимой и может быть представлена в виде балки с одной шарнирно-неподвижной и одной шарнирно-подвижной опорами.

Осевые перемещения вала в фиксирующей опоре зависят от собственного осевого зазора в подшипниках, способов крепления колец подшипников на валу и в корпусе, а также от собственной осевой жесткости подшипников.

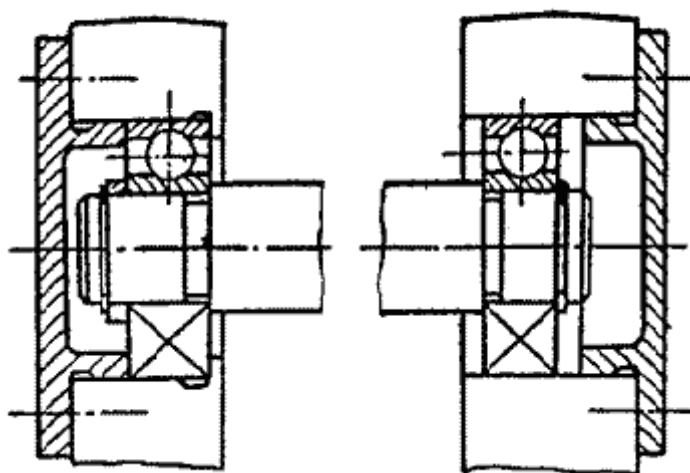


Рис. 1.7.3. Установка вала с одной плавающей и одной фиксирующей опорой.

Основные достоинства этой схемы:

- не требуется точно выдерживать расстояние между опорами, что особенно важно при расположении опор в отдельных корпусах;
- опоры могут быть расположены на любом расстоянии друг от друга, так как температурные или упругие деформации валов будут компенсироваться осевыми перемещениями плавающей опоры;
- высокая осевая жесткость фиксирующей опоры, особенно в случае применения двух роликовых конических или двойных упорных подшипников.

Недостатком этой схемы является сложная конструкция крепления подшипников на валу и в корпусе.

Схема 3. Обе опоры фиксирующие.

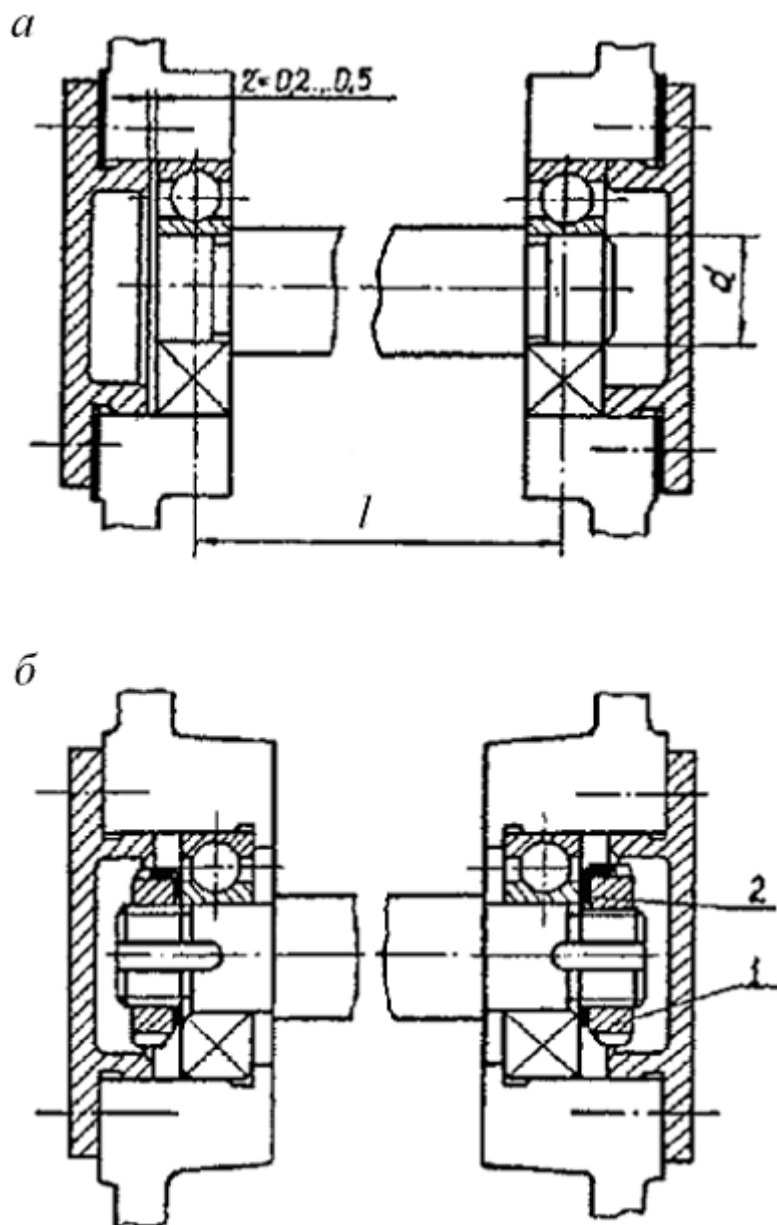


Рис. 1.7.4. Установка вала на двух фиксирующих опорах. а) схема «враспор»; б) схема «врастяжку». 1 — круглая шлицевая гайка; 2 — многолапчатая шайба.

Каждая из опор фиксирует вал в одном направлении. Различаются схемы «враспор» (Рис. 1.7.4, а) и «врастяжку» (Рис. 1.7.4, б).

Схема установки вала «враспор» конструктивно является наиболее простой. Она широко применяется при малых расстояниях между опорами. При больших расстояниях между опорами (для радиальных шарикоподшипников  $l/d = 8...10$ ) для исключения защемления вала из-за его температурных деформаций необходимо предусматривать зазор  $Z$  между крышкой и наружным кольцом подшипника.

При установке вала «врастяжку» расстояние между подшипниками можно увеличить (для радиальных шарикоподшипников  $l/d = 10...12$ ), так как температурные деформации вала ведут к увеличению зазоров в подшипниках.

## 1.8. Крепление подшипников на валу

На рисунке 1.8.1 приведены способы крепления подшипников на валу, которые применяются при нагружении вала значительной осевой силой в обоих направлениях.

Надежное крепление подшипника осуществляется круглой шлицевой гайкой (Рис. 1.8.1, а), которая от самопроизвольного откручивания стопорится многолапчатой шайбой. Стопорная шайба имеет один внутренний выступ и шесть наружных выступов-лапок. Внутренний выступ шайбы заходит в специально выполненный паз на валу, а один из ее наружных выступов отгибается в шлиц гайки. Просто и надежно крепление концевой шайбой (Рис. 1.8.1, б). В этом случае штифт фиксирует шайбу от поворота относительно вала. Чтобы концевые шайбы при высоких частотах вращения не вызывали дисбаланса, они центрируются по отверстию подшипника (Рис. 1.8.1, в) или по валу (Рис. 1.8.1, г).

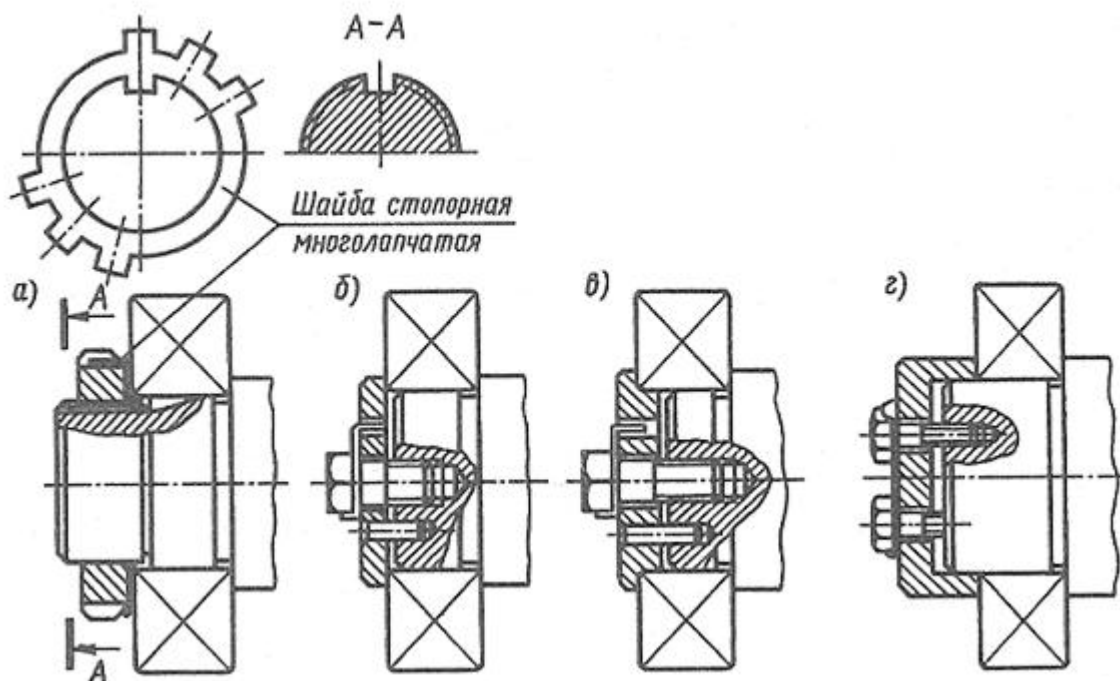


Рис. 1.8.1. Крепление подшипников на валах.

Во всех вариантах необходимо предусматривать стопорение винтов, крепящих шайбу к торцу вала, от самоотвинчивания. На Рис. 1.8.1, б), в стопорение винта осуществляется шайбой стопорной с нос-

ком, а на Рис. 1.8.1, г) — деформируемой шайбой, установленной под оба винта сразу. Концы шайбы отгибаются на грани головок винтов.

Все большее применение находит крепление подшипников пружинным упорным плоским кольцом (Рис. 1.8.2, в).

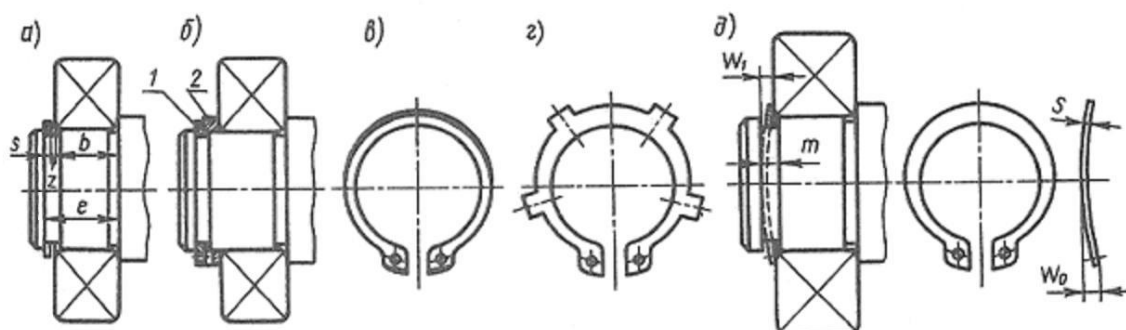


Рис. 1.8.2. Крепление подшипников на валах.

Наличие зазора между упорным кольцом и подшипником (Рис. 1.8.2, а)  $z = e - s - b$ , является недостатком данного крепления. Для его устранения целесообразно между подшипником и пружинным упорным кольцом 1 ставить компенсаторное кольцо 2 (Рис. 1.8.2, б).

Особенностью конструкции подшипника качения является то, что его внутреннее кольцо является весьма податливой деталью. Чтобы внутреннее кольцо было установлено на валу точно, без перекоса, его необходимо поджимать при сборке к заплечику вала или к торцу детали, установленной на валу. Кольцо подшипника должно прилегать к упорному заплечику своей плоской торцевой поверхностью. С одной стороны высота заплечика вала должна быть больше координаты фаски подшипника, с другой — должна быть выбрана с учетом возможности снятия подшипника с вала.

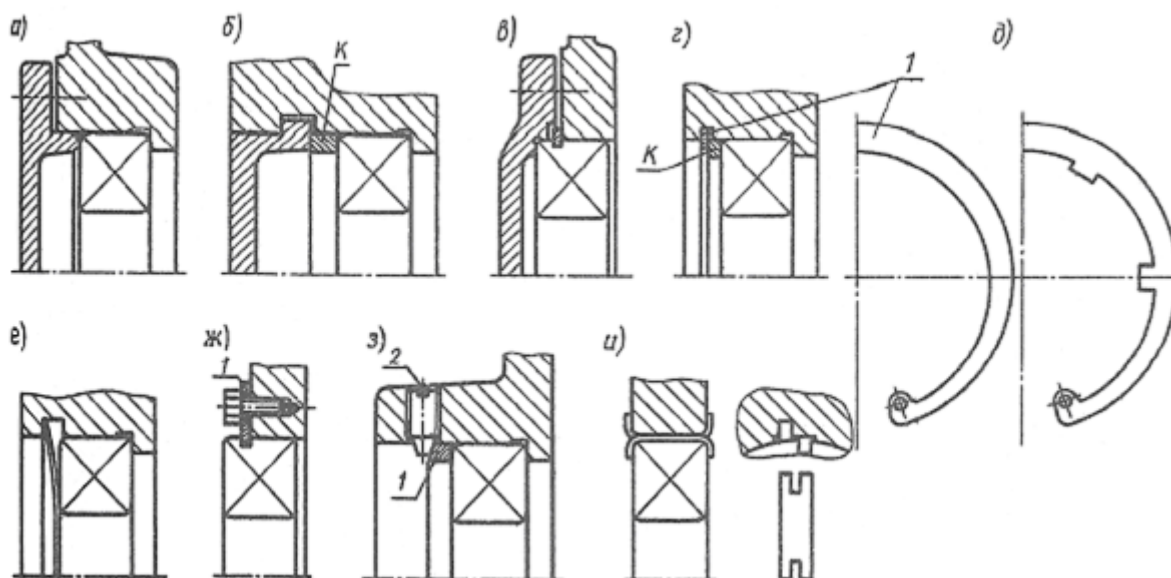


Рис. 1.8.3. Крепление подшипников на валах.



Если же не удастся создать заплечик вала требуемой высоты, то используются варианты:

а) между заплечиком вала и кольцом подшипника ставится промежуточное кольцо необходимой высоты (Рис. 1.8.3, а);

б) создают заплечик установкой пружинного упорного плоского кольца в канавку вала (Рис. 1.8.3, б);

в) устанавливают дополнительное кольцо 1, улучшающее контакт подшипника с пружинным кольцом (Рис. 1.8.3, в);

г) в канавку на валу устанавливают два полукольца Г-образного или прямоугольного сечения, которое от выпадения удерживает внутреннее кольцо подшипника (Рис. 1.8.3, г), пружинное кольцо (Рис. 1.8.3, д), неразъемное кольцо 2 (Рис. 1.8.3, е, ж).

## 1.9. Методика расчета осей в ANSYS

### 1.9.1. Расчет на чистый изгиб (для оси)

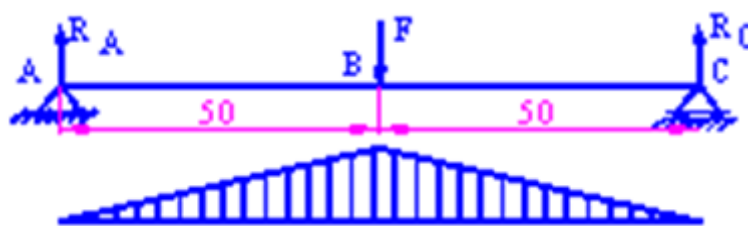


Рис. 1.9.1. Расчетная схема.

Дано:  $F = 1000\text{Н}$ ,  $l = 100\text{мм}$ , материал сталь 45 ( $[\sigma]_и = 60\text{МПа}$ ).

Найти: диаметр вала  $d$ .

Определяем:

- реакции опор  $R_A = R_C = 500\text{Н}$ ;
- изгибающий момент в сечениях A и C:  $M_и = 0$ ;
- изгибающий момент в сечении B:  $M_и = R_A \cdot l/2 = R_C \cdot l/2 = 25000\text{Н}\cdot\text{мм}$ .

Искомый диаметр  $d = 3\sqrt{\frac{M_и}{0,1[\sigma]_и}} = 12,7\text{ мм}$  в соответствии со стан-

дартным рядом принимаем  $d = 14\text{мм}$ .

### 1.9.2. Построение модели

1) Определение конечного элемента и задание констант.

При расчете вал принимается за балку, лежащую на шарнирных опорах.

Для задания элемента нужно выбрать: “Main Menu → Preprocessor → Element Type → Add/Edit/Delete”

В появившемся окне выбираем в левой колонке “Structural Beam”, в правой — “2D elastic 3”. Нажимаем “Ok”. Выбран элемент “BEAM3” (его свойства можно посмотреть в библиотеке конечных элементов). Закрываем окно выбора элемента.

Чтобы задать константы выбираем “Main Menu → Preprocessor → Real Constants → Add/Edit/Delete”.

Нажимаем “Add” затем “Ok”. Появляется список параметров элемента. Заполняем поля как показано на Рис. 1.9.2.

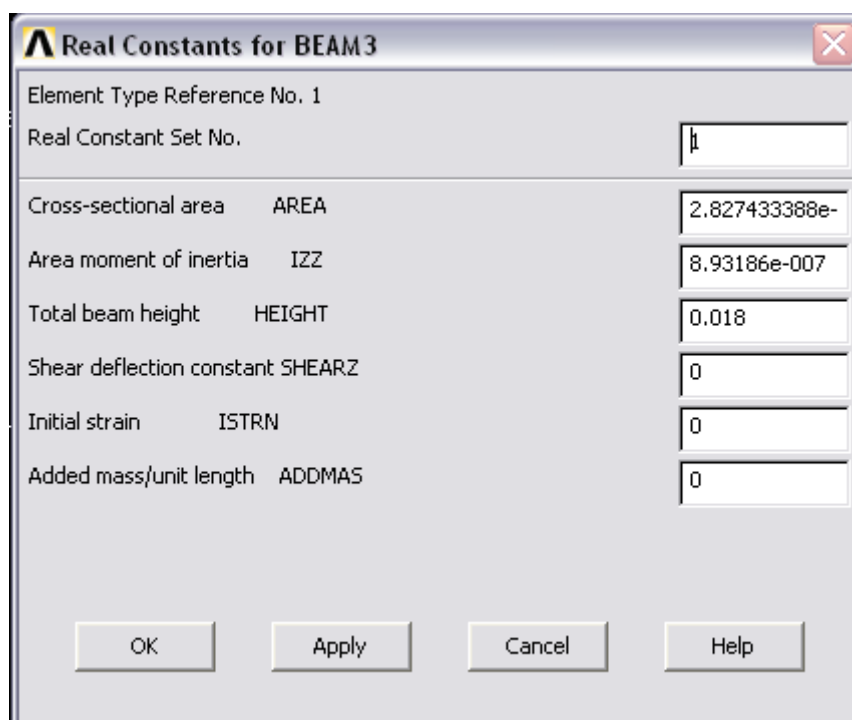


Рис. 1.9.2. Задание параметров конечного элемента.

## 2) Задание свойств материала.

Выбираем “Main Menu → Preprocessor → Material Props → Material Models”, в появившемся окне выбираем двойным кликом “Structural”, “Linear”, “Elastic”, “Isotropic” и вводим значения как показано на Рис. 1.9.3.

Нажимаем “Ok”. Затем в поле “Density” вводим значение “7800” и выходим из меню задания свойств материала.

## 3) Построение геометрии модели.

Нажимаем “Main Menu → Preprocessor → Modeling → Create → Nodes → In Active CS” в появившемся окне вводим “1” в “node number” и нажимаем “Apply”. По умолчанию координаты узла будут: 0,0,0. Далее вводим “node number” = “11” и координаты, представленные на Рис. 1.9.4.

Для создания остальных узлов нажимаем “Main Menu → Preprocessor → Modeling → Create → Nodes → Fill between Nds” выбираем

мышкой узел “1” и “11”, нажимаем “Ok”. В появившемся окне нажимаем “Ok” не изменяя ничего (Рис. 1.9.5).

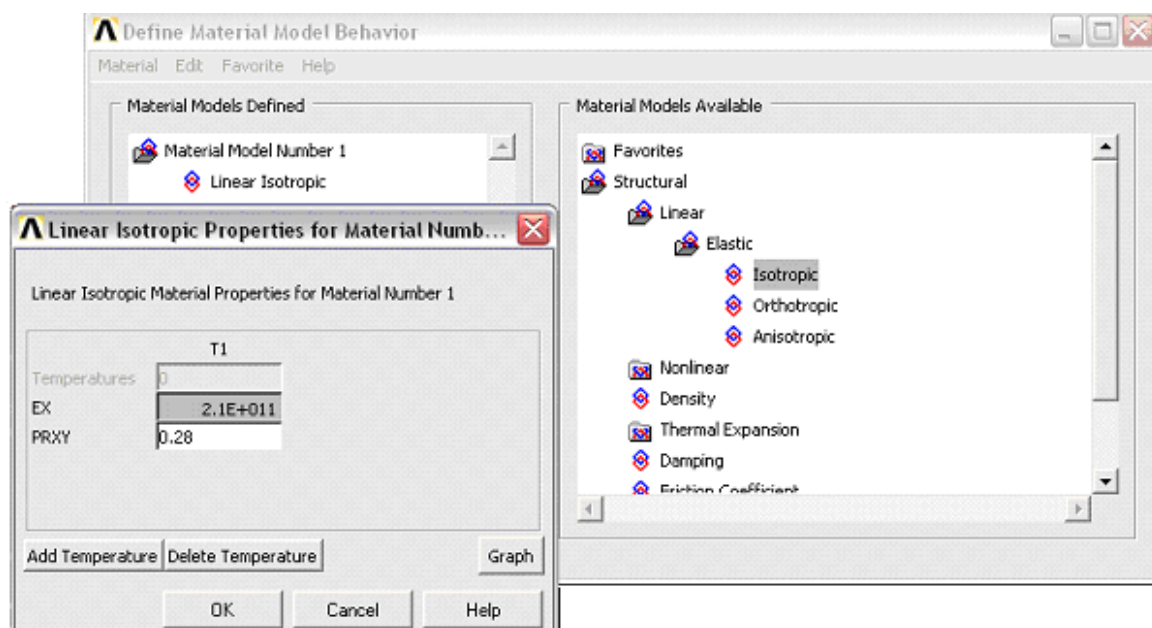


Рис. 1.9.3. Задание свойств материала.

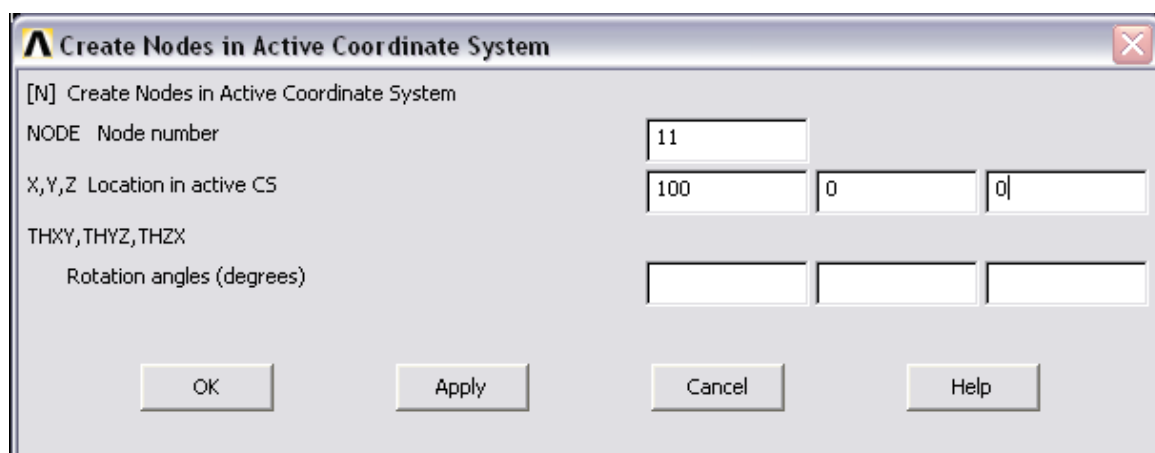


Рис. 1.9.4. Задание узла модели через координаты.

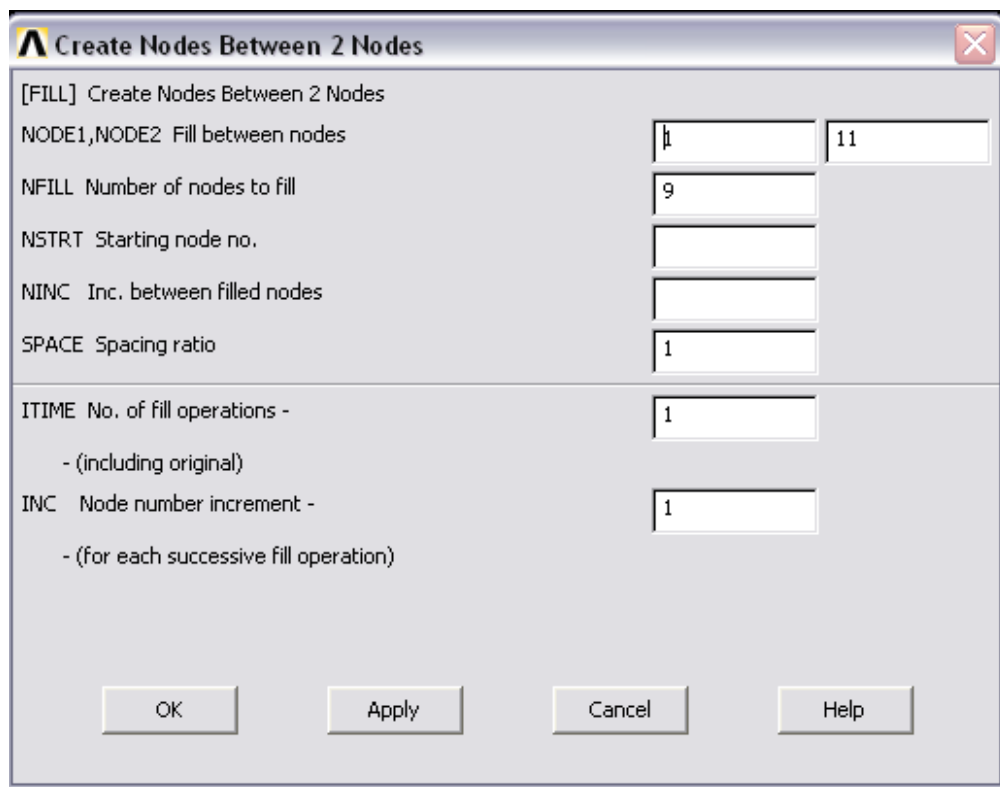
Далее на построенных узлах создаем элементы: “Main Menu → Preprocessor → Modeling → Create → Elements → Auto Numbered → Thru Nodes”, щелкаем мышкой по узлам “1” и “2” и нажимаем “Ok”. Далее копируем созданный элемент на остальных узлах “Main Menu → Preprocessor → Modeling → Copy → Elements → Auto Numbered” и выбираем “Pick All” в появившемся окне вводим “total number of copies” (Рис. 1.9.6, 1.9.7).

#### 4) Задание опор и силы:

Нажимаем “Main Menu → Solution → Define Loads → Apply → Structural → Displacement → On Nodes”, выбираем мышкой узлы “1” и

“11” и нажимаем “Ok”. В появившемся окне выделяем “All DOF” и нажимаем “Ok” (Рис. 1.9.8).

Далее выполняем команду “Main Menu → Solution → Define Loads → Apply → Structural → Force/Moment → On Nodes” (Рис. 1.9.9, 1.9.10).



1 2 3 4 5 6 7 8 9 10 11

Рис. 1.9.5. Создание 9 узлов между 1 и 11.

### 1.9.3. Проведение расчета

Выполняем команду “Main Menu → Solution → New Analysis” и выбираем “Static” анализ.

Выполняем команду “Main Menu → Solution → Solve → Current LS” и ждем пока появится окно “Solution is done”.

Выполняем команду “Main Menu → General Postproc → Read Results → First Set”, затем команду “Main Menu → General Postproc → Plot Results → Deformed Shape” выбираем “Def + undeformed” и нажимаем “Ok”. На Рис. 1.9.11 представлен результат.

Далее по созданной модели можно получить сведения о моментах и относительных перемещениях. Рассчитанный аналитически момент в сечении *B* совпал с моментом в узле 6:  $M_{\text{и}} = 25000 \text{ Н}\cdot\text{мм}$ .

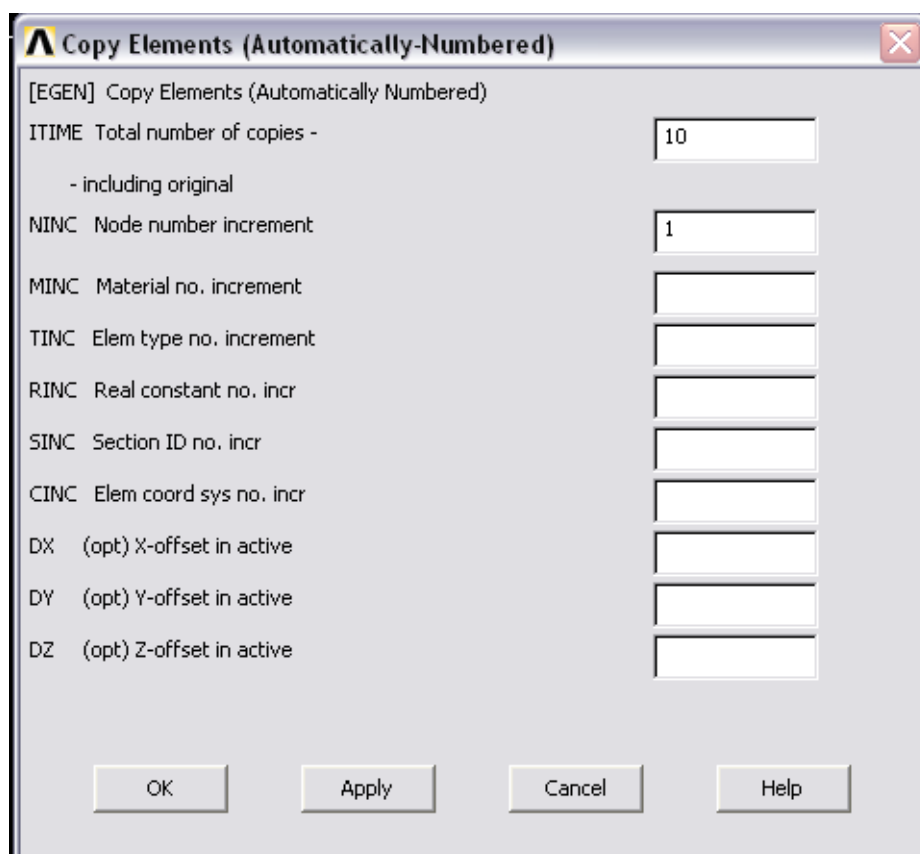


Рис. 1.9.6. Создание элементов модели.

1 2 3 4 5 6 7 8 9 10 11

Рис. 1.9.7. Модель оси.

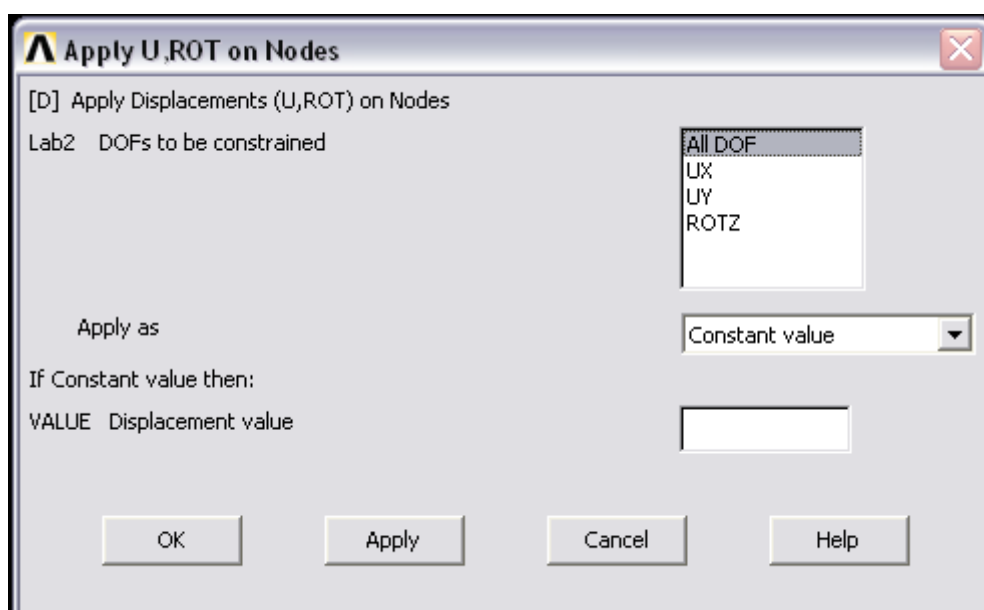


Рис. 1.9.8. Задание закрепления в узлах по всем направлениям.

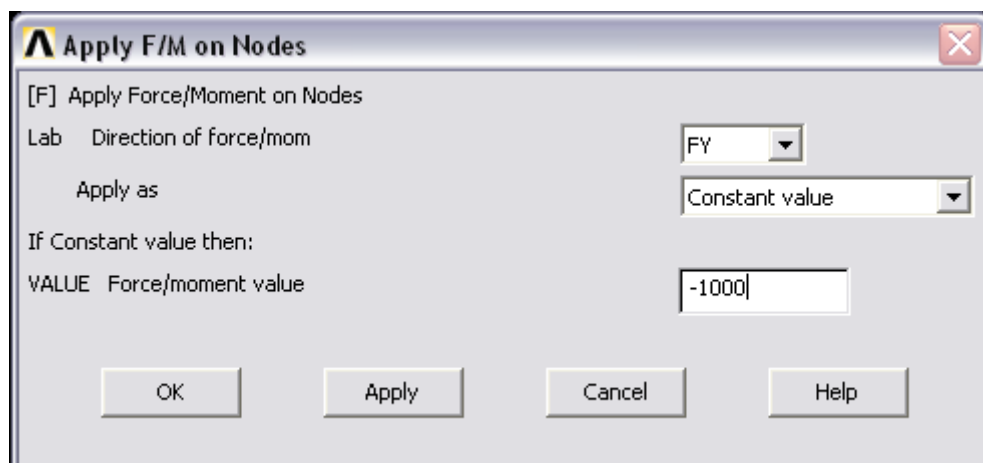


Рис. 1.9.9. Задание воздействия.



Рис. 1.9.10. Модель оси для расчета на изгиб.

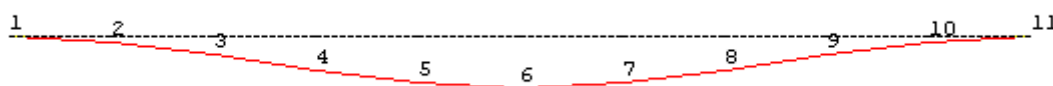


Рис. 1.9.11. Прогиб оси.

## **Глава 2. Разработка параметрической модели зубчатой передачи**

Появление систем компьютерного проектирования позволило значительно упростить разработку новых изделий. Современные системы обладают возможностью создания трехмерных моделей деталей, их сборки. Но одним из важнейших свойств, учитываемых при выборе CAD-системы, является возможность создания параметрических моделей. Параметризация упрощает изменения проектируемого изделия и дает возможность использовать уже существующие модели с новыми параметрами.

Зубчатая передача — это механизм, который с помощью зубчатого зацепления передает или преобразует движение с изменением угловых скоростей и моментов.

Зубчатые передачи применяют для преобразования и передачи вращательного движения между валами с параллельными, пересекающимися и перекрещивающимися осями, а также для преобразования вращательного движения в поступательное, и наоборот.

Зубчатые передачи составляют наиболее распространенную и важную группу механических передач. Их применяют в широком диапазоне областей и условий работы.

Основные преимущества зубчатых передач: высокая нагрузочная способность; большая долговечность и надежность работы (например, для редукторов общего применения установлен ресурс ~30 000 ч); высокий КПД (до 0,97...0,98 в одной ступени); постоянство передаточного отношения (отсутствие проскальзывания); возможность применения в широком диапазоне скоростей (до 150 м/с), мощностей (до десятков тысяч кВт) и передаточных отношений (до нескольких сотен и даже тысяч).

К недостаткам зубчатых передач могут быть отнесены требования высокой точности изготовления, шум при работе со значительными скоростями, высокую жесткость, не позволяющую компенсировать динамические нагрузки. Отмеченные недостатки не снижают существенного преимущества зубчатых передач перед другими.

Принцип действия зубчатой передачи основан на зацеплении пары зубчатых колес.

По расположению осей валов различаются: передачи с параллельными осями, которые выполняются с цилиндрическими колесами внешнего или внутреннего зацепления; передачи с пересекающи-

мися осями — конические колеса; передачи с пересекающимися осями — цилиндрические винтовые, конические гипоидные, червячные. Кроме того, применяют передачи между зубчатым колесом и рейкой.

По расположению зубьев на колесах различают передачи: прямозубые и косозубые. По форме профиля зубов различают эвольвентные и круговые передачи.

## 2.1. Параметры и конструкции зубчатых передач

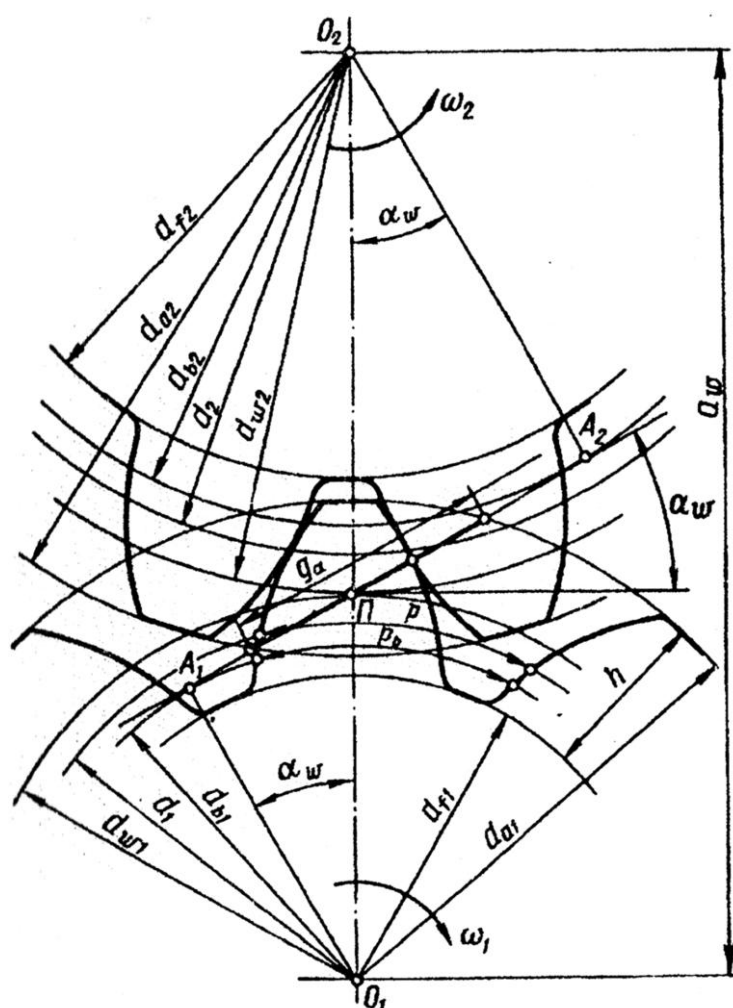


Рис. 2.1.1. Геометрические параметры зубчатого зацепления.

Основным кинематическим условием, которому должны удовлетворять профили зубьев, является постоянство мгновенного передаточного отношения передачи. Этому условию удовлетворяют многие классы кривых. Для обеспечения высокого КПД, прочности и долговечности колес профили должны обеспечивать малые скорости скольжения и достаточные радиусы кривизны в точках контактах.



Профили должны допускать легкое изготовление, в частности нарезание простым инструментом независимо от числа зубьев колес.

Меньшее из пары зубчатых колес называют шестерней, а большее — колесом. Параметрам шестерни приписывают индекс 1, а параметрам колеса — 2 (Рис. 2.1.1). Кроме того, различают индексы, относящиеся:  $w$  — к начальной поверхности или окружности;  $b$  — к основной поверхности или окружности;  $a$  — к поверхности или окружности вершин и головок зубьев;  $f$  — к поверхности или окружности впадин и ножек зубьев. Параметрам, относящимся к делительной поверхности или окружности, дополнительного индекса не приписывают.

Общие понятия о параметрах пары зубчатых колес и их взаимосвязи проще всего уяснить, рассматривая прямозубые колеса. При этом особенности косозубых колес рассматривают дополнительно:

- передаточное число  $u = z_2/z_1$ ;
- $z_1$  и  $z_2$  — число зубьев шестерни и колеса;
- $p$  — делительный окружной шаг;
- $p_b = p \cos \alpha$  — основной окружной шаг зубьев;
- $\alpha$  — угол профиля делительный (равный углу профиля исходного контура), по ГОСТ 13755 – 81,  $\alpha = 20^\circ$ ;
- $\alpha_w$  — угол зацепления или угол профиля начальный:  
 $\cos \alpha_w = a \cos \alpha / a_w$ ;
- $m = p/\pi$  — окружной модуль зубьев (основная характеристика размеров зубьев); значения модулей стандартизованы СТ СЭВ 310 – 76 в диапазоне 0,05... 100мм (Табл. 2.1.1);

Таблица 2.1.1

Стандартные значения модуля

| Ряды | Модуль, мм  |
|------|---|
| 1-й  | 1; 1,25; 1,5; 2; 2,5; 3; 4; 5; 6; 8; 10; 12; 16; 20; 25               |
| 2-й  | 1,125; 1,375; 1,75; 2,25; 2,75; 3,5; 4,5; 5,5; 5,7; 9; 11; 14; 18; 22 |

- $d = pz/\pi = mz$  — делительный диаметр (диаметр окружности, по которой обкатывается инструмент при нарезании);
- $d_b = d \cos \alpha$  — основной диаметр (диаметр окружности, разверткой которой являются эвольвенты зубьев);
- $d_{w1}$  и  $d_{w2}$  — начальные диаметры (диаметры окружностей, по которым пара зубчатых колес обкатывается в процессе вращения)

У передач без смещения и при суммарном смещении  $X_\Sigma = 0$  начальные и делительные окружности совпадают:

$d_{w1} = 2a_w / (z_2/z_1 + 1)$ ,  $d_{w2} = 2a_w - d_{w1}$ , где  $a_w = 0,5(d_{w1} + d_{w2})$  — межосевое расстояние.

Для передач без смещения при  $X_\Sigma = 0$ ,  $a_w = a = 0,5m(z_1 + z_2)$ ,

- $h = m(h_a^* + c^*)$  — высота зуба;
- $d_a = d + 2mh_a^*$  — диаметр вершин зубьев;
- $d_f = d - 2m(h_a^* + c^*)$  — диаметр впадин;
- $h_a^*$  — коэффициент высоты головки зуба (по ГОСТ 13755 – 81,  $h_a^* = 1$ );
- $c^*$  — коэффициент радиального зазора (по ГОСТ 13755 – 81,  $c^* = 0,25$ );
- $h = 2,25m$ ;
- $A_1A_2$  — линия зацепления (общая касательная к основным окружностям);

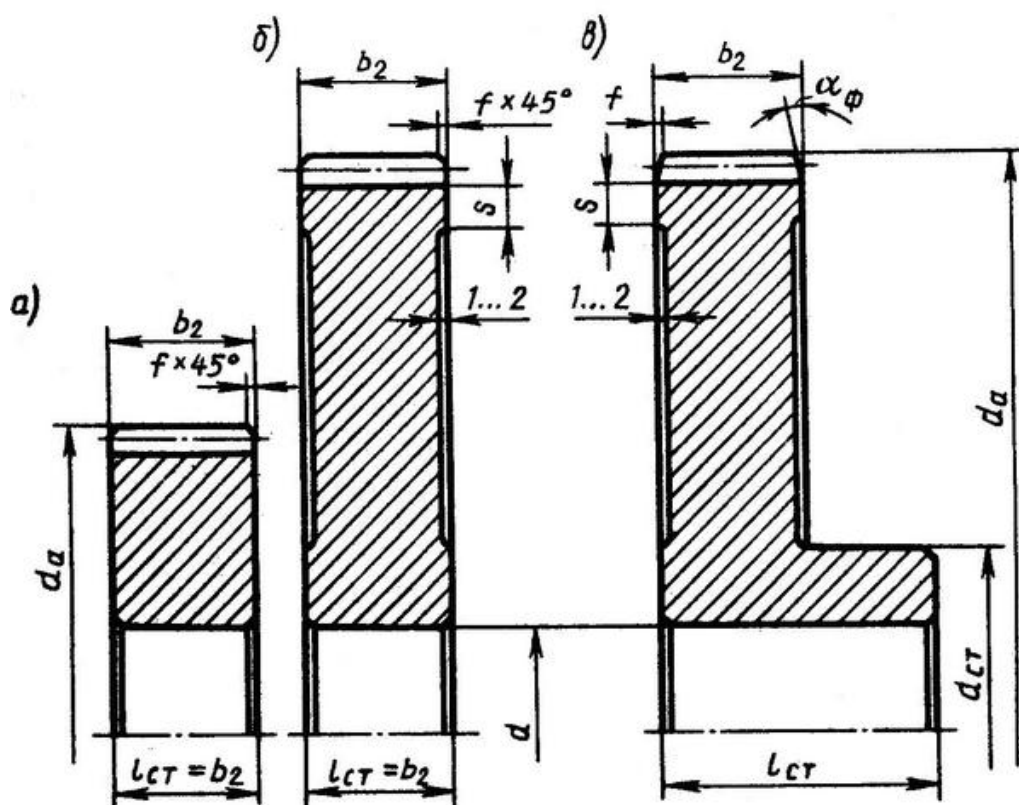


Рис. 2.1.2. Формы зубчатых колес.

- $g_\alpha$  — длина активной линии зацепления (отсекаемая окружностями вершин зубьев);
- $P$  — полюс зацепления (точка касания начальных окружностей и одновременно точка пересечения линии центров колес  $O_1O_2$  с линией зацепления).

Форма зубчатого колеса может быть плоской (Рис. 2.1.2, а, б) или с выступающей ступицей (Рис. 2.1.2, в). Значительно реже (в одноступенчатой передаче) — с полумуфтой (Рис. 2.1.2, г).

пенчатых редукторах) колеса делаются со ступицей, выступающей в обе стороны.

В зависимости от получения заготовки зубчатые колеса подразделяются на литые (Рис. 2.1.3, а), кованные или штампованные, изготовленные механической обработкой (Рис. 2.1.3, б), сварные (Рис. 2.1.3, в).

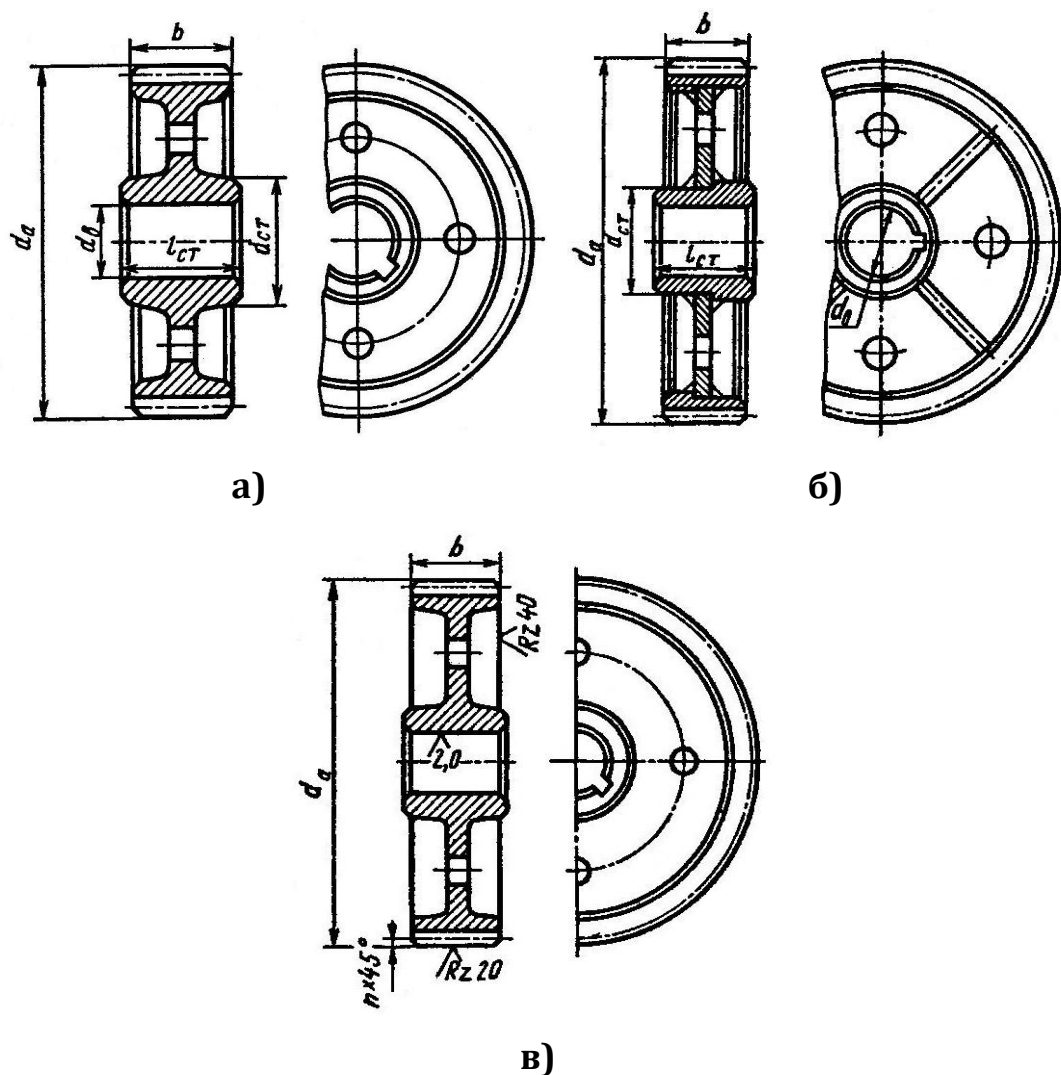


Рис. 2.1.3. Зубчатые колеса.

Зубчатые колеса, у которых диаметр впадин незначительно превышает диаметр вала в месте посадки зубчатого колеса, изготавливаются как одно целое с валом. Такую конструкцию (Рис. 2.1.4) называют валом-шестерней. В остальных случаях зубчатое колесо выполняется отдельно, после чего насаживается на вал.

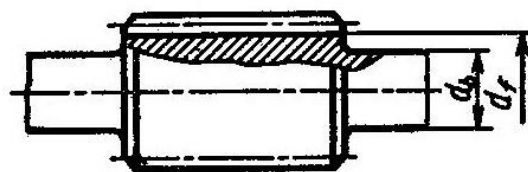


Рис. 2.1.4. Вал-шестерня.

Колеса диаметром меньше 400мм имеют форму диска с выточками или без выточек. Чаще всего эти колеса изготавливают из поковок. Колеса диаметром более 400–500мм изготавливают со спицами (Рис. 2.1.5) различного сечения.

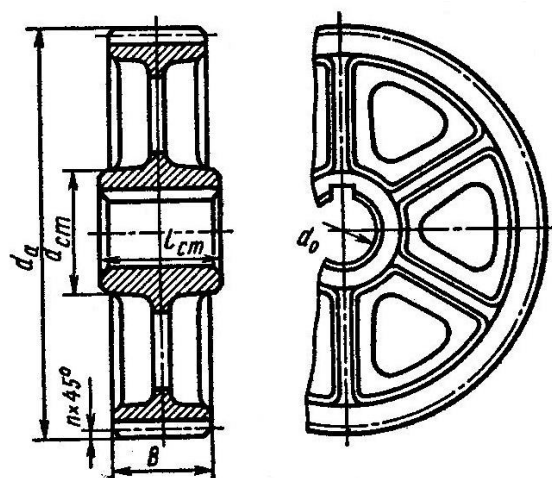


Рис. 2.1.5. Зубчатое колесо со спицами.

Для уменьшения массы зубчатых колес толщина диска принимается значительно меньшей ширины обода. Кроме того, в дисках между ободом и ступицей предусматриваются круглые отверстия для удобства крепления зубчатых колес на станках при обработке.

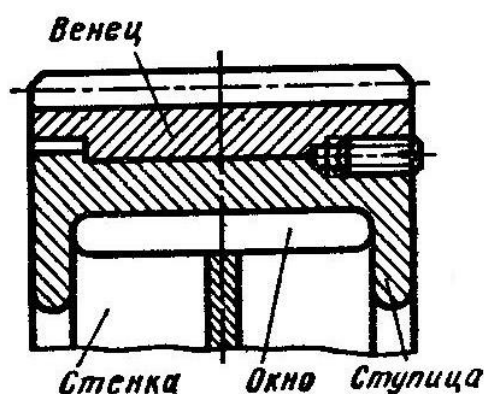


Рис. 2.1.6. Бандажированное зубчатое колесо.

При конструировании колеса наиболее важным требованием является его жесткость. Основные соотношения элементов зубчатых

колес в зависимости от их конструкции приведены в специальных справочниках.

Для экономии высокопрочных дорогостоящих материалов изготавливаются сборные конструкции — бандажированные колеса (Рис. 2.1.6). В этом случае зубчатый венец колеса изготавливается из качественной стали, а центральная часть делается из менее дорогого материала (например, чугуна).

## 2.2. Особенности расчета косозубых цилиндрических передач

Косозубые зубчатые передачи (Рис. 2.2.1), как и прямозубые, предназначены для передачи момента между параллельными валами.

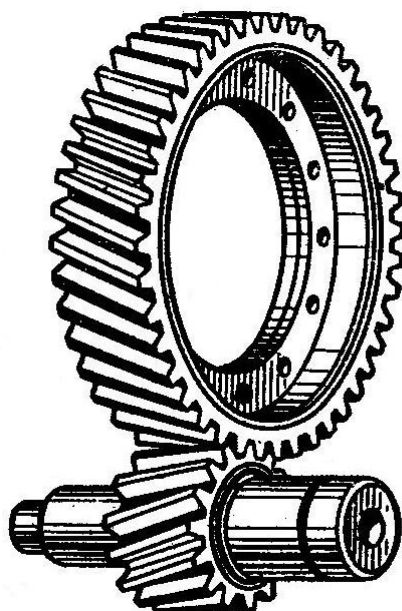


Рис. 2.2.1. Косозубая зубчатая передача.

У косозубых колес оси зубьев располагаются не по образующей делительного цилиндра, а по винтовой линии, составляющей с образующей угол  $\beta$ . Угол наклона зубьев  $\beta$  принимают равным  $8-15^\circ$ , он одинаков для обоих колес, но на одном из сопряженных колес зубья наклонены вправо, а на другом — влево. Передаточное число для одной пары  $u \leq 12$ . В прямозубых передачах линия контакта параллельна оси, а в косозубых она расположена по диагонали на поверхности зуба (контакт в прямозубых передачах осуществляется вдоль всей длины зуба, а в косозубых — сначала в точке увеличивается до прямой, «диагонально» захватывающей зуб, и, затем, постепенно уменьшается до точки).

Достоинства косозубых передач по сравнению с прямозубыми: уменьшение шума при работе, меньшие габаритные размеры, высокая плавность зацепления, большая нагрузочная способность, значительно меньшие дополнительные динамические нагрузки.

Профиль косого зуба в нормальном сечении  $n-n$  совпадает с профилем прямого зуба. В торцовом сечении  $t-t$  параметры косого зуба изменяются в зависимости от угла  $\beta$ .

Косозубые колеса в отличие от прямозубых имеют два шага и два модуля: в нормальном сечении по делительной окружности — нормальный шаг  $p_n$ , в торцовой плоскости — торцовый шаг  $p_t$ . Из условия, что модуль зацепления равен шагу, деленному на число  $\pi$ , имеем  $m_n = p_n/\pi$ ,  $m_t = p_t/\pi$ . Для косозубых колес значения нормального модуля  $m_n$  стандартизованы, так как профиль косого зуба в нормальном сечении соответствует исходному контуру инструментальной рейки и, следовательно,  $m = m_n$  (косозубые колеса нарезаются тем же способом и инструментом, что и прямозубые). Геометрические параметры цилиндрической косозубой передачи с эвольвентным профилем зуба рассчитываются по формулам, приведенным в Табл. 2.4.1. По торцовому модулю  $m_t$  рассчитываются делительные (начальные) диаметры, а по  $m_n$  — все остальные размеры зубчатых колес. Ввиду того, что в торцовом сечении косозубого зацепления профили зубьев имеют эвольвентное очертание, принято все расчеты и исследования проводить по торцовому сечению, в котором угол зацепления  $\alpha_t$  отличается от угла зацепления  $\alpha_n$  в нормальной плоскости. В расчетах на прочность и в производстве исходят из нормального модуля и угла зацепления  $\alpha_n$ , величина которого принимается равной  $20^\circ$ . Между углами  $\alpha_t$  и  $\alpha_n$  может быть установлена следующая зависимость:  $\operatorname{tg} \alpha_n = \operatorname{tg} \alpha_t \cos \beta$ .

Рассмотрим возможности параметризации на примере создания в системе CATIA V5 параметрической модели зубчатой передачи.

## **2.3. Создание параметрической модели цилиндрической прямозубой передачи в CAD-системе CATIA V5**

Запишем основные параметры и формулы, которые понадобятся для моделирования в виде следующей таблицы.

Таблица 2.3.1

**Основные параметры и формулы для построения геометрии  
цилиндрического прямозубого зубчатого колеса**

| №  | Обозначение    | Единицы измерения   | Формула                   | Описание  |
|----|----------------|---------------------|---------------------------|---|
| 1  | a              | градусы (угол)      | 20deg                     | профильный угол исходного контура                 |
| 2  | z              | целое (integer)     | —                         | число зубьев зубчатого колеса                     |
| 3  | m              | рациональное (real) | —                         | модуль цилиндрического зубчатого колеса           |
| 4  | h <sub>a</sub> | рациональное (real) | —                         | коэффициент высоты головки зуба (по умолчанию 1)  |
| 5  | h <sub>f</sub> | рациональное (real) | —                         | коэффициент высоты ножки зуба (по умолчанию 1.25) |
| 6  | d              | миллиметры (mm)     | $m \cdot z$               | диаметр делительной окружности                    |
| 7  | d <sub>b</sub> | миллиметры (mm)     | $d \cdot \cos(a)$         | диаметр основной окружности                       |
| 8  | d <sub>a</sub> | миллиметры (mm)     | $d + 2 \cdot h_a \cdot m$ | диаметр окружности вершин                         |
| 9  | d <sub>f</sub> | миллиметры (mm)     | $d - 2 \cdot h_f \cdot m$ | диаметр окружности впадин                         |
| 10 | r <sub>f</sub> | миллиметры (mm)     | $0.4 \cdot m$             | радиус кривизны переходной кривой зуба            |
| 11 | t              | рациональное (real) | $0 \leq t \leq 1$         | параметр для построения эвольвенты                |

|    |            |                      |  |  |
|----|------------|----------------------|--|--|
| 12 | x          | миллимет-<br>ры (mm) | $\frac{db}{2} * (\sin(t * \pi * 1 \text{rad}) ) - \frac{db}{2} * t * \pi * \cos(t * \pi * 1 \text{rad})$ | х координата для построения эвольвентного профиля зуба |
| 13 | y          | миллимет-<br>ры (mm) | $\frac{db}{2} * \cos(t * \pi * 1 \text{rad}) + \frac{db}{2} * t * \pi * \sin(t * \pi * 1 \text{rad})$    | у координата для построения эвольвентного профиля зуба |
| 14 | disk_depth | миллимет-<br>ры (mm) | —  | толщина диска  |

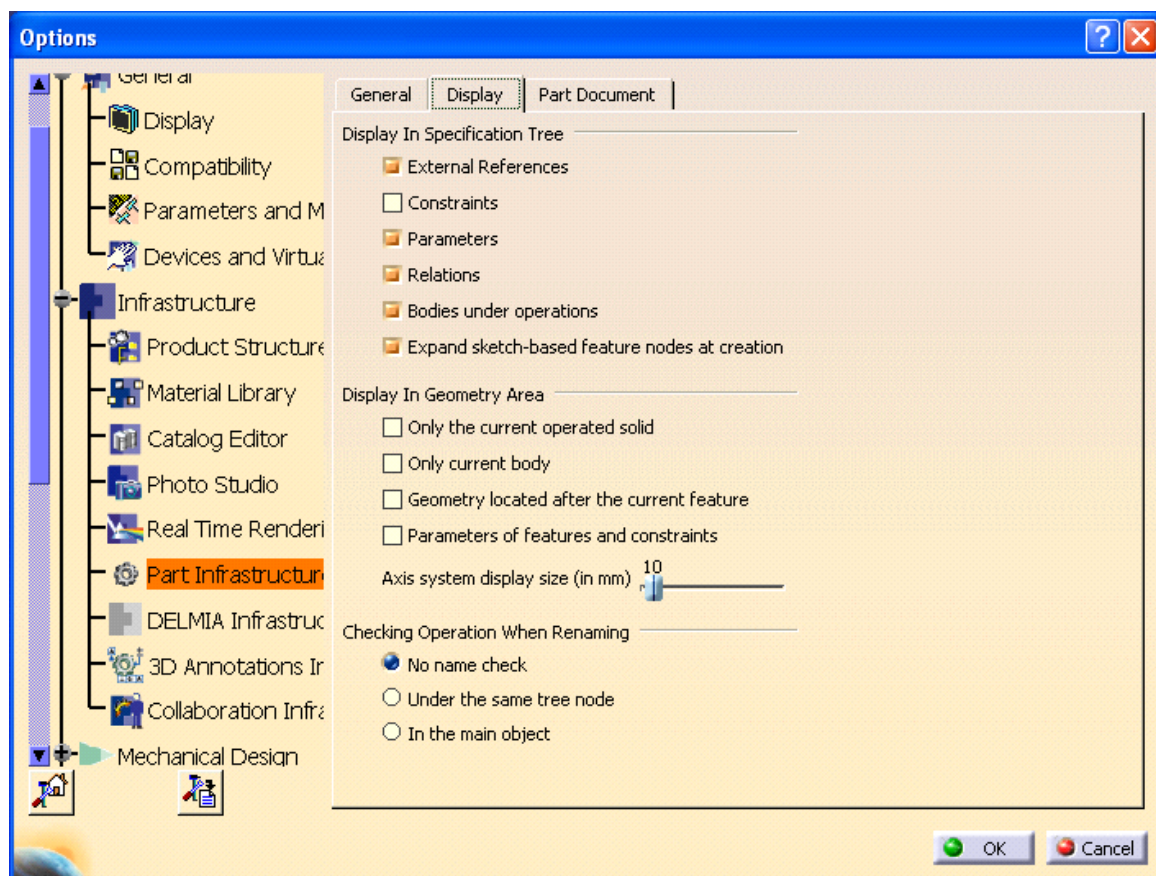


Рис. 2.3.1. Диалоговое окно изменений свойств системы.

#### Шаг 1. Создание новой модели.

Войдите в систему и создайте новый “Part” (“File → New...” затем из списка выберите “Part”). Присвойте ему имя Gear.

#### Шаг 2. Отображение параметров и формул в дереве модели.

Зайдите в опции системы (“Tools → Options...”). Слева в дереве выберите “Infrastructure → Part Infrastructure” перейдите на закладку “Display” (Рис. 2.3.1). Убедитесь, что выделенные пункты активны. Если это не так, активизируйте их.



Затем, в дереве выберите “General → Parameters and Measure”. Перейдите на закладку “Knowledge” и активизируйте выделенные пункты (Рис. 2.3.2).

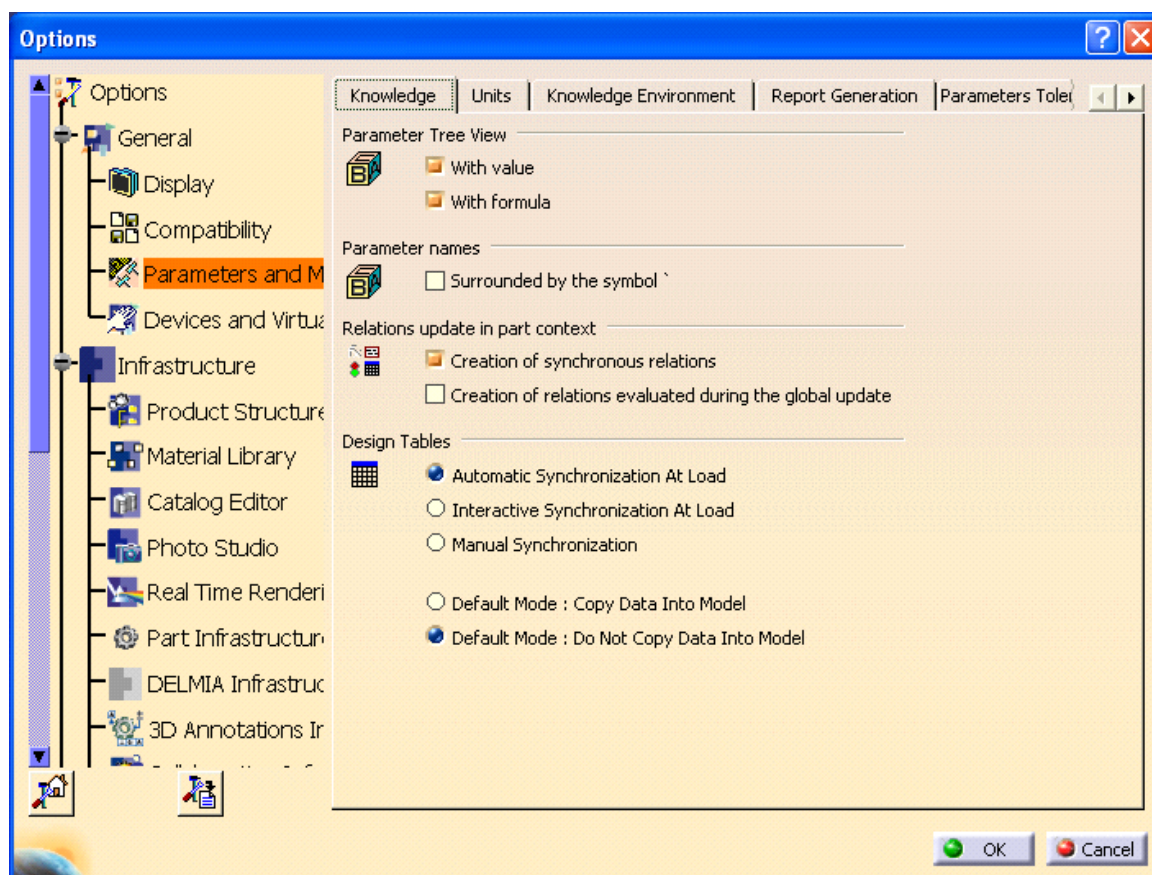



Рис. 2.3.2. Диалоговое окно изменений свойств системы.

### Шаг 3. Задание основных параметров.

Перейдите в модуль “Generative Shape Design” и щелкните по кнопке .

Теперь Вы можете задать основные параметры и формулы необходимые для построения модели.

Например (Рис. 2.3.3), чтобы задать модуль  $m$  зубчатого колеса выберите значение “Real” в поле 1, нажмите на кнопку “New Parameter of type”, в поле 2 задайте имя « $m$ », а в поле 3 значение этого параметра. Щелкните на кнопке “Apply”.

Точно также задайте значения для  $z$ ,  $a$ ,  $ha$ ,  $hf$  и  $disk\_depth$ . Эти параметры появятся в дереве модели (Рис. 2.3.4).

Теперь зададим параметры, значения которых высчитываются по формулам.

Для этого в том же окне “Formulas” создайте новый параметр  $d$ , тип которого “Length”, но вместо того, чтобы задать его значение нажмите на кнопку “Add Formula”.

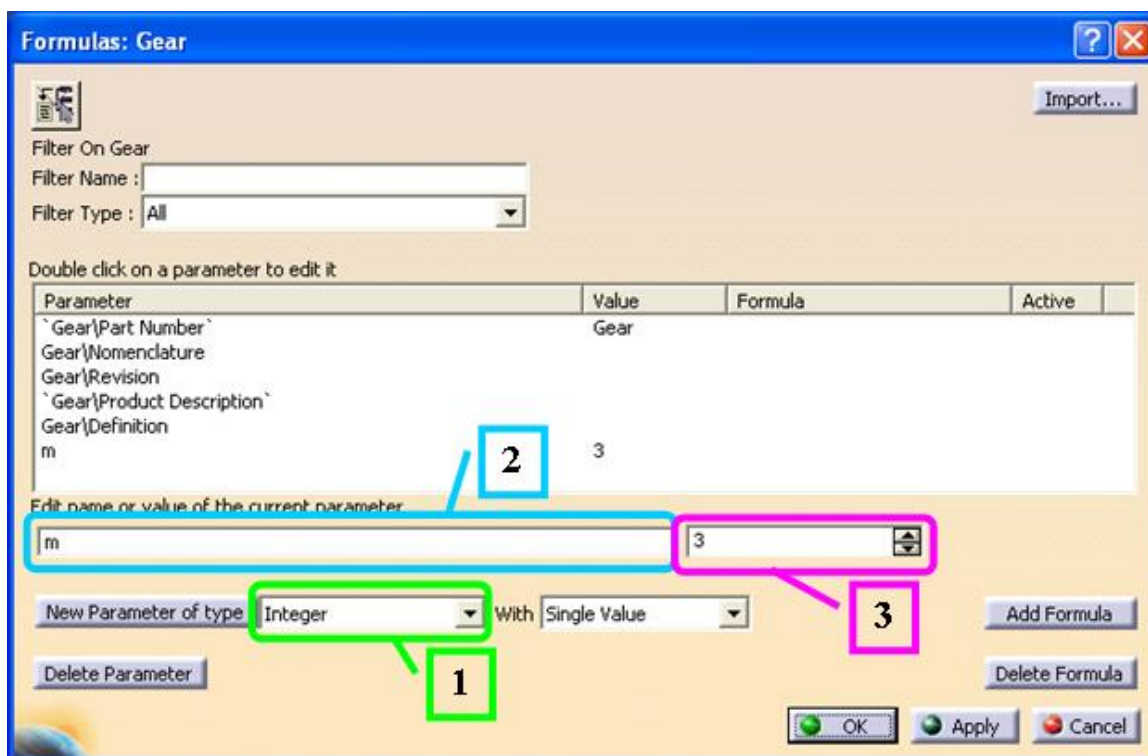


Рис. 2.3.3. Диалоговое окно задания параметров.

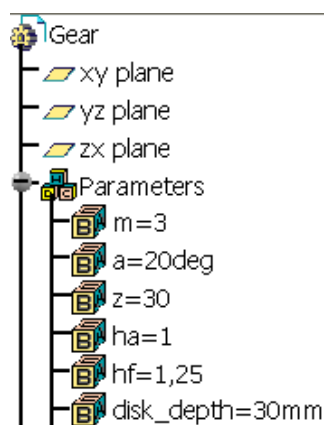


Рис. 2.3.4. Дерево модели.

В появившемся окне в поле 1 (Рис. 2.3.5) напишите формулу соответствующую данному параметру из Табл. 2.3.1.

Заметьте, что когда в формуле Вы записываете наименование одного из параметров, заданных ранее, его значение отображается в поле 2.

Введите формулы для всех остальных параметров. Получится дерево их представления, показанное на Рис. 2.3.6.

Шаг 4. Задание координат точек эвольвенты с помощью закона (law).

Для того чтобы создать закон нажмите на кнопку “fog” и в появившемся окне введите имя (Рис. 2.3.7).

Нажмите “OK”.

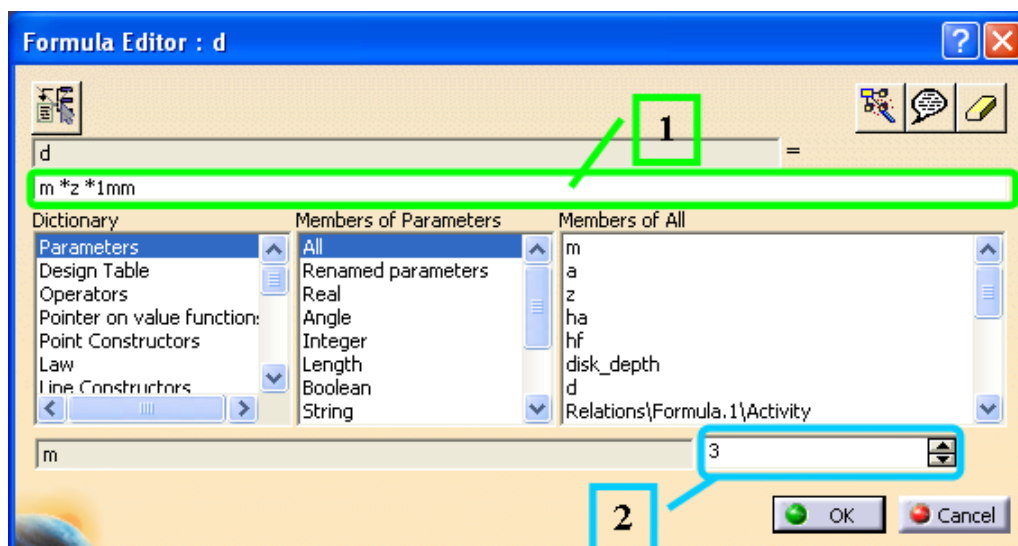


Рис. 2.3.5. Диалоговое окно задания формул.

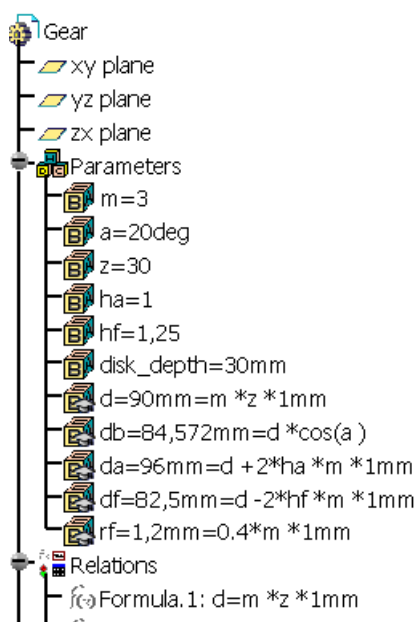


Рис. 2.3.6. Дерево модели.

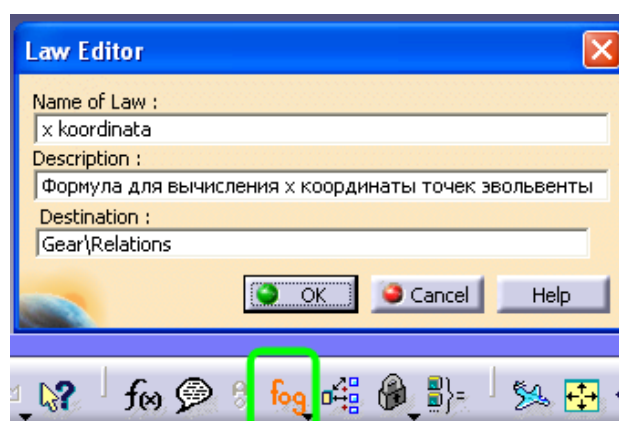


Рис. 2.3.7. Диалоговое окно задания имени и описания создаваемого закона.

В появившемся окне (Рис. 2.3.8) создайте два новых параметра:  $t$  (тип Real) и  $x$  (тип Length).

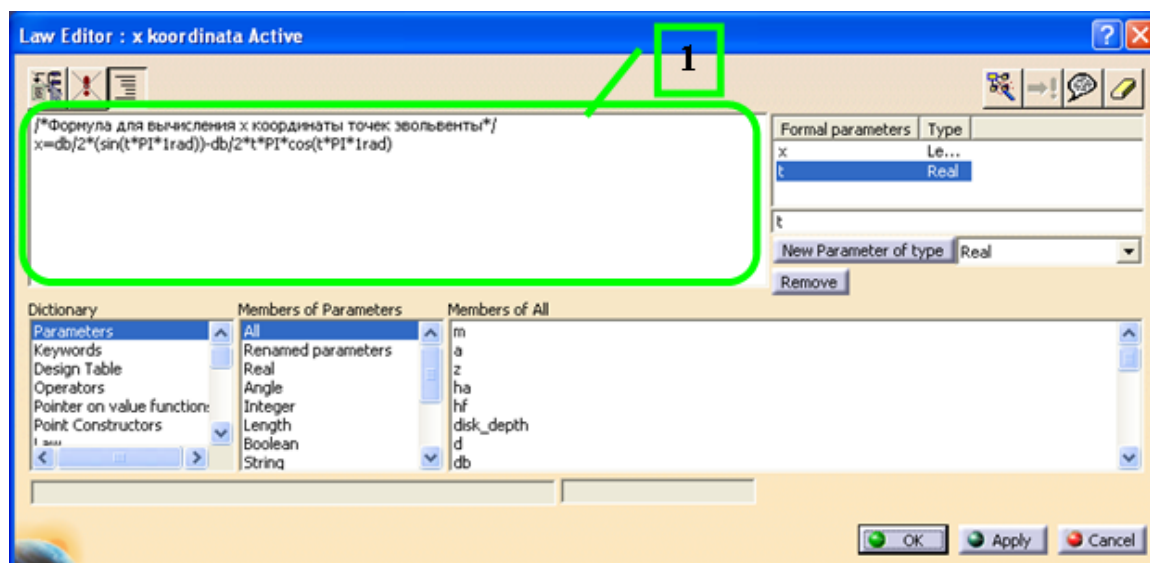


Рис. 2.3.8. Диалоговое окно задания формулы соотношения параметров.

В поле 1 запишите формулу для вычисления координаты  $x$ :  
Создайте закон для координаты  $y$ .

#### Шаг 5. Создание профиля зуба.

Создайте 5 точек на плоскости  $XY$ . Измените “H” и “V” координаты точек. Например, для точки 1:

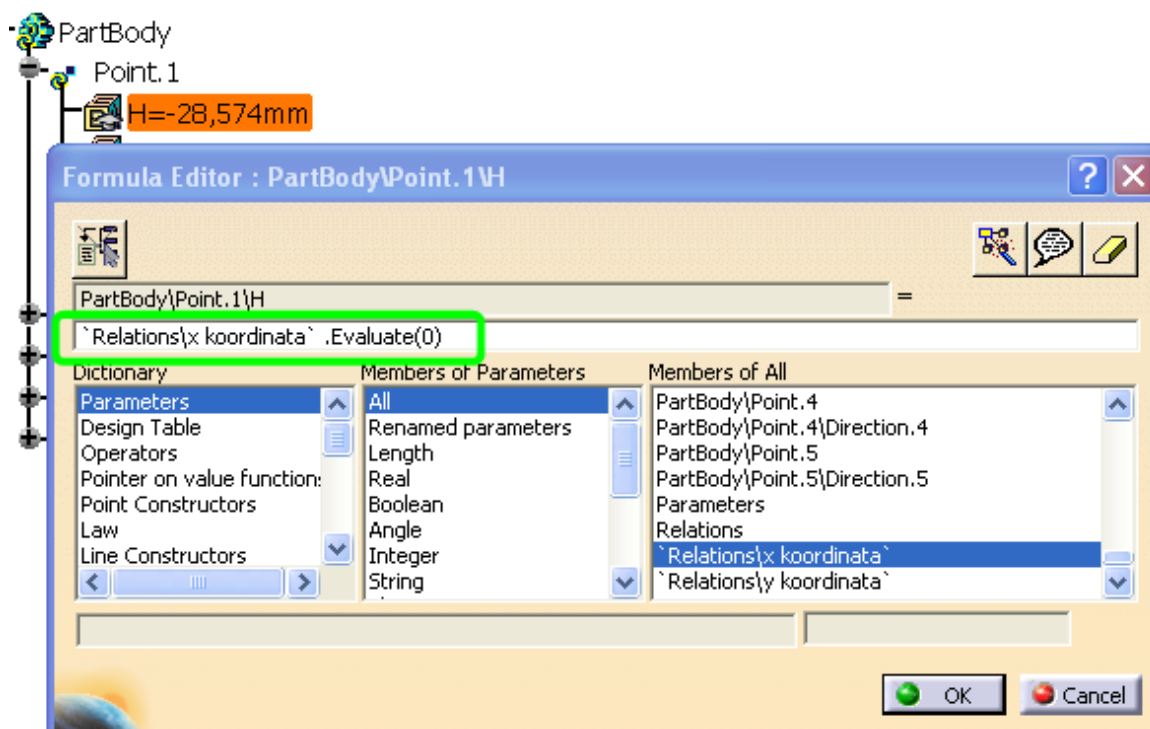


Рис. 2.3.9. Диалоговое окно редактора формул.

Щелкните правой кнопкой на “Н” координате и выберите “Edit Formula”. Введите в выделенное поле указанную формулу. Для этой точки  $t = 0$  (Рис. 2.3.9).

Нажмите “ОК”. Положение точки изменится.

Также измените координату “V”, выбрав соответственно закон для координаты y.

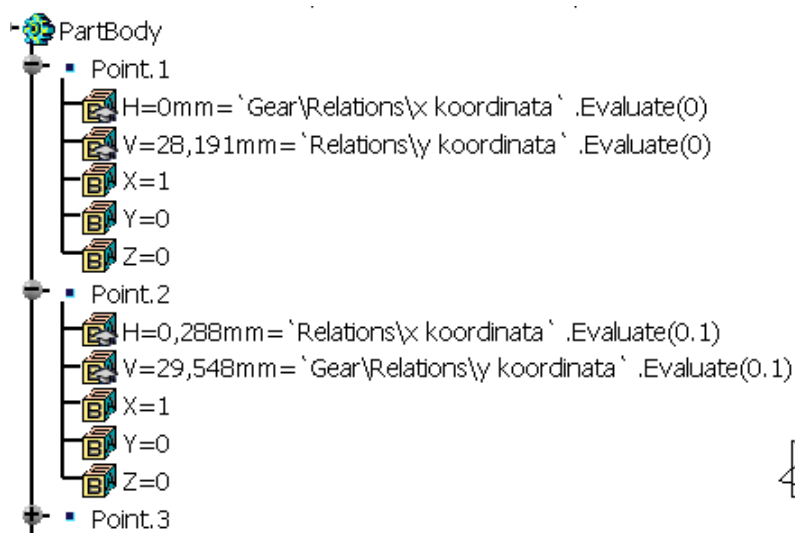


Рис. 2.3.10. Дерево модели с заданными, с помощью формул, координатами точек.

Повторите эти операции для остальных точек (Рис. 2.3.10). Параметр  $t$  изменяется от 0.1 (для точки 2) до 0.4 (для точки 5).

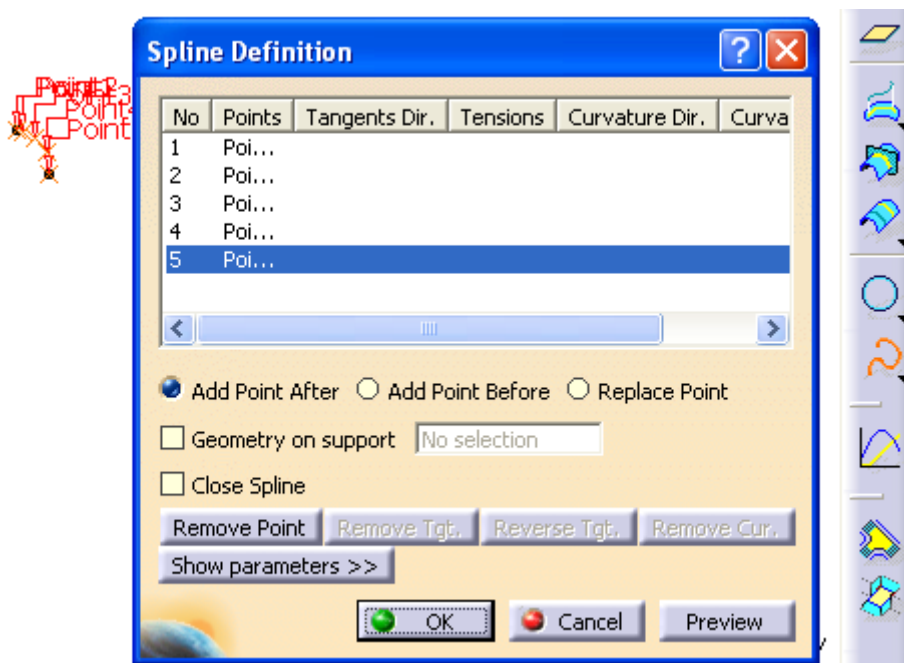


Рис. 2.3.11. Диалоговое окно создания сплайна.

Создайте сплайн, используя построенные точки (Рис. 2.3.11).

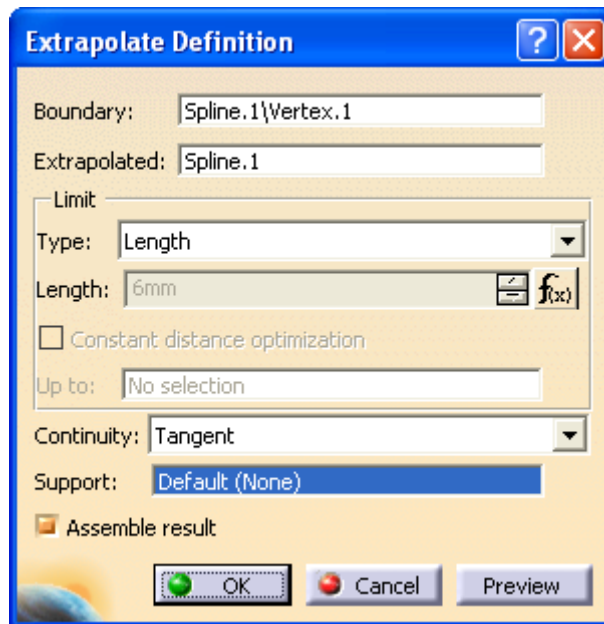



Рис. 2.3.12. Диалоговое окно команды “Extrapolate”.

Используя команду “Extrapolate” , удлините сплайн в сторону центра колеса на расстояние  $2 \cdot r_m$  (Рис. 2.3.12). Это обусловлено тем, что при  $z < 45$  окружность впадин лежит ниже основной окружности, и эвольвента на этом отрезке не определена.

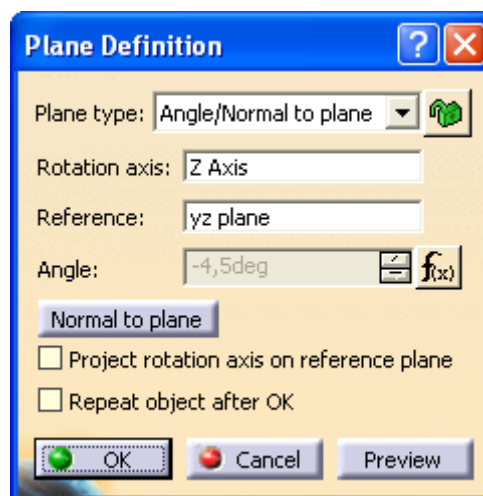




Рис. 2.3.13. Диалоговое окно создания плоскости.

Далее, создайте плоскость под углом, равным  $90\text{deg}/z$  к плоскости  $ZY$  (Рис. 2.3.13).

Используя команду “Circle” , постройте 3 окружности: делительную, впадин и вершин (Рис. 2.3.14, 2.3.15).

Создайте скругление между эвольвентой и окружностью впадин, с помощью команды “Corner” (Рис. 2.3.16).

С помощью команды “Split”  удалите лишние участки эвольвенты (Рис. 2.3.17).



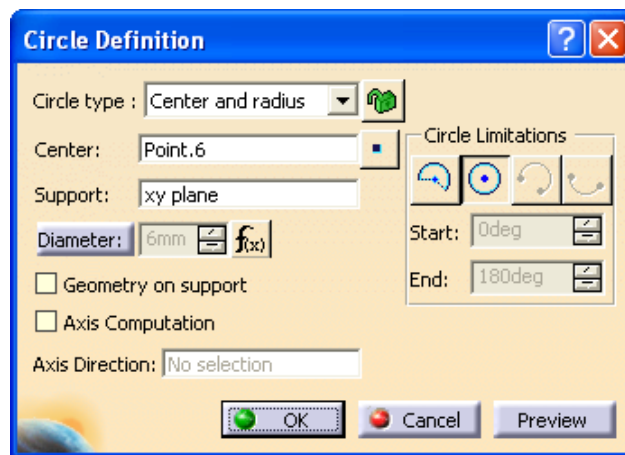


Рис. 2.3.14. Диалоговое окно команды “Circle”.

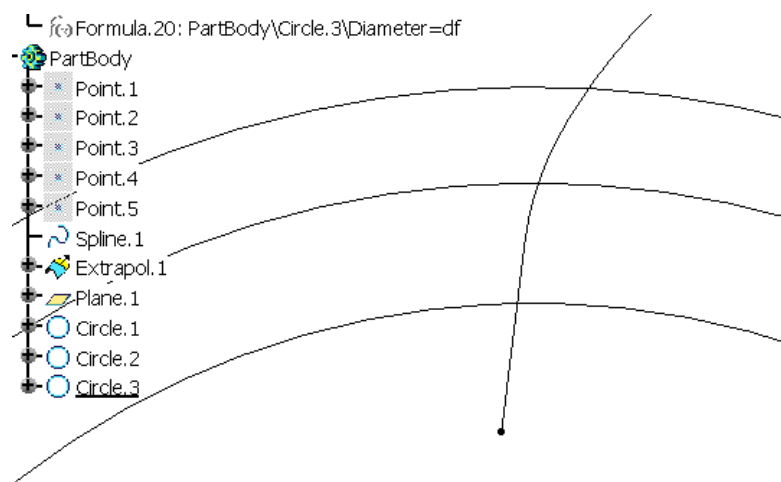


Рис. 2.3.15. Часть экрана с тремя окружностями.

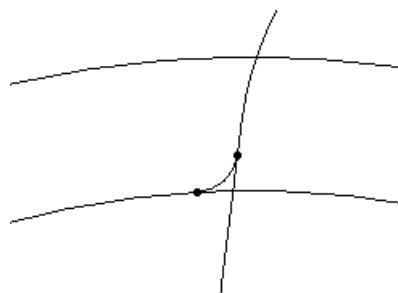


Рис. 2.3.16. Результат применения команды “Corner”.

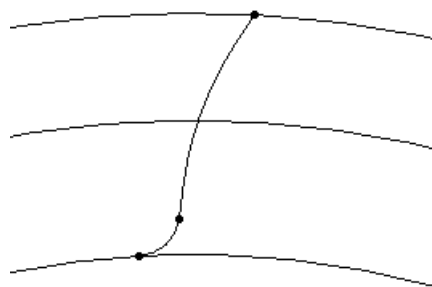



Рис. 2.3.17. Результат применения команды “Split”.

Создайте зеркальную копию получившегося профиля с помощью команды “Symmetry” . В качестве плоскости симметрии используйте ранее созданную плоскость (Рис. 2.3.18).

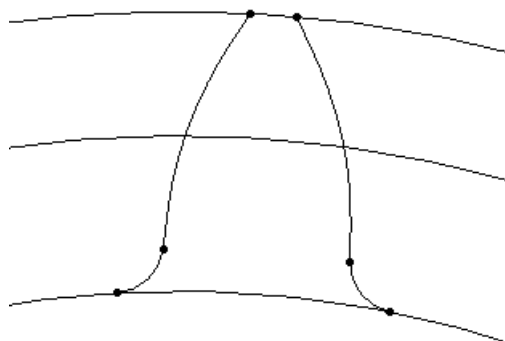



Рис. 2.3.18. Результат применения команды “Symmetry”.

Используя команду “Trim” , обрежьте окружности впадин и вершин (Рис. 2.3.19).

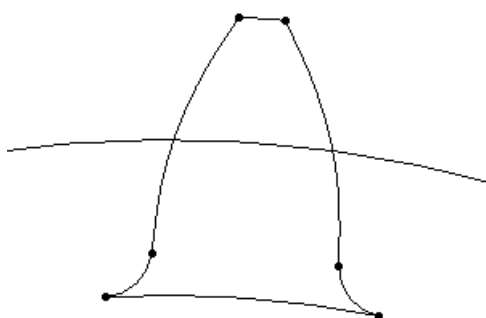



Рис. 2.3.19. Результат применения команды “Trim”.

Используя команду “Join” , объедините куски профиля в единое целое.

Перейдите в модуль “Part Design”. Создайте новое тело (“Insert → Body”).

Используя команду “Pad” , создайте зуб, задав длину равную значению параметра `disk_depth` (Рис. 2.3.20).

Создайте круговой массив элементов, укажите с помощью формул число зубьев и угловой шаг ( $360\text{deg}/z$ ) (Рис. 2.3.21).

Создайте цилиндр, диаметр которого равен диаметру впадин, а высота — значению параметра `disk_depth` (Рис. 2.3.22).

Получилось зубчатое колесо. Теперь чтобы создать колесо с другим количеством зубьев и/или модулем, необходимо просто изменить значения этих параметров в дереве модели (Рис. 2.3.23).



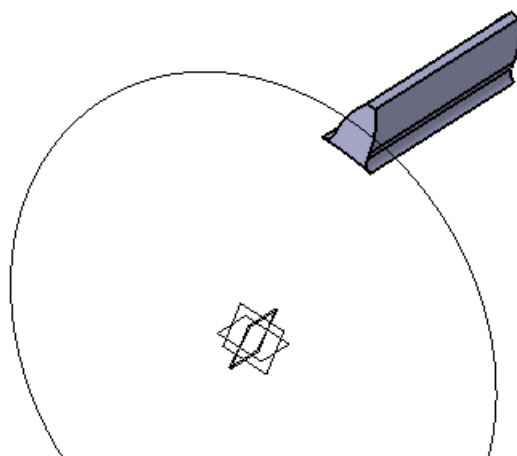


Рис. 2.3.20. Результат применения команды “Pad”.

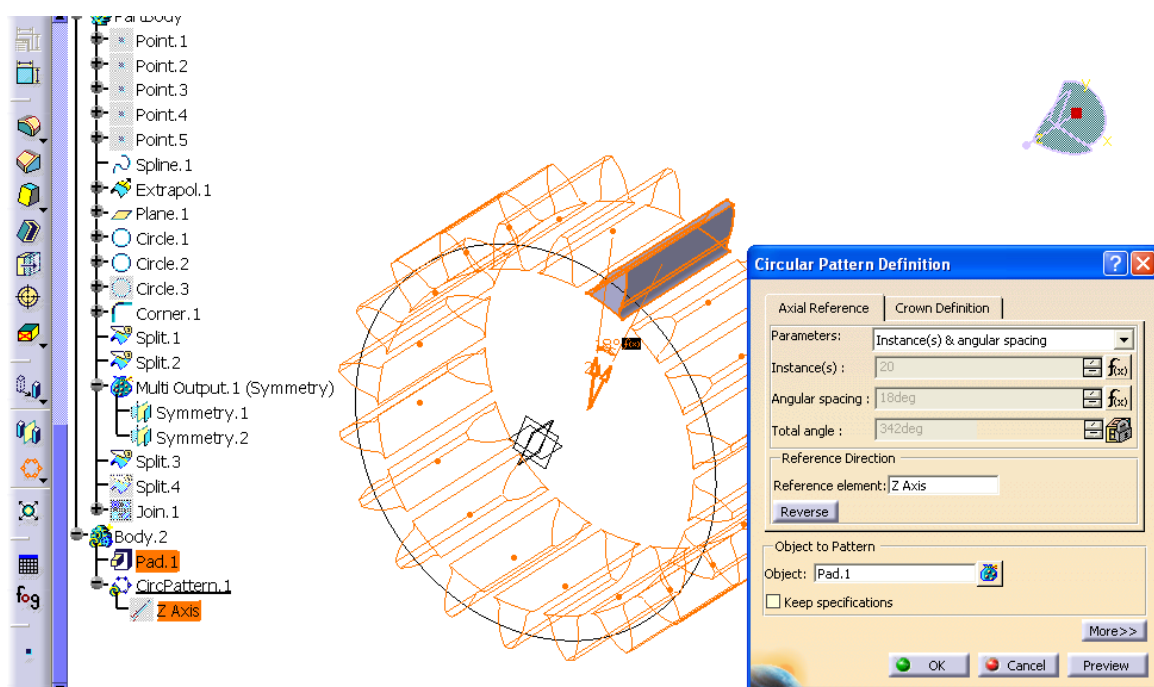


Рис. 2.3.21. Предварительный просмотр результатов команды “Circular Pattern”.

В результате получаем готовую модель цилиндрического зубчатого колеса с другим количеством зубьев (Рис. 2.3.24).

#### Шаг 6. Сборка двух колес.

Создайте точку контакта: для этого на плоскости XY создайте произвольную точку и задайте ее координаты с помощью формул

$$\text{'PartBody\contact point\H'} = \text{Relations\y .Evaluate(a /180deg)}$$

$$\text{'PartBody\contact point\V'} = \text{Relations\y .Evaluate(a /180deg)}$$

Создайте два зубчатых колеса с помощью построенной модели, задав  $z_1 = 18, z_2 = 39, m = 2.5$ .

Сохраните каждое в отдельный файл.

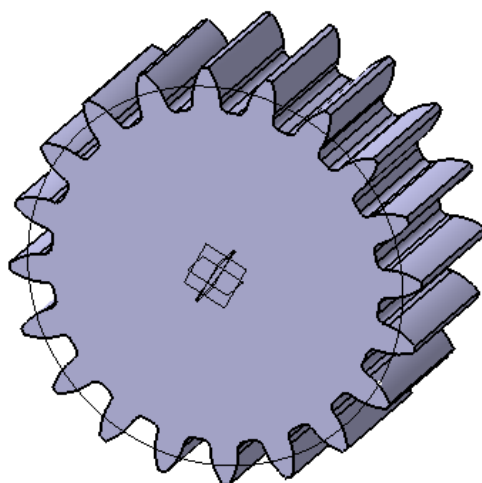


Рис. 2.3.22. Модель зубчатого колеса.

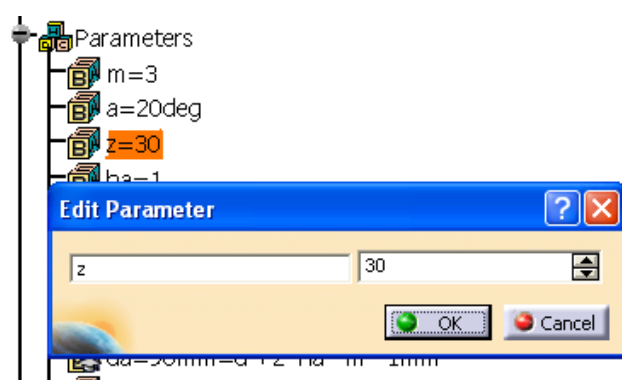


Рис. 2.3.23. Диалоговое окно редактора значений параметров.

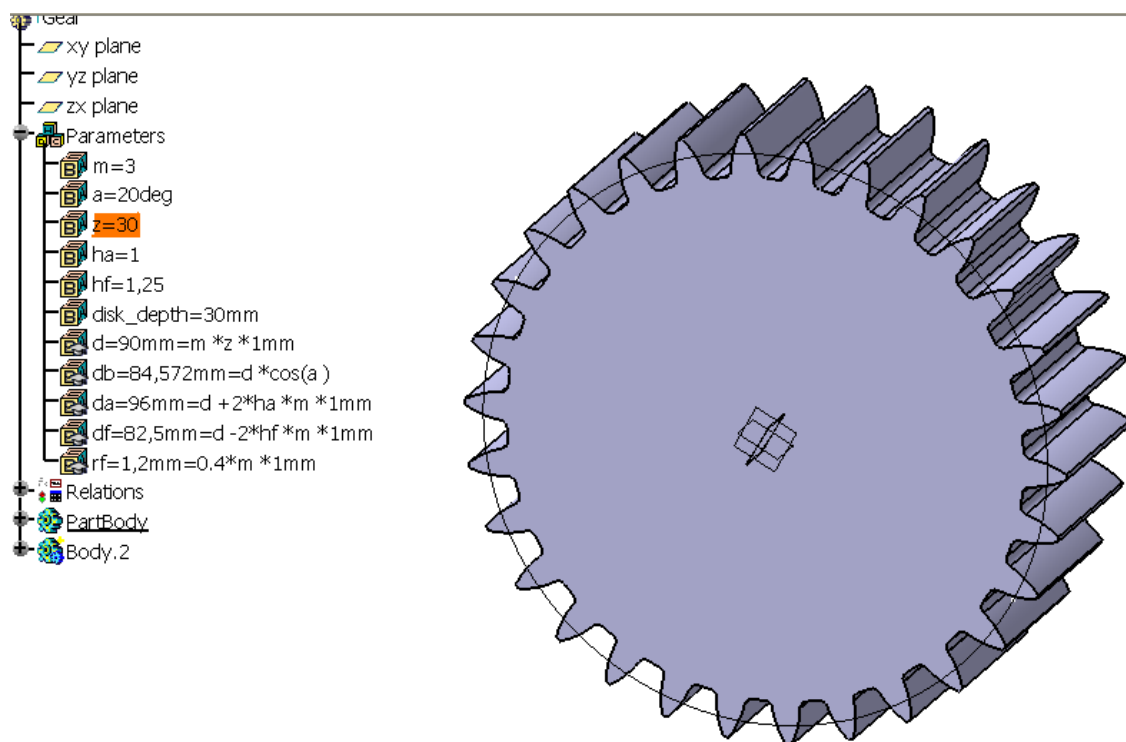





Рис. 2.3.24. Модель зубчатого колеса после изменения количества зубьев.

Перейдите в модуль “Assemble Design”. Добавьте одно из колес в пустую сборку, зафиксируйте его с помощью команды “Fix” . Добавьте второе колесо. Наложите связь параллельности на оси колес с помощью команды “Angle Constraint” . Затем, с помощью команды “Offset” , задайте расстояние между плоскостями XY обоих колес, равным 0мм (Рис. 2.3.25).

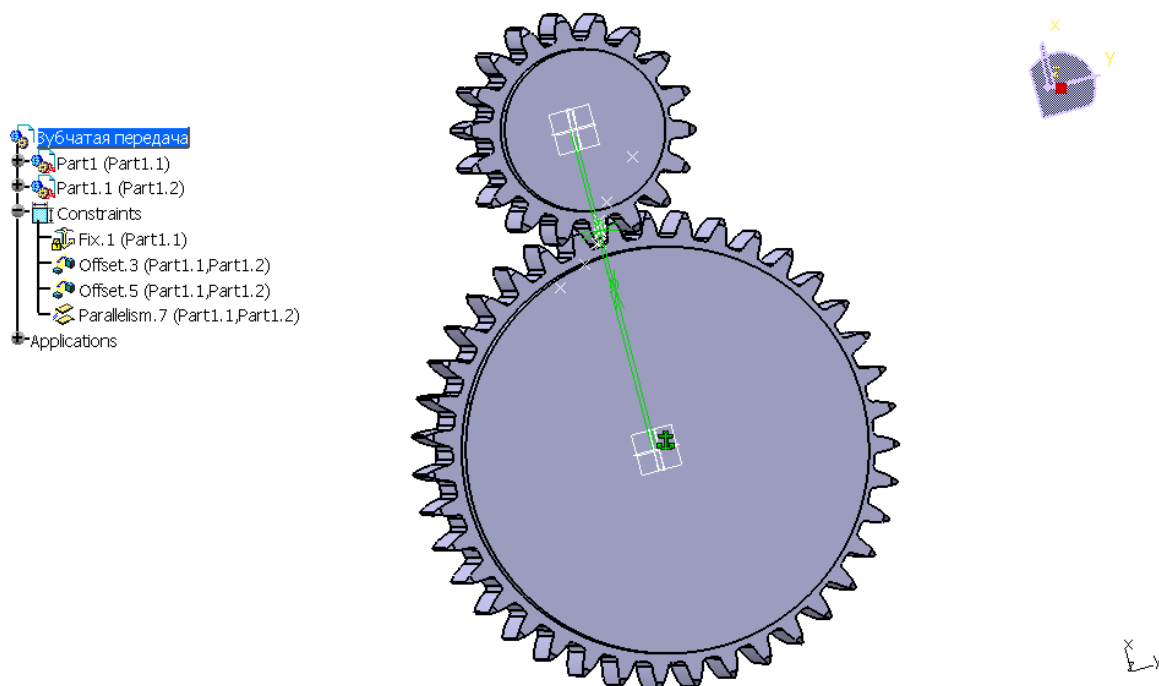


Рис. 2.3.25. Результат сборки двух зубчатых колес.

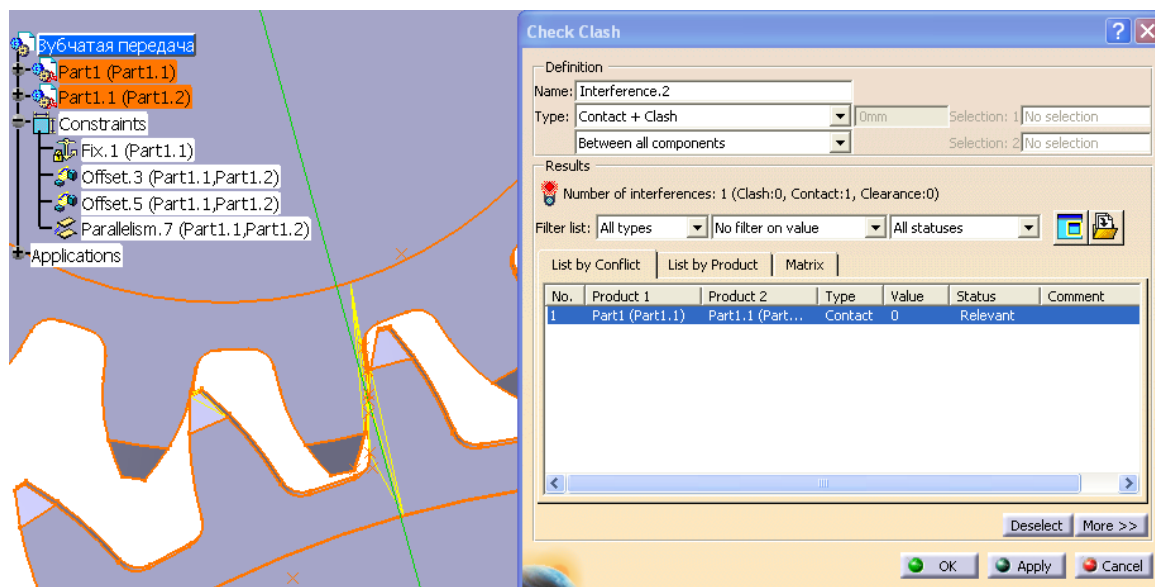


Рис. 2.3.26. Диалоговое окно команды “Clash” с полученными результатами проверки.

И, наконец, задайте расстояние между точками контакта равным 0мм.

С помощью команды “Clash”  можно убедиться, что между двумя колесами нет пересечений (Рис. 2.3.26).

Попробуйте теперь изменить количество зубьев у колес. Колеса перестроятся, но останутся в контакте.

## 2.4. Создание параметрической модели цилиндрической косозубой передачи в CAD-системе CATIA V5

Создание модели косозубого колеса очень похоже на предыдущий пример, поэтому здесь будут описаны лишь основные различия.

Во-первых, появляется новый параметр  $\beta$  — угол наклона зубьев. Во-вторых, появляется разница между нормальным ( $m_n$ ) и торцевым ( $m_t$ ) модулем,  $m_t = m_n / \cos(\beta)$ .

Таблица 2.4.1

Основные параметры и формулы для построения геометрии цилиндрического косозубого зубчатого колеса

| № | Обозначение | Единицы измерения   | Формула                          | Описание   |
|---|-------------|---------------------|----------------------------------|--|
| 1 | $\beta$     | градусы (угол)      | —                                | угол наклона зубьев                              |
| 2 | $a$         | градусы (угол)      | $20\text{deg}$                   | профильный угол исходного контура                |
| 3 | $\alpha_t$  | градусы (угол)      | $\arctan(\tan(a) / \cos(\beta))$ | профильный угол на торце                         |
| 4 | $z$         | целое (integer)     | —                                | число зубьев зубчатого колеса                    |
| 5 | $m = m_n$   | рациональное (real) | —                                | модуль цилиндрического зубчатого колеса          |
| 6 | $m_t$       | рациональное (real) | $m_n / \cos(\beta)$              | торцевой модуль                                  |
| 7 | $h_a$       | рациональное (real) | —                                | коэффициент высоты головки зуба (по умолчанию 1) |


|    |            |                     |   |  |
|----|------------|---------------------|---|--|
| 8  | hf         | рациональное (real) | —   | коэффициент высоты ножки зуба (по умолчанию 1.25)      |
| 9  | d          | миллиметры (mm)     | $mt * z$  | диаметр делительной окружности                         |
| 10 | db         | миллиметры (mm)     | $d * \cos(at)$  | диаметр основной окружности                            |
| 11 | da         | миллиметры (mm)     | $d + 2 * ha * mn$   | диаметр окружности вершин                              |
| 12 | df         | миллиметры (mm)     | $d - 2 * hf * mn$   | диаметр окружности впадин                              |
| 13 | rf         | миллиметры (mm)     | $0.4 * mt$  | радиус кривизны переходной кривой зуба                 |
| 14 | t          | рациональное (real) | $0 \leq t \leq 1$   | параметр для построения эвольвенты                     |
| 15 | x          | миллиметры (mm)     | $db/2 * (\sin(t * \pi * 1rad)) - db/2 * t * \pi * \cos(t * \pi * 1rad)$   | x координата для построения эвольвентного профиля зуба |
| 16 | y          | миллиметры (mm)     | $(db/2 * \cos(t * \pi * 1rad)) + (db/2 * t * \pi) * \sin(t * \pi * 1rad)$ | y координата для построения эвольвентного профиля зуба |
| 17 | disk_depth | миллиметры (mm)     | —   | толщина диска  |

Итак, добавим новые параметры в нашу модель и изменим формулы.

Теперь необходимо изменить Body.2.

Удалите Pad.1, CircPattern и Pad2 из дерева Body.2.

Перейдите в модуль “Generative Shape Design”.

Используя делительную окружность, создайте цилиндрическую поверхность с помощью команды “Extrude”  (Рис. 2.4.1).

На поверхности XY с помощью “Sketcher” создайте точку на делительной окружности в центре профиля зуба (Рис. 2.4.2).

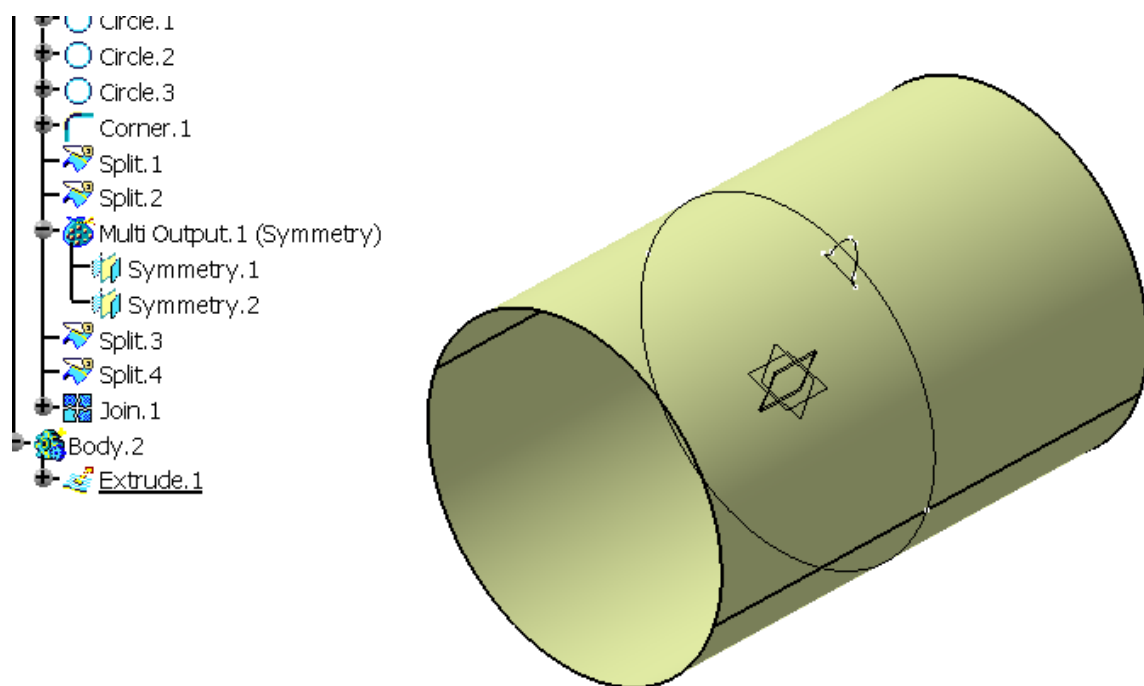


Рис. 2.4.1. Результат применения команды “Extrude”.

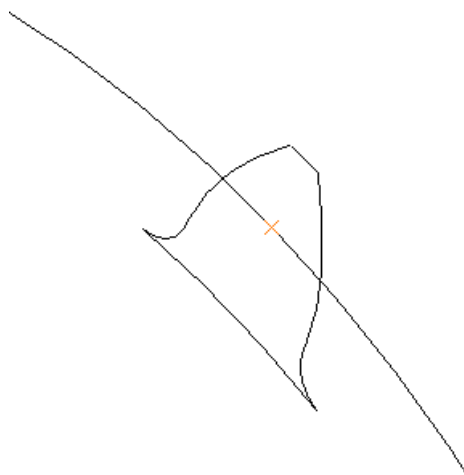




Рис. 2.4.2. Построение начальной точки винтовой линии.

Используя команду “Helix” , создайте винтовую линию.  $Pitch$  (шаг) =  $\pi \cdot d \cdot \tan(90deg - \beta)$ ,  $Height$  (длина витка) =  $\pi \cdot d$  (Рис. 2.4.3).

Перейдите в модуль “Part Design”. Используя команду “Rib” , создайте модель зуба (Рис. 2.4.4).

Создайте круговой массив зубьев и цилиндр. Модель косозубого цилиндрического колеса готова (Рис. 2.4.5).

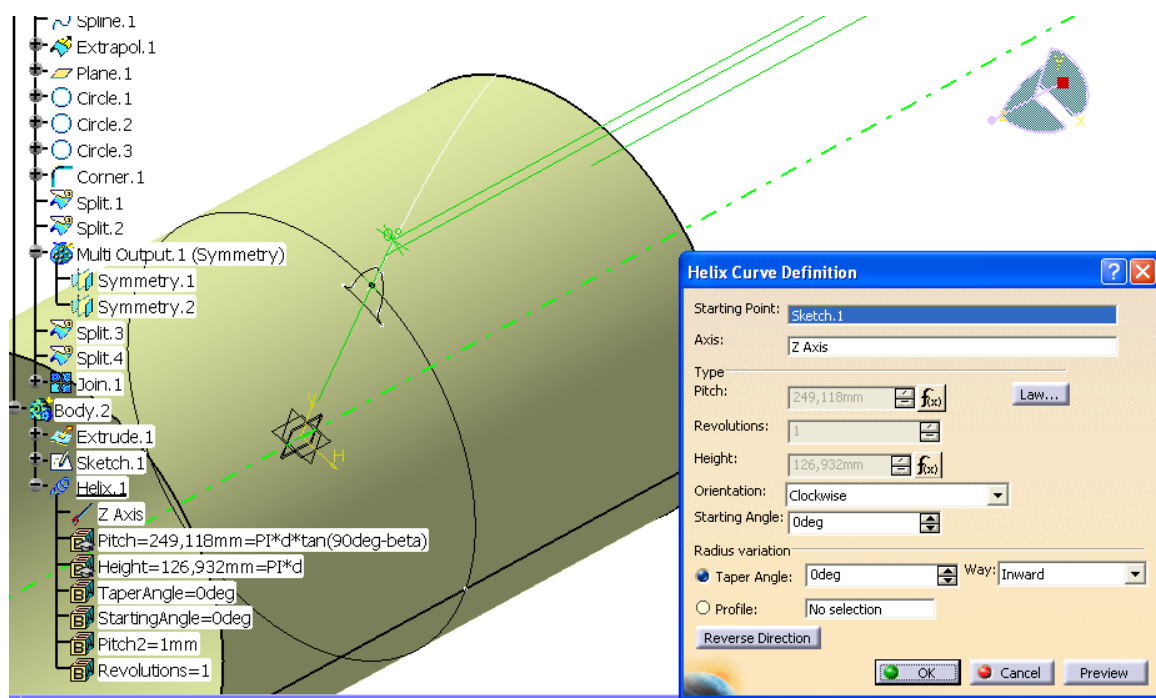


Рис. 2.4.3. Построение винтовой линии.

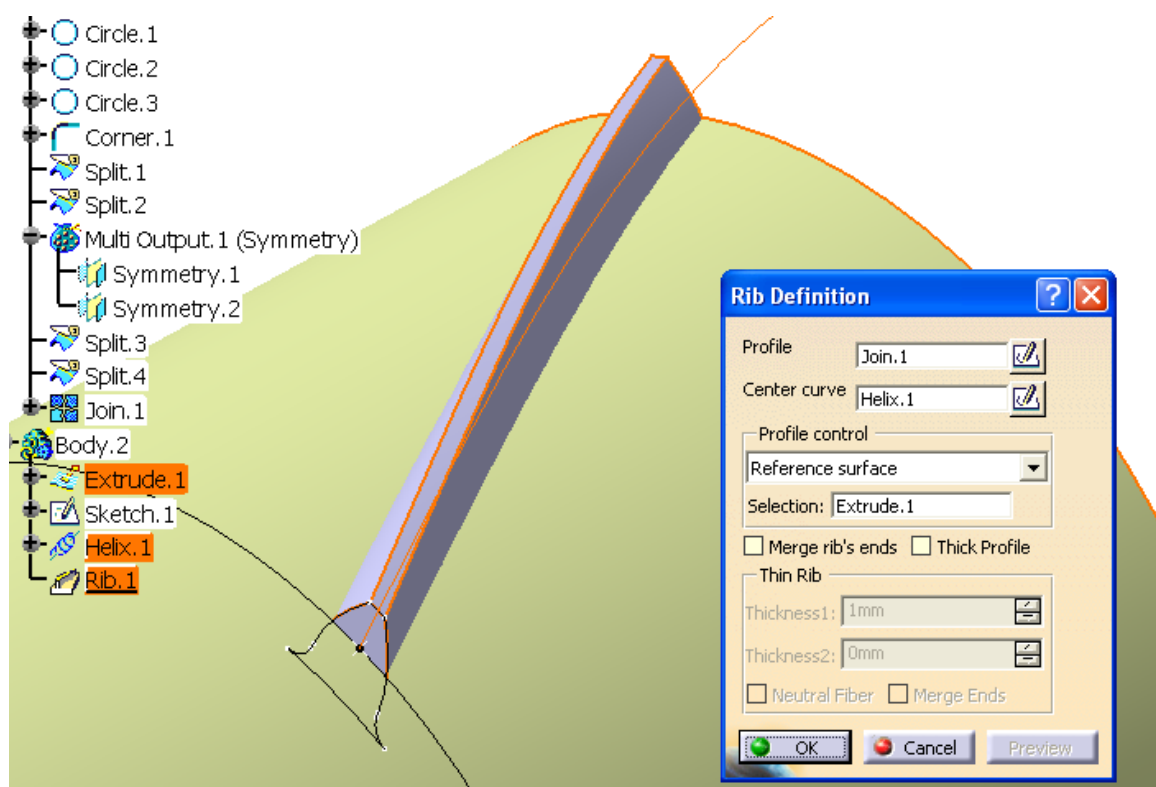


Рис. 2.4.4. Предварительный просмотр результатов команды “Rib”.



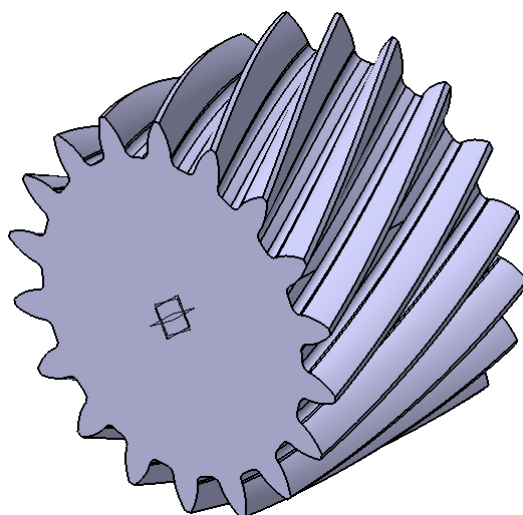


Рис. 2.4.5. Модель косозубого зубчатого колеса.

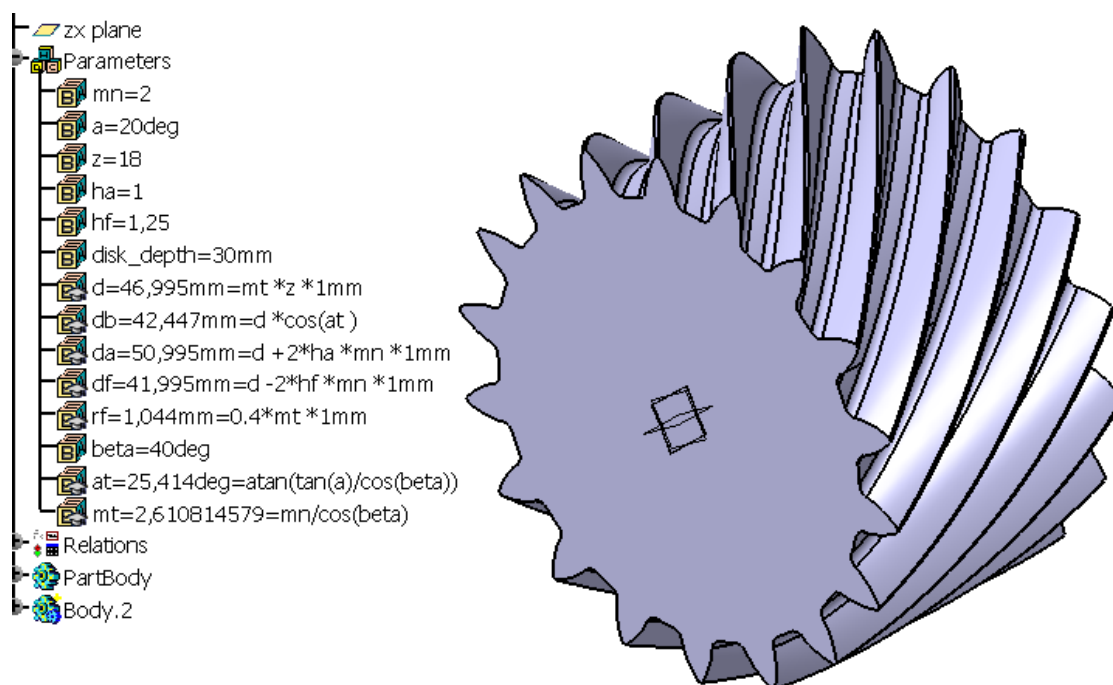


Рис. 2.4.6. Модель косозубого зубчатого колеса с другим количеством зубьев.

Теперь поменяв, например, угол  $\beta$  получим другое колесо (Рис. 2.4.6).

При этом следует отметить, что при  $\beta = 0$  получим модель прямозубого колеса (Рис. 2.4.7).



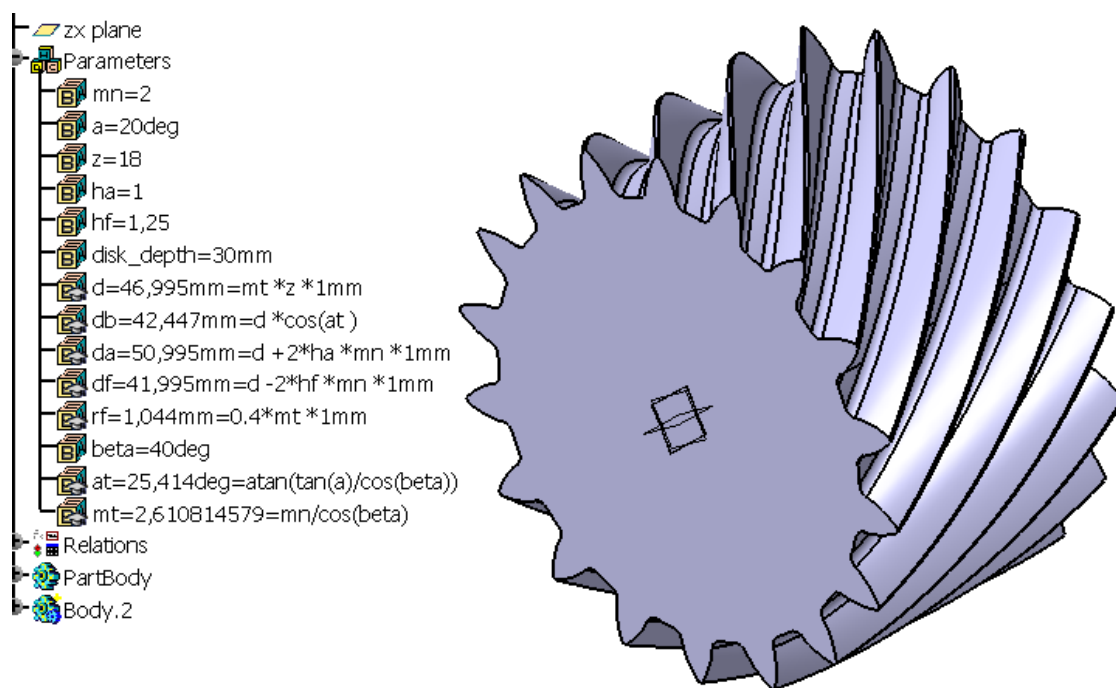


Рис. 2.4.7. Модель косозубого зубчатого колеса при  $\beta = 0^\circ$ .

# Глава 3. Червячные передачи

## 3.1. Общие сведения

Червячной — называется зубчатая передача, состоящая из двух подвижных звеньев — червяка и зубчатого колеса и предназначенная для передачи и преобразования вращательного движения между ортогональными перекрещивающимися осями. Червяком называется звено, наружная поверхность которого имеет форму винта. Червячным колесом называется зубчатое колесо с косыми зубьями, которое зацепляется с червяком.

По форме тела червяки разделяются на цилиндрические (Рис. 3.1.1 а), в), глобоидные (Рис. 3.1.1 б) и тороидные. Наибольшее применение находят цилиндрические червяки как более простые в изготовлении и обеспечивающие достаточно высокую нагрузочную способность.

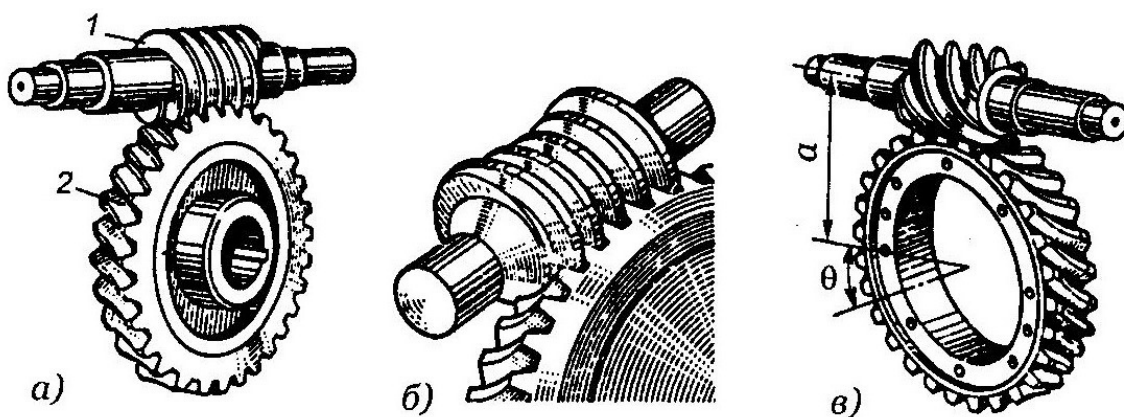


Рис. 3.1.1. Форма тела червяка: а), в) цилиндрические б) глобоидные.

По форме боковой поверхности витка червяки разделяются на архимедовы (обозначение ZA), конволютные (ZN), эвольвентные (Z1), нелинейчатые с поверхностью, образованной конусом (ZK) и с вогнутым профилем витка (ZT).

### Преимущества:

- благодаря малому числу заходов червяка ( $z_1 = 1 \dots 4$ ) червячная передача позволяет реализовывать в одной ступени большие передаточные отношения;
- обладает высокой плавностью, низким уровнем вибраций и шума;
- позволяет обеспечить самоторможение червячного колеса (при малых углах подъема витка передача движения от вала червячного колеса к червяку становится невозможной).

### Недостатки:

- высокая скорость скольжения вдоль линии зуба, что ведет к повышенной склонности к заеданию (необходимы специальные смазки и материалы для зубчатого венца червячного колеса), снижению КПД и более высокому тепловыделению;
- значительное выделение теплоты в зоне зацепления червяка с колесом. Для уменьшения нагрева в червячной передаче применяются масляные резервуары с ребристыми стенками с целью более интенсивной теплоотдачи в окружающий воздух, обдув корпуса и другие способы охлаждения;
- ограниченная возможность передачи значительных мощностей, обычно до 50 кВт;
- необходимость применения для венцов червячных колес дефицитных антифрикционных материалов;
- повышенное изнашивание.

## 3.2. Геометрические параметры и способы изготовления червячных передач

Расчет геометрии зацепления цилиндрической червячной передачи регламентируется ГОСТ 19650 – 74. При буквенных обозначениях червяк отличается индексом 1, а колесо индексом 2. Связь между основными параметрами червяка — диаметром начального цилиндра  $d_{w1}$ , ходом винтовой линии  $p_{z1}$  и углом ее наклона  $\beta_{w1}$  устанавливается следующими соотношениями (Рис. 3.2.1)  $p_{z1} = \pi \cdot d_{w1} \cdot \tan \beta_{w1}$ ,  $p_1 = t \cdot \pi$ , причем связь между ходом винтовой линии  $p_{z1}$  и шагом многозаходного винта задается при помощи соотношения  $p_{z1} = p_1 z_1$ .

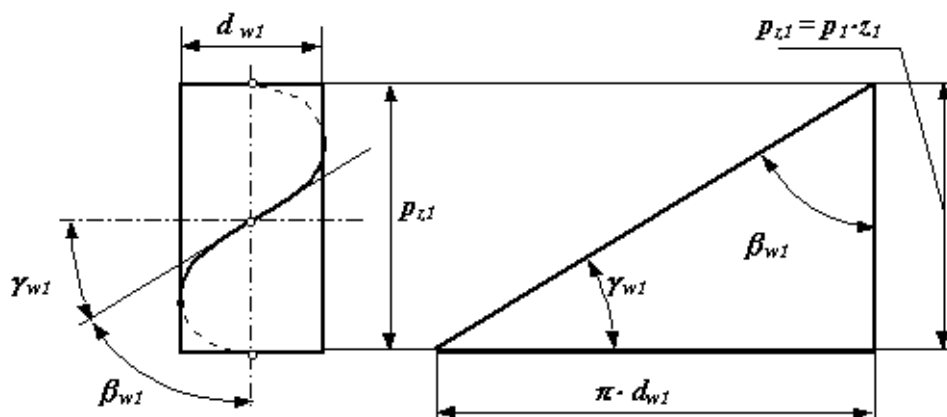


Рис. 3.2.1. Основные параметры червяка.

### 3.2.1. Расчет геометрии зацепления

Исходные данные:  $m$  — модуль осевой,  $q$  — коэффициент диаметра червяка,  $z_1$  — число витков червяка,  $a_w$  — межосевое расстояние,  $x$  — коэффициент смещения червяка,  $u$  — передаточное число.

Осевой модуль  $m$ , коэффициент диаметра червяка  $q$  и межосевое расстояние  $a_w$  выбираются из стандартного ряда. Соотношения исходных данных регламентированы ГОСТ – 2144-76 (в ред. 1992 года).

Параметры инструмента (Рис. 3.6.2):  $h = (h_w + c_1)$  — коэффициент высоты витка,  $h_a$  — коэффициент высоты головки,  $s$  — коэффициент расчетной толщины,  $r_f$  — коэффициент радиуса кривизны переходной кривой,  $c_{1,2} = 0,25 \dots 0,5$ ,  $s = 0,75\text{Чр}$ ,  $r_f = 0,3 \dots 0,45$ .

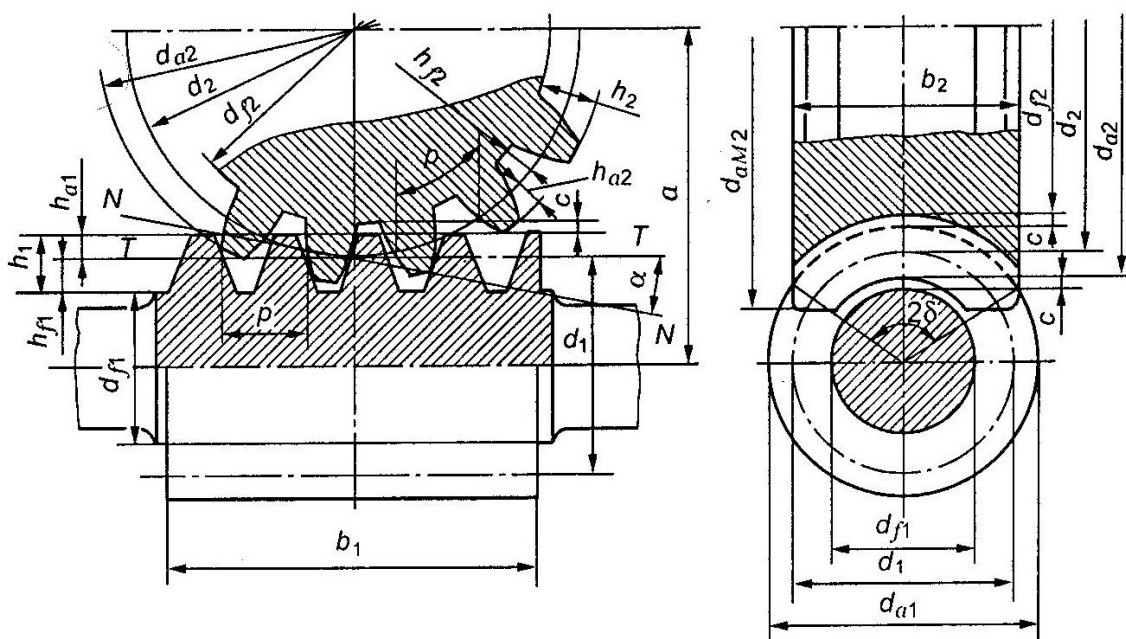


Рис. 3.6.2. Параметры инструмента.

Число зубьев колеса:  $z_2 = z_1 \cdot u$ .

Число витков червяка  $z_1$  зависит от передаточного отношения  $u$ . Рекомендуется  $z_1 = 4$  при  $u = 8 \dots 14$ ,  $z_1 = 2$  при  $u = 16 \dots 28$ ;  $z_1 = 1$  при  $u > 31,5$ . Минимальное число зубьев колеса  $z_{2\min} = 28$ , максимальное —  $z_{2\max} = 80$ .

Коэффициент смещения: (если задано межосевое расстояние)  $x = a_w/m - 0,5 \cdot (z_2 + q)$ .

Межосевое расстояние: (если задан коэффициент смещения)  $a_w = 0,5 \cdot m \cdot (z_2 + q + 2 \cdot x)$ .

Делительные диаметры: червяка —  $d_1 = m \cdot q$ , колеса —  $d_2 = m \cdot z_2$ .

Начальные диаметры: червяка —  $d_{w1} = m \cdot (q + 2 \cdot x)$ , колеса —  $d_{w2} = a_w - d_{w1}$ .

Делительный угол подъема витка червяка:  $\gamma = \arctg(z_1/q)$ .

Начальный угол подъема витка червяка:  $\gamma_w = \arctg(z_1 m/d_{w1})$ .

Основной угол подъема витка червяка: (только для червяков ZI)  
 $\gamma_b = \arccos(\cos \alpha_n \cdot \cos \gamma)$ .

Основной диаметр червяка:  $d_{b1} = z_1 \cdot m / (\tg \gamma_b)$ .

Высота витка червяка:  $h_1 = h \cdot m$ .

Высота головки витка червяка:  $h_{a1} = h_a \cdot m$ .

Диаметры вершин: витков червяка —  $d_{a1} = m \cdot (q + 2 \cdot h_{a1})$ ,  
 зубьев червячного колеса в средней торцовой плоскости —  
 $d_{a2} = m \cdot (z_2 + 2 \cdot h_a + 2 \cdot x)$ .

Диаметры впадин: червяка —  $d_{f1} = m \cdot (q - 2 \cdot h_{f1})$ , червячного  
 колеса —  $d_{f2} = d_{a2} - 2 \cdot (h_a + x) \cdot m$ .

Наибольший диаметр червячного колеса:  $d_{aM2} \leq d_{a2} + 6 \cdot m / (z_1 + 2)$ .

Ширина венца червячного колеса: при  $z_1 \leq 3 \Rightarrow b_2 \leq 0,75 \cdot d_{a1}$ ,  
 при  $z_1 = 4 \Rightarrow b_2 \leq 0,67 \cdot d_{a1}$ .

Длина нарезанной части червяка: (при  $x = 0$ )

при  $z_1 \leq 3 \Rightarrow b_1 \geq (11 + 0,06 \cdot z_2) \cdot m$ ,

при  $z_1 = 4 \Rightarrow b_1 \geq (12,5 + 0,09 \cdot z_2) \cdot m$ .

Угол обхвата:  $2\delta$ , где  $\sin \delta = b_2 / (d_{a1} - 0,5 \cdot m)$ .

### 3.2.2. Пример расчета параметров червячного зацепления

Исходные данные:

|  |                     |
|--|---------------------|
| Осевой модуль червяка                        | $m = 2,5$           |
| Число модулей в делительном диаметре червяка | $q = 8$             |
| Число витков червяка                         | $z_1 = 2$           |
| Число зубьев червячного колеса               | $z_2 = 32$          |
| Межосевое расстояние [2], стр. 614           | $a_w = 50\text{мм}$ |

Рассчитываемые параметры:

|  |   |
|--|---|
| Делительный диаметр                        | $d_1 = m \cdot q = 20\text{мм}$                                 |
| Расчетный шаг червяка                      | $p = 7,8\text{мм}$  |
| Ход витка                                  | $p_h = p_{z1} = 15,708$   |
| Высота головки витка червяка и зуба колеса | $h_{a1} = h_{a2} = m = 2,5\text{мм}$                            |
| Высота ножки витка червяка и зуба колеса   | $h_{f1} = h_{f2} = 1,2 \cdot m = 3\text{мм}$                    |
| Угол профиля витка в осевом сечении        | $40^\circ$  |
| Диаметр вершин витков                      | $d_{a1} = d_1 + 2 \cdot h_{a1} = d_1 + 2 \cdot m = 25\text{мм}$ |

|  |   |
|--|---|
| Диаметр впадин витков                                | $d_{f1} = d_1 - 2 \cdot h_{f1} = d_1 - 2,4 \cdot m = 14\text{мм}$ |
| Делительный угол подъема линии витка                 | $14,0362^\circ$   |
| Коэффициент смещения                                 | $x = a_w/m - 0,5 \cdot (q + z_2) = 0$                             |
| Длина нарезанной части червяка при $x < 0$ и $x = 0$ | $b_1 = m \cdot (10 + 5,5 \cdot  x  + z_1) = 30\text{мм}$          |
| Диаметр начального цилиндра                          | $d_{w1} = m \cdot (q + 2 \cdot x) = 20\text{мм}$                  |
| Угол подъема линии витка на начальном цилиндре       | $14,0362^\circ$   |
| Делительный (начальный) диаметр                      | $d_2 = d_{w2} = m \cdot z_2 = 80\text{мм}$                        |
| Диаметр вершин зубьев                                | $d_{a2} = d_2 + 2 \cdot m \cdot (1 + x) = 85\text{мм}$            |
| Диаметр впадин зубьев                                | $d_{f2} = d_2 - 2 \cdot m \cdot (1,2 - x) = 74\text{мм}$          |
| Наибольший диаметр червячного колеса                 | $d_{ae2} = d_{a2} + 6 \cdot m / (z_1 + 2) = 88,75\text{мм}$       |
| Ширина венца   | $b_2 = 0,355 \cdot a_w = 17,75\text{мм}$                          |
| Угол обхвата   | $48,4^\circ$  |

### 3.2.3. Геометрические показатели качества зацепления

Подрезание зубьев червячного колеса отсутствует, если выполняется условие:  $x \geq h_{a1}^* - (z_2 - \sin^2 \alpha)/2$ .

Заострение зубьев в средней торцовой плоскости отсутствует, если выполняется условие:  $a_w \leq m \cdot (0,55 \cdot z_2 - 0,64 - 0,024 \cdot \alpha) + 0,5 \cdot d_{a1}$ .

Заострение зубьев в средней торцовой плоскости отсутствует, если выполняется условие:  $E = z_2 \cdot (\operatorname{tg} \alpha_{a2} - \operatorname{tg} \alpha) / (2\pi) + 2 \cdot (h_a^* - x - 0,1) / (\pi \cdot \sin 2\alpha) \geq 1,2$ .

### 3.2.4. Способы изготовления

При нарезании зубьев червячных колес, входящих в зацепление с многозаходным червяком, часто используется стандартная однозаходная червячная фреза. Однако при этом зацепление получается не сопряженным, пятно контакта имеет малые размеры, между зубьями требуется увеличенный боковой зазор и, в результате, передача имеет низкую нагрузочную способность и невысокую долговечность.

Для получения точной (сопряженной) передачи можно идти двумя путями: нарезать червячные колеса фрезой, являющейся копией червяка данной пары; нарезать червячные колеса резцом-летучкой при тангенциальной подаче фрезерного суппорта станка, при кото-

ром точно воспроизводится боковая поверхность витков многозаходного червяка. Первый путь в условиях индивидуального производства не приемлем, а второй путь можно использовать при наличии станков с тангенциальной подачей. Если таких моделей станков нет, то можно использовать обычные зубофрезерные станки, сделав у них несложную реконструкцию.

Литые чугунные червяки и червячные колеса в настоящее время почти совершенно не применяются вследствие их низкого коэффициента полезного действия. Червяки нарезаются либо резцами на токарных станках, либо дисковыми фрезами на специальных червячно-фрезерных станках. Для качественных червячных передач применяются шлифованные червяки с твердыми рабочими поверхностями витков. Червячные колеса нарезаются либо червячными фрезами, либо летучками (вращающимися резцами), однако второй способ малопроизводителен. Для достижения полного контакта в зацеплении червяка с червячным колесом при нарезании последнего червячной фрезой необходимо, чтобы рабочая поверхность червяка не отличалась от поверхности исходного инструментального червяка (на котором располагаются режущие кромки фрезы.) Этого можно достигнуть при применении архимедовых и эвольвентных червяков и соответственно спрофилированных фрез. Изготовление эвольвентных червячных фрез требует специального оборудования и может быть освоено лишь на заводах, специализирующихся на производстве червячных передач. Преимущество эвольвентного червячного зацепления заключается в том, что эвольвентные червяки можно шлифовать плоской стороной шлифовального камня. Наибольшим распространением пользуются архимедовы червяки. Шлифование таких червяков производится конусной стороной шлифовального круга на червячно-шлифовальных станках. Для избежания погрешностей, обусловленных фасонной правкой шлифовальных кругов некоторые заводы изготавливают червяки с такой формой рабочих поверхностей витков, которые получают в результате шлифования конусными кругами при прямолинейной их правке и соответственно профилируют червячные фрезы. Червяки для особо качественных передач рекомендуются после шлифования подвергать притирке (с деревянными червячными колесами.)

### 3.3. Выбор материала и особенности расчета червяка и червячного колеса

Высокие скорости скольжения требуют, чтобы материалы червяка и колеса имели низкий коэффициент трения, повышенную износостойкость и пониженную склонность к заеданию. Выполнение червячной пары из однородных материалов не дает желаемых результатов, т. к. оно не обеспечивает достаточной износостойкости и сопротивления заеданию, поэтому червяк и колесо изготавливают из различных материалов.

**Червяки.** Для червяков применяются те же марки сталей, что и для зубчатых колес. Наиболее часто применяются цементированные стали 20Х, 12ХН3А, 18ХГТ, твердость после закалки 57...64 HRC, а также стали 45, 40Х, 40ХН с поверхностной или объемной закалкой до твердости 46...56 HRC. Поверхность витков обязательно шлифуют и полируют, для обеспечения наибольшей стойкости зубьев червячного колеса против изнашивания и повышения КПД.

**Червячные колеса.** Червячные колеса для экономии изготавливаются составными: венец из антифрикционных, относительно дорогих и малопрочных материалов (бронзы), а колесный центр из стали, при небольших нагрузках — из чугуна. Марка бронзы выбирается в зависимости от скорости скольжения  $v_s$ , значение которой ориентировочно определяется по приближенной зависимости (м/с):  $v_s = 4,3 \cdot 10^{-4} \cdot n_1 \cdot (M_2)^{1/3}$ , где  $M_2$  — момент на валу червячного колеса (Н·м),  $n_1$  — частота вращения червяка (об/мин).

При  $v_s = 5 \dots 25$  м/с применяют оловянные бронзы марок Бр010Н1Ф1, Бр010Ф1.

При  $v_s = 2 \dots 5$  м/с применяют более дешевые безоловянные бронзы марок БрА9ЖЗЛ, БрА10Ж4Н4Л, а также латунь марки ЛЦ23А6ЖЗМц2.

Чем выше содержание олова в бронзе, тем она дороже, но тем выше и сопротивление заеданию.

#### Расчет прочности зубьев червячного колеса.

Усилия в зацеплении:

- окружная сила на червяке и осевая на колесе (Н):  $F_{t1} = F_{a2} = 2 \cdot M_1/d_1$ ;
- окружная сила на колесе и осевая на червяке (Н):  $F_{t2} = F_{a1} = 2 \cdot M_2/d_2$ ;
- радиальная сила на червяке и колесе (Н):  $F_{r1} = F_{r2} = F_{t2} \cdot \operatorname{tg} \alpha$ , где  $\alpha = 20^\circ$  — угол зацепления.

Расчетные коэффициенты:



- коэффициент нагрузки  $KH = KF = K_\beta \cdot K_v$ , где  $K_\beta$  — коэффициент концентрации нагрузки,  $K_v$  — коэффициент динамической нагрузки.

Хорошая прирабатываемость зубьев колеса уменьшает значение  $K_\beta$ . При постоянной внешней нагрузке  $K_\beta = 1$ .

Червячные передачи работают плавно и бесшумно, поэтому динамические нагрузки, возникающие в зацеплении, невелики и зависят от окружной скорости колеса  $v_2$ .

При  $v_2 \leq 3 \text{ м/с}$  принимается, что  $K_v = 1$ . При  $v_2 > 3 \text{ м/с}$  принимается, что  $K_v = 1,1 \dots 1,3$ .

Для определения  $K_v$ , находится окружная скорость червячного колеса (мм/с)  $v_2 = \omega_1 \cdot d_2/2 \cdot u$ , где  $\omega_1$  — угловая скорость червяка ( $\text{с}^{-1}$ ).

#### Проверка контактной прочности зубьев.

Если фактическая скорость скольжения  $v_{сф}$  отличается от ориентировочно принятой  $v_s$ , то необходимо уточнить значение  $[\sigma_{H2}]$ , используя значение  $v_{сф}$ . Расчетное контактное напряжение на поверхности зубьев колеса и условие прочности имеет вид

$$\sigma_{H2} = 170 \cdot q/z_2 \cdot \sqrt{\left(\left((z_2/q) + 1\right)/\alpha_w\right)^3 \cdot M_2 \cdot K_H} \leq [\sigma_{H2}].$$

При невыполнении условия прочности следует увеличить  $\alpha_w$ , выбрать другой материал колеса с бóльшим  $[\sigma_{H2}]$  и т. д.

Проверка прочности зубьев червячного колеса на изгиб: напряжение изгиба и условие прочности ( $\text{Н/мм}^2$ )

$$\sigma_{F2} = 0,7 \cdot Y_{F2} \cdot F_{t2} \cdot K_F / (z_2 \cdot m) \leq [\sigma_{F2}].$$

где  $Y_{F2}$  — коэффициент формы зуба червячного колеса, определяется в зависимости от эквивалентного числа зубьев колеса  $z_{v2} = z_2 / \cos^3 \gamma$  при помощи таблицы

|          |     |      |      |      |      |      |      |      |      |      |      |
|----------|-----|------|------|------|------|------|------|------|------|------|------|
| $z_{v2}$ | 28  | 30   | 32   | 35   | 37   | 40   | 45   | 50   | 60   | 80   | 100  |
| $Y_{F2}$ | 1,8 | 1,76 | 1,71 | 1,64 | 1,61 | 1,55 | 1,48 | 1,45 | 1,40 | 1,34 | 1,30 |

При невыполнении условия прочности действия те же, что и при проверке контактной прочности.

### **3.4. Конструкции червячных редукторов**

Среди редукторов общего назначения червячные редукторы (Рис. 3.4.1) имеют достаточно широкое применение (до 30%). Наибольшее распространение имеют одноступенчатые червячные редукторы с диапазоном передаточных чисел  $u = 8 \dots 63$ .

Кинематические схемы одноступенчатых червячных редукторов представлены на Рис. 3.4.2.

При больших передаточных числах применяются двухступенчатые червячные редукторы или комбинированные, состоящие из червячной и зубчатой передач.

Червячные редукторы выполняются со следующими вариантами расположения червяка 1 и червячного колеса 2: червяк над колесом (Рис. 3.4.2, а), червяк под колесом (Рис. 3.4.2, б), с вертикальным расположением оси червяка (Рис. 3.4.2, в), с вертикальным расположением оси колеса (Рис. 3.4.2, г). Выбор схемы редуктора определяет компоновка агрегатов привода.

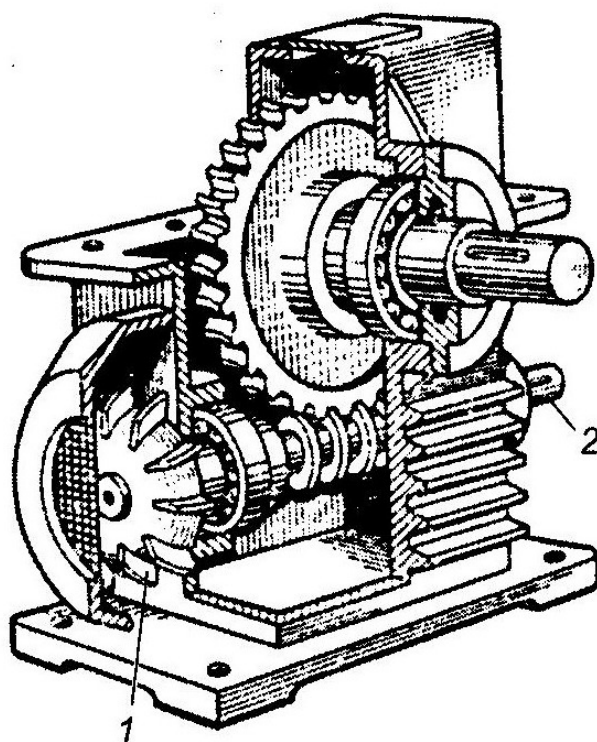


Рис. 3.8.1. Червячный редуктор.

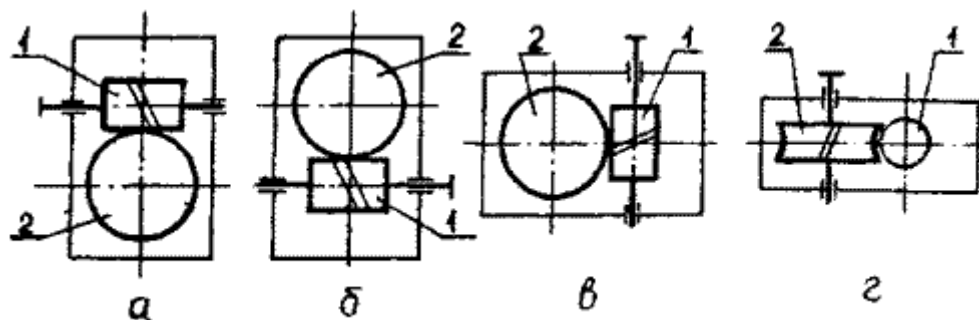


Рис. 3.8.2. Варианты расположения червяка 1 и червячного колеса 2.

В одноступенчатых редукторах корпус, как правило, делается неразъемным. В редукторах с  $\alpha_w > 250^\circ$  корпус выполняют с плоскостью разъема по оси вала червячного редуктора. В неразъемных корпусах для возможности сборки на боковых стенках выполняют отвер-

ствия диаметром, несколько большим наибольшего диаметра  $d_a$  червячного колеса. Чтобы увеличить жесткость корпуса и создать условия для крепления привертных крышек в местах расположения подшипников, предусмотрены бобышки и ребра жесткости.

Крышки и корпус редуктора — литые, изготовлены обычно из чугуна или из алюминиевых сплавов. Для увеличения теплоотдачи и понижения температуры масла на боковых стенках корпуса и крышках выполняются ребра.

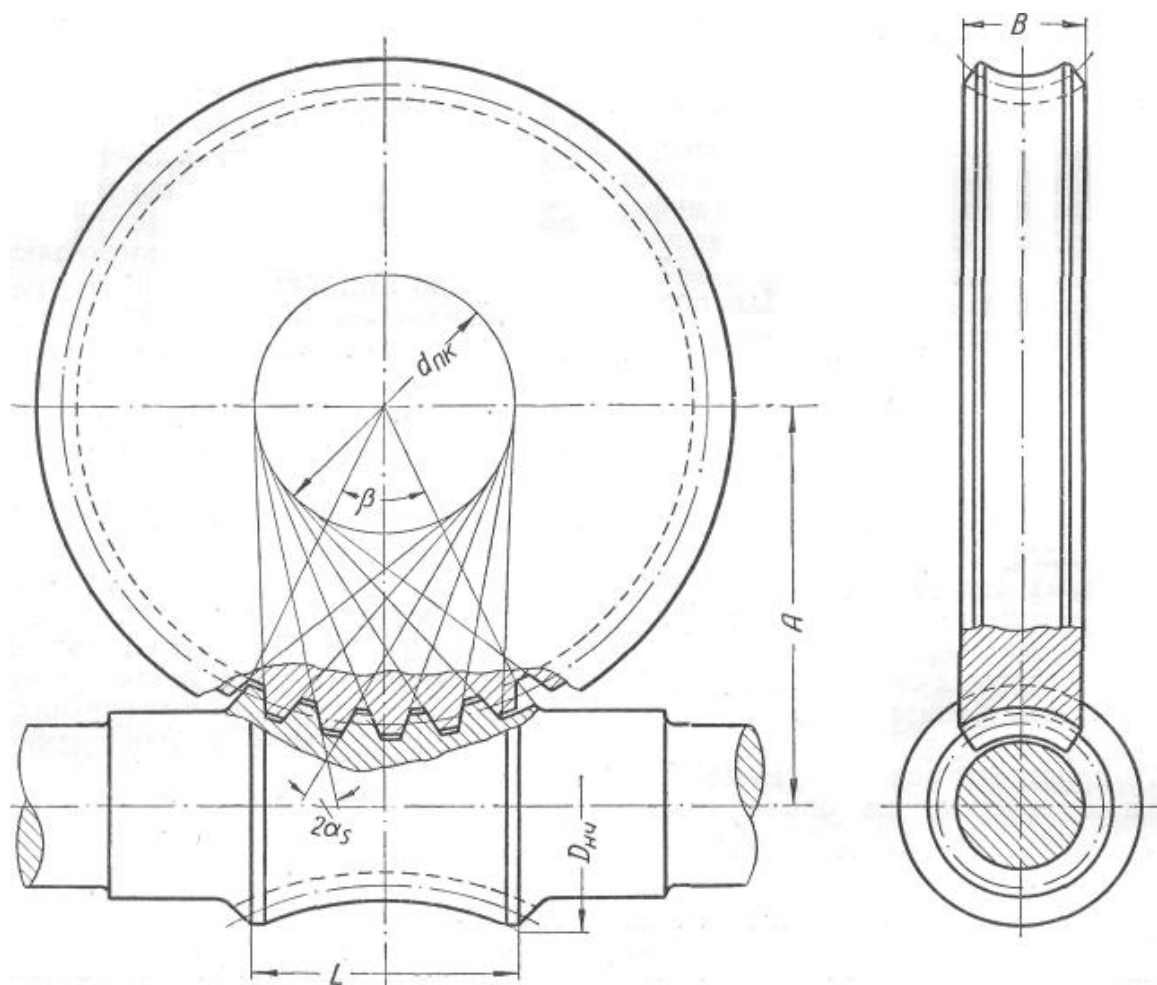


Рис. 3.4.3. Глобоидная передача.

Передачи с винтовым профилем и глобоидные передачи. Глобоидная передача (Рис. 3.4.2), разновидность червячной передачи, в которой образующая червяка имеет глобоидную (вогнутую) форму. Рабочая поверхность глобоидного червяка образуется вращением вокруг его оси  $O_1O_2$  дуги окружности диаметром  $d$ , ограниченной углом обхвата  $2^\circ$ . Профили образуются прямыми, касательными к профилирующей окружности  $d_0$ .

Глобоидная передача — зубчатовинтовая передача, получает всё большее распространение благодаря высокой нагрузочной способности, которая обусловлена одновременным зацеплением большо-


го числа зубьев (4 – 7) и благоприятным расположением линий контакта. При работе глобоидной передачи создаётся жидкостный или полужидкостный режим трения, при котором контактные поверхности зубьев колеса и витков червяка полностью или в большей части разделены устойчивым слоем смазки. Средние и мощные глобоидные передачи при одинаковых размерах с обычной червячной передачей способны передавать в 3 – 5 раз бóльшую мощность и, наоборот, при той же передаваемой мощности размеры и масса глобоидной передачи оказываются значительно меньшими. К недостаткам глобоидной передачи относятся: более сложное изготовление и сборка, чем обычных червячных передач; работа в напряжённом тепловом режиме и необходимость в искусственном охлаждении. Наиболее эффективно применение глобоидной передачи для работы с большими нагрузками в установившемся режиме, а также при необходимости создания компактного и лёгкого оборудования (например, в транспортных и горных машинах, самолётах и т. п.).


Передачи с винтовым профилем. Пара косозубых колёс (в частном случае одно из них может быть прямозубым), передающих вращение между непараллельными и непересекающимися валами, называется винтовой (точнее, винтовой цилиндрической, в отличие от винтовой конической — гипоидной) зубчатой передачей. Винтовая зубчатая передача (за исключением винтовой передачи, осуществляемой шестерней и рейкой) теоретически имеет контакт в одной точке, практически же вследствие износа и деформации зубьев контакт распространяется на некоторую очень небольшую поверхность. В связи с этим винтовые передачи могут передавать лишь малые нагрузки, и для надёжной работы они должны изготавливаться из материалов, образующих противозадирную комбинацию.


При межосевом угле 5 – 90° рекомендуется применять вместо винтовых передач червячные с многозаходными червяками, обеспечивающие надёжную работу при значительно более высоких нагрузках.

### **3.5. Методика создания модели червячного вала в среде системы трехмерного моделирования CATIA V5**

Все необходимые данные для моделирования червяка получены с чертежа.

Для того чтобы начать моделирование выберите “File → New” (или щелкните по иконке “New” ) , в появившемся диалоговом окне “List of Types” выберите “Part”. Откроется новый документ.

Для удобства моделирования вначале создадим систему координат. Щелкните по иконке “Axis System” , в появившемся окне “Axis System Definition” нажмите кнопку “OK”.

Выберите плоскость XY и щелкните по иконке “Sketcher” , чтобы перейти в режим эскизирования. По полученным с чертежа данным построим скетч, как показано на Рис. 3.9.1.

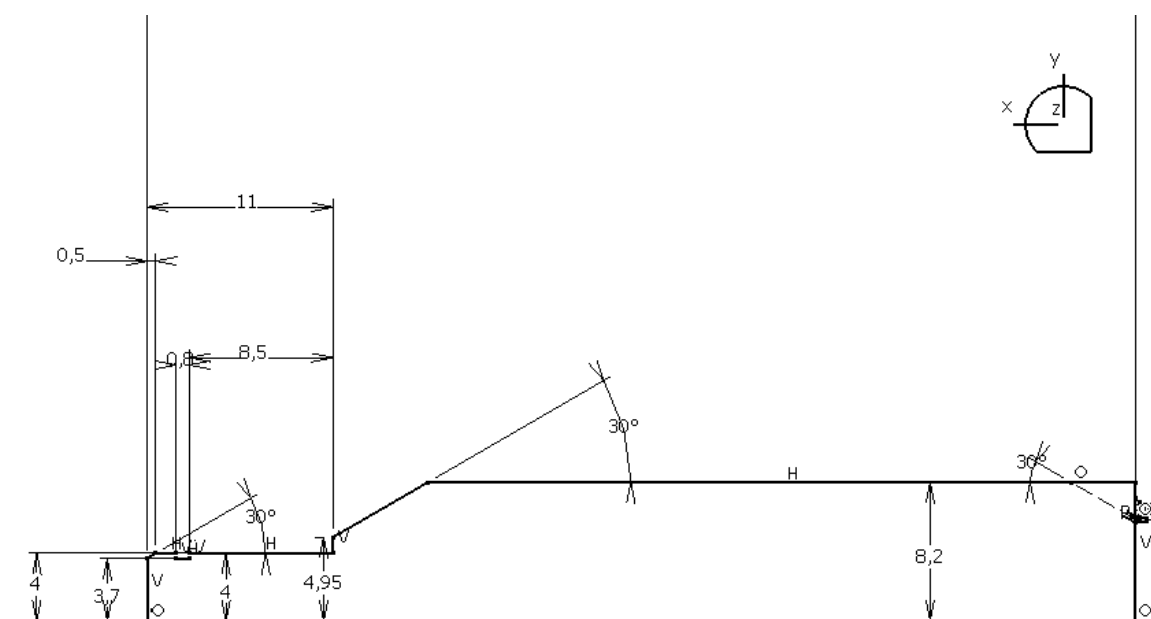






Рис. 3.9.1. Контур червяка.

Так как основой для дальнейшего моделирования червяка является тело вращения, то в качестве эскиза мы создаем лишь половину контура. В данном эскизе моделируется не весь вал, а только та его часть, на которой будет находиться червяк. Это сделано для удобства дальнейшего моделирования. После того, как скетч построен, нажмите на иконку “Exit Workbench” .

Нажмите на иконку “Shaft”  на панели “Sketch-Based Features”. Появится диалоговое окно “Shaft Definition” (Рис. 3.9.2). В поле “First Angle” введите значение 360deg, в поле “Selection” — Sketch.1. Нажмите “OK”.

Выберите торцовую плоскость, с которой начинаются витки червяка. Щелкните по иконке “Sketcher”  и перейдите в режим эскизирования.

Построим три замкнутых контура. Боковые стороны контуров являются эвольвентным профилем, а верхняя и нижняя — дугами окружностей вершин и впадин соответственно. Диаметр впадин равен

9,9, диаметр вершин — 16,4, а делительный диаметр составляет 13,5. Как построить эвольвентный профиль смотрите в разделе 3.6. После того, как скетч построен, нажмите на иконку “Exit Workbench” .

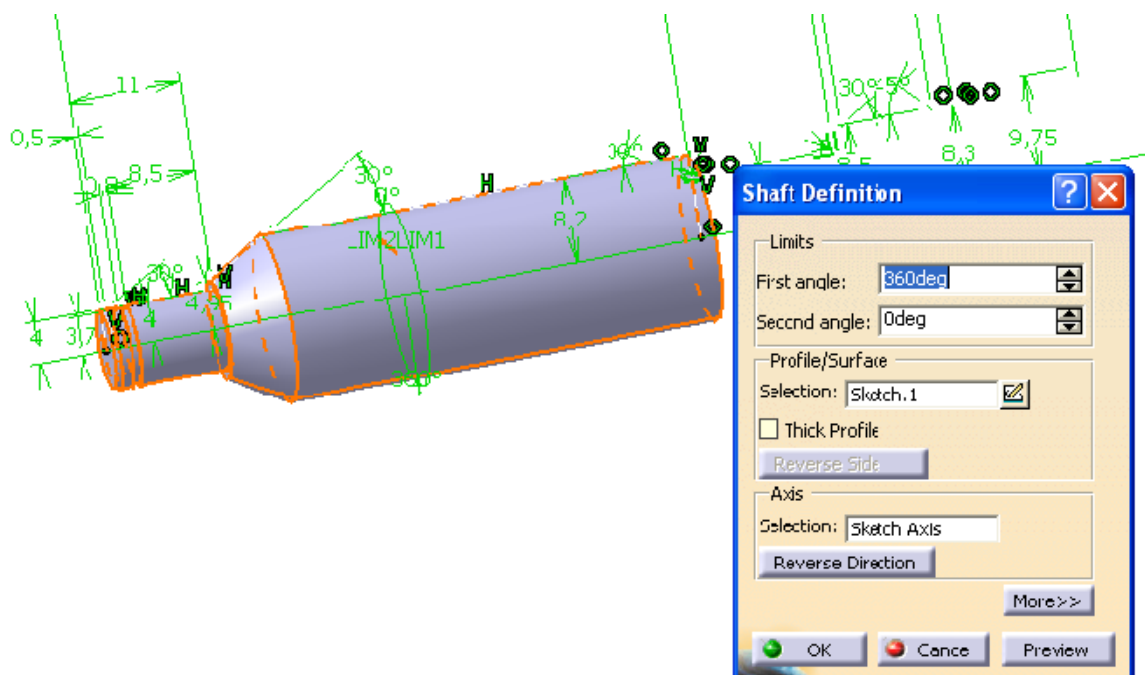





Рис. 3.9.2. Предварительный просмотр результатов команды “Shaft”.


Снова выберите торцовую плоскость, как и для предыдущего скетча. Щелкните по иконке “Sketcher”  и перейдите в режим эскизирования.


При помощи иконки “Circle”  на панели “Profile” создайте окружность диаметром 13,5 с центром в начале системы координат скетча, которая будет являться начальной точкой (“Starting Point”) для спирали (“Helix”). В качестве направляющей кривой для образования профиля червяка мы используем спираль (“Helix”).

Для этого щелкните “Start → Shape → Generative Shape Design”.

На панели “Wireframe” выберите вкладку “Helix” . Появится диалоговое окно “Helix Curve Definition” (Рис. 3.9.3).

В поле “Starting Point” выберите Sketch.3. Оставшиеся параметры введите, как показано в диалоговом окне “Helix Curve Definition”.

На двух предыдущих шагах мы получили профиль и направляющую для создания профиля червяка. Воспользуемся командой “Slot”  для того, чтобы получить профиль червяка. В появившемся окне “Slot Definition” в меню “Profile” укажите Sketch.4, а в меню “Curve” — Helix.3, “Profile Control” — Keep Angle. В результате мы получим показанное на Рис. 3.9.4 изображение.

Скруглим впадины витков червяка. Для этого воспользуемся командой “Edge Fillet” . В появившемся окне “Edge Fillet Definition” в поле “Radius” введите значение 0,5мм; в поле “Object(s) to Fillet” выберите шесть ребер, как показано на Рис. 3.9.5, в поле “Setback Distance” укажите 10мм. Шесть ребер получается, потому что мы выбираем ребро с каждой стороны витка, а у нас червяк трехвитковый — по построению мы использовали в качестве профиля три эвольвентных зуба. Таким образом, получается 6 ребер.

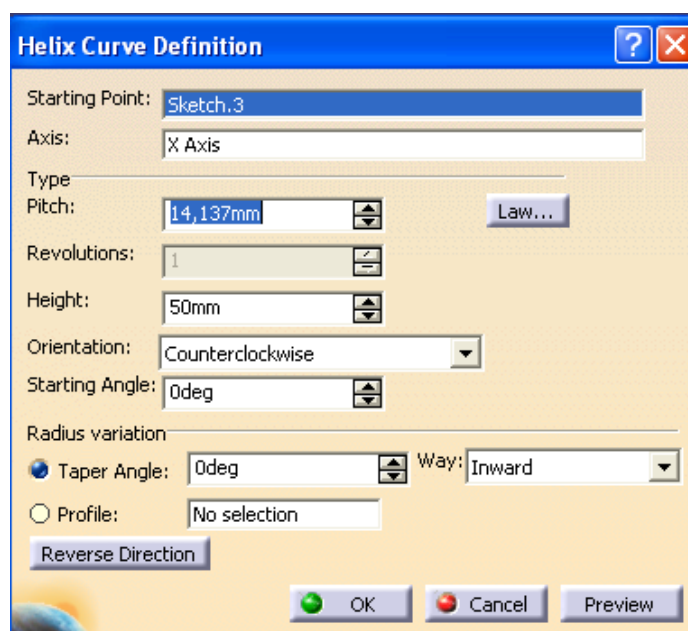


Рис. 3.9.3. Диалоговое окно команды “Helix”.

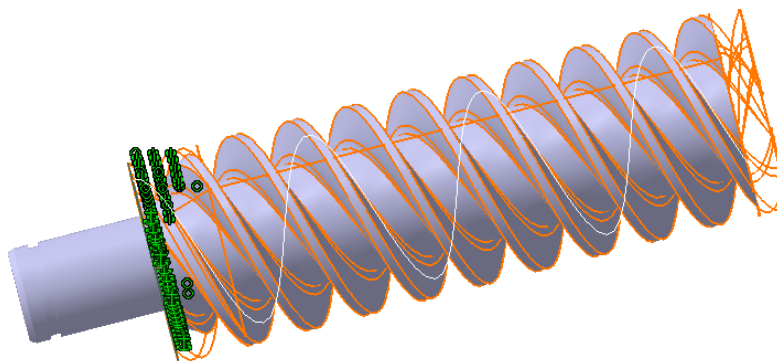




Рис. 3.9.4. Результат применения команды “Slot”.





После того, как скетч построен, нажмите на иконку “Exit Workbench” .

Далее нажмите на иконку “Shaft”  на панели “Sketch-Based Features”. Появится диалоговое окно “Shaft Definition”. В поле “First Angle” введите значение 360deg, в поле “Selection” окна “Profile/Surface” — Sketch.5, в поле “Selection” окна “Axis” — Sketch Axis. Нажмите “OK”. На Рис. 3.9.8 показан результат.

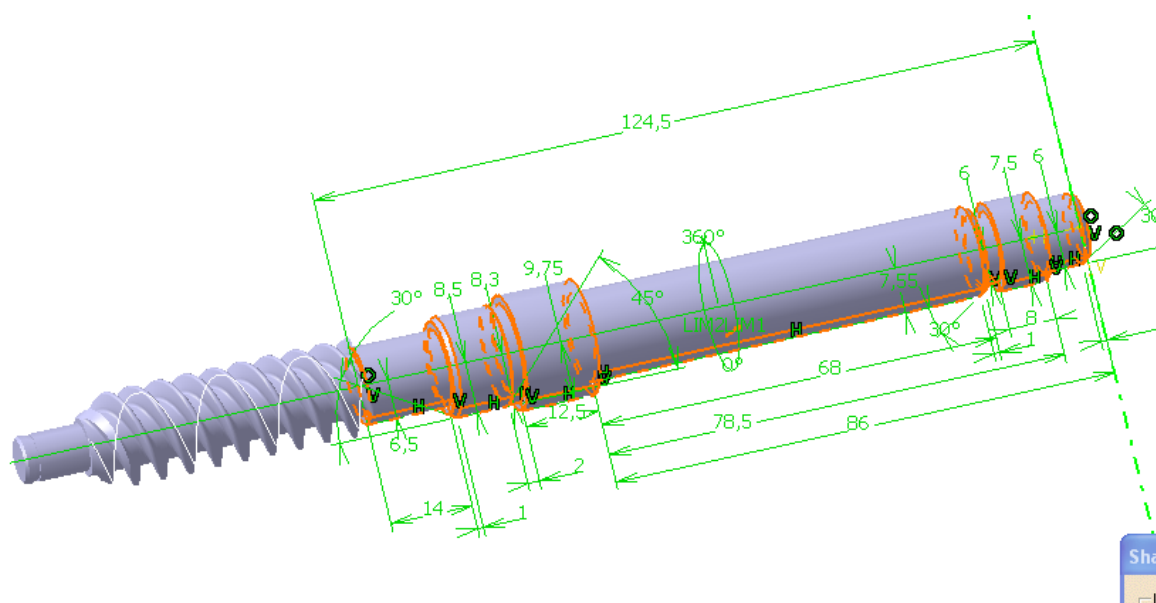



Рис. 3.9.8. Результат применения команды “Shaft”.

Нажмите на иконку “Chamfer”  и сделайте фаску на двух ребрах указанных на Рис. 3.9.9. Введите параметры как указано в диалоговом окне “Chamfer Definition”. Нажмите “OK”.

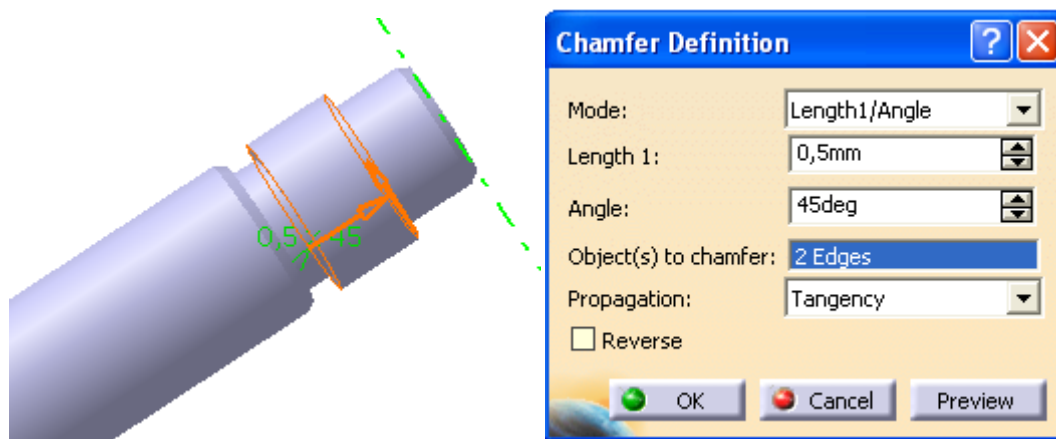




Рис. 3.9.9. Создание фаски.

Нажмите на иконку “Thread/Tap Definition”  в меню “Dress-Up Features”. В диалоговом окне “Thread/Tap Definition” укажите следующие данные:

- “Lateral Face” — Face.4;

- “Limit Face” — Face.3;
- “Type” — No Standard;
- “Thread Diameter” — 12мм;
- “Thread Depth” — 7.5мм;
- “Pitch” — 1мм.

Нажмите на иконку “Edge Fillet” . В появившемся окне “Edge Fillet Definition” в поле “Radius” введите значение 0,2мм, в поле “Object(s) to Fillet” выберите два ребра, как показано на Рис. 3.9.10. Нажмите “OK”.

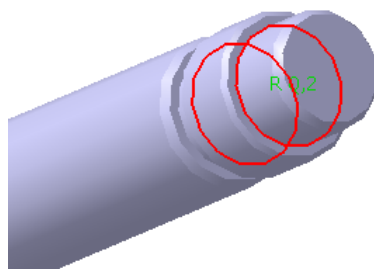




Рис. 3.9.10. Построение скруглений.

Выберите поверхность так, как показано на Рис. 3.9.11. Щелкните по иконке “Sketcher”  и перейдите в режим эскизирования. При помощи меню “Profile” и панели инструментов “Constraint” постройте эскиз. После того, как скетч построен, нажмите на иконку “Exit Workbench” .

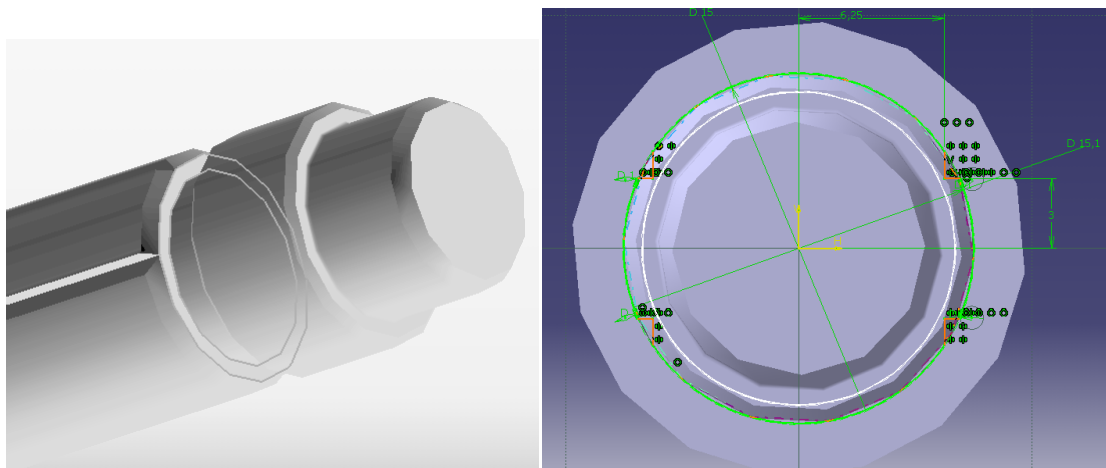






Рис. 3.9.11. Создание эскиза пазов.

Нажмите на иконку “Pocket”  в меню “Sketch-Based Features”. Диалоговое окно “Pocket Definition” заполните, как показано на Рис. 3.9.12. Таким образом, мы получим пазы на валу червяка (см. Рис. 3.9.12).

Скруглим ребра на поверхности вала. Для этого нажмите на иконку “Edge Fillet”  и выберите два ребра на поверхности червячного вала, как это показано на Рис. 3.9.13. Величину радиуса введите равной 0,2мм.

Витки червяка по краям вала должны быть скошены под углом  $\beta = 30^\circ$ . Для этого мы построим методом вращения поверхность и удалим ту часть витков, которая попала в пересечение с построенной поверхностью. Для этого щелкните по иконке “Sketcher”  и перейдите в режим эскизирования. При помощи инструментария данного подменю постройте скетч, как показано на Рис. 3.9.14. Для выхода нажмите на иконку “Exit Workbench” .

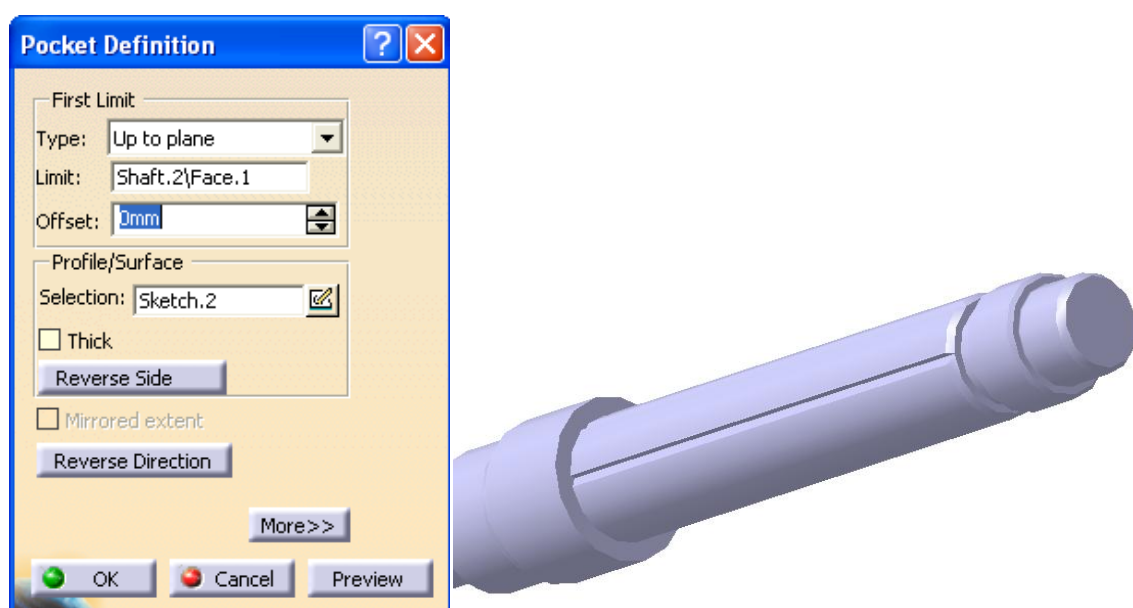


Рис. 3.9.12. Создание пазов.

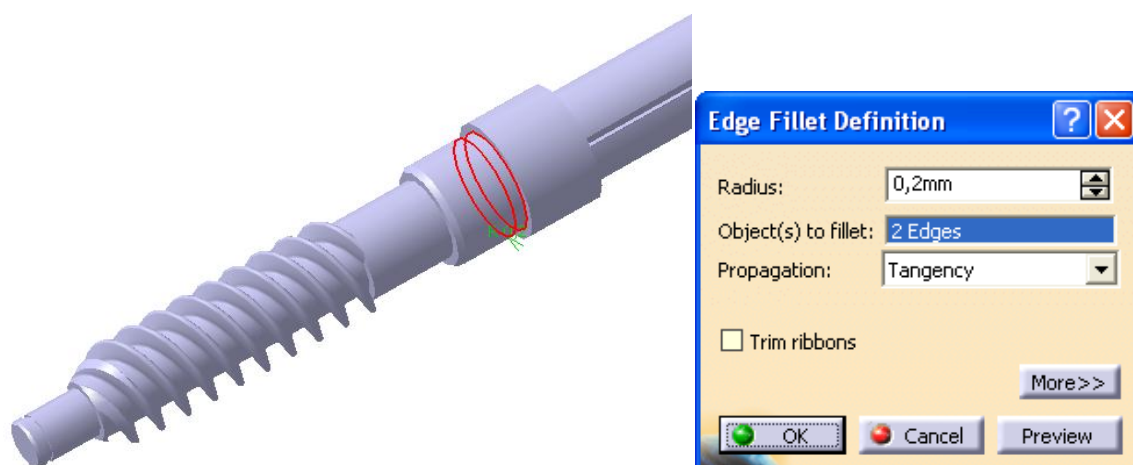


Рис. 3.9.13. Создание скруглений.

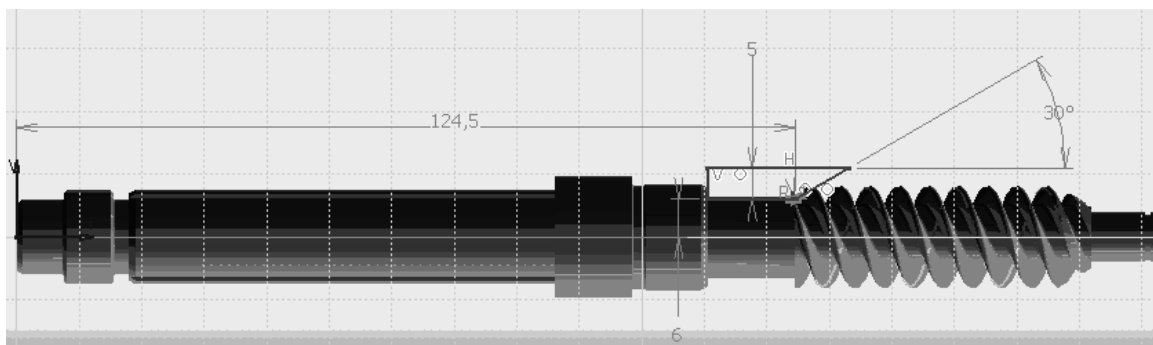



Рис. 3.9.14. Создание эскиза.

Щелкните по иконке “Groove Definition”  на панели инструментов “Sketch-Based Features”. В поле “First Angle” введите значение 360deg, в качестве профиля выберите Sketch. 6. Нажмите “OK” (Рис. 3.9.15). Таким образом, построенная поверхность срезала поверхность витков червяка по краю под углом 30 градусов.

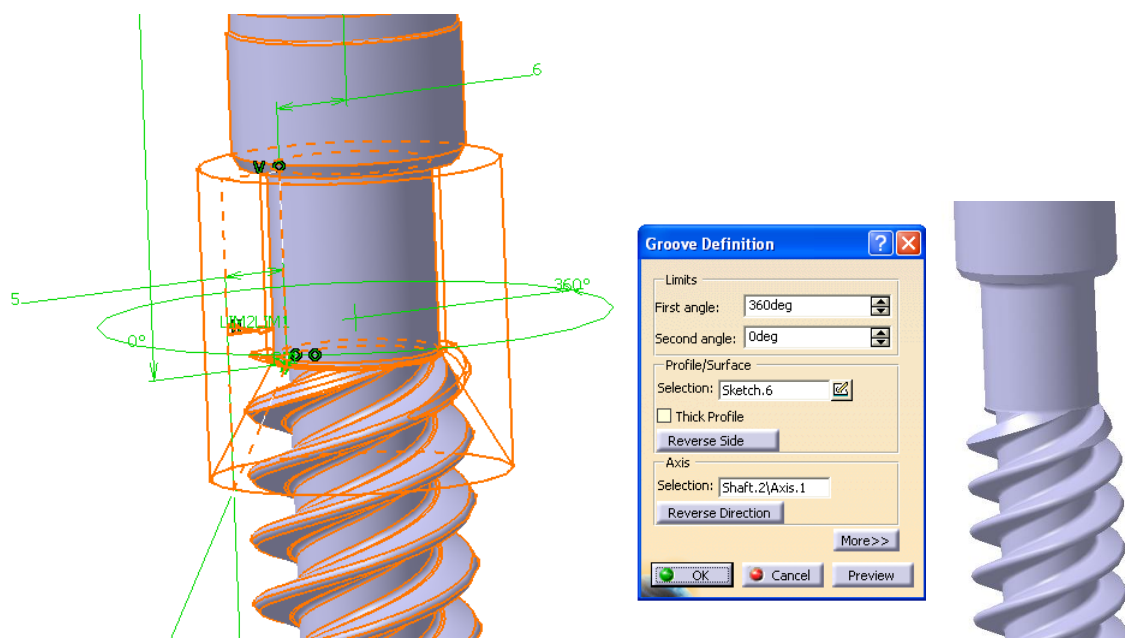




Рис. 3.9.15. Результат применения команды “Groove Definition”.

Скруглим ребро на поверхности вала. Для этого нажмите на иконку “Edge Fillet”  и выберите ребро на поверхности червячного вала, как показано на Рис. 3.9.16. Величину радиуса введите равной 1мм. Нажмите “OK”.

Модель червячного вала готова. Осталось добавить материал. Для этого нажмите на иконку “Apply Material”  и выберите из появившегося каталога материалов Сталь. В поле название введите: Сталь 40Х ГОСТ 4543 – 71. Нажмите “OK”. Построенная модель червячного вала выглядит так, как показано на Рис. 3.9.17.

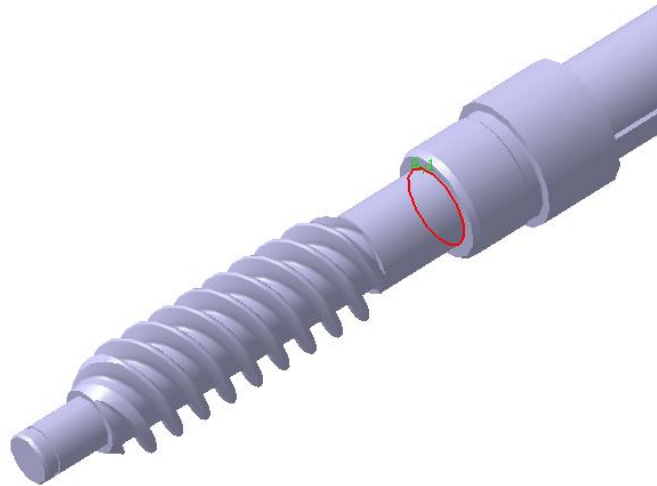
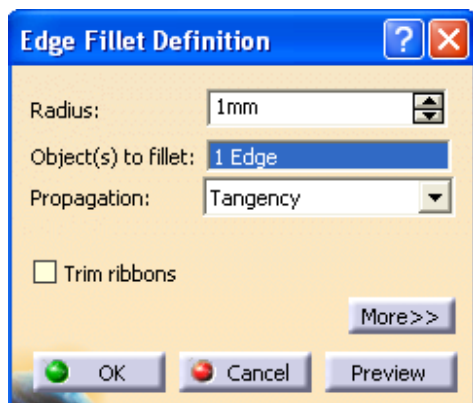


Рис. 3.9.16. Построение скругления.

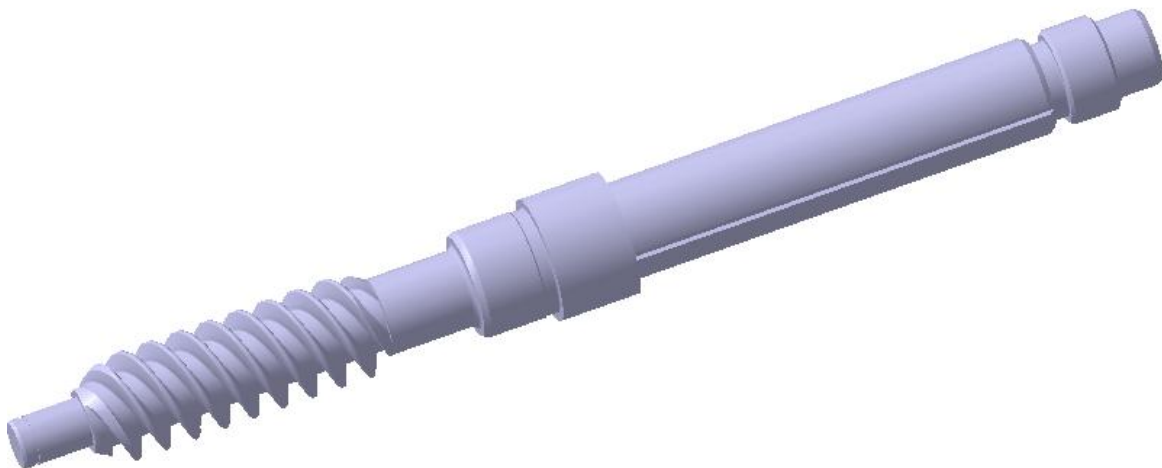






Рис. 3.9.17. Модель червячного вала.



### 3.6. Методика создания модели червячного колеса в среде системы трехмерного моделирования CATIA V5


Приступим к формированию профиля зуба червячного колеса. Выберите плоскость  $XZ$ , щелкните по иконке “Sketcher”  и перейдите в режим эскизирования.



При помощи иконки “Circle”  постройте в качестве вспомогательных (пунктирной линией) четыре окружности с центром в начале координат диаметрами 78,9мм, 80,697мм, 82,5мм и 88мм.


Нажмите на иконку “Line”  и постройте вспомогательную ось под углом в 2,367 градусов к вертикальной оси “V”. От построенной вспомогательной оси (также пунктирной линией) постройте пять вспомогательных линий из центра системы координат под углом 5, 10, 15, 20 и 25 градусов. Они послужат нам для построения эвольвентного профиля, который будет являться частью профиля межзубной впадины колеса.


На пересечении построенных линий и окружности диаметром 80,697мм при помощи иконки “Point”  постройте пять точек. Из каждой построенной точки проведите касательную линию по направлению к вспомогательной оси (как при построении эвольвентного профиля) длиной соответственно 3,521мм — для 5 градусов, 7,042мм — для 10 градусов, 10,563мм — для 15 градусов, 14,084мм — для 20 градусов, 17,605мм — для 25 градусов. На окончании построенных линий получим пять точек и еще одну точку на пересечении вспомогательной оси и окружности диаметром 80,697мм.

При помощи сплайна соедините полученные шесть точек (иконка “Spline” ) Часть сплайна, выступающую за пределы окружности диаметром 85,5мм обрежьте, воспользовавшись иконкой “Trim” .

Получившуюся часть эвольвентного профиля симметрично отразите относительно вспомогательной оси. Для этого воспользуйтесь иконкой “Symmetry”  на панели “Operation”. Симметричная часть эвольвенты будет отмечена как вспомогательная геометрия пунктирной линией.


Далее из точки пересечения основного эвольвентного профиля и окружности диаметром 80,697мм постройте вспомогательную линию, касательную к окружности, которая послужит нам в качестве оси симметрии. Для этого воспользуйтесь иконками “Line”  и “Constraints Defined in Dialog Box”  (выберите построенную линию и окружность одновременно и нажмите на позиции “Tangency”).


Выберите вспомогательный эвольвентный профиль и, воспользовавшись иконкой “Symmetry”  на панели “Operation”, отразите его симметрично относительно построенной в предыдущем пункте оси.



Далее полученную в предыдущем пункте часть эвольвентного профиля отразите симметрично вертикальной оси “V”, воспользовавшись иконкой “Symmetry”  на панели “Operation”. Основной эвольвентный профиль также отразите относительно вертикальной оси “V”.

Постройте две окружности в качестве вспомогательных диаметрами 78,9мм и 85,5мм. Обрежьте все части построенного в предыдущих пунктах эвольвентного профиля, которые лежат выше окруж-



ности диаметром 85,5мм и ниже окружности диаметром 78,9мм. Все части эвольвентного профиля, заключенные между двумя данными окружностями выделите как основную геометрию, воспользовавшись иконкой “Construction/Standard Element” .

Таким образом, мы получили почти готовый профиль зуба червячного колеса. Обрежьте части построенных в предыдущем пункте окружностей, которые выступают за пределы эвольвентного профиля, воспользовавшись иконкой “Trim” . Профиль зуба готов (см. Рис. 3.10.1).

Построим два вспомогательных скетча Sketch.3 и Sketch.4, которые необходимы нам для создания винтовой линии. Выберите плоскость XZ, щелкните по иконке “Sketcher”  и перейдите в режим эскизирования. Постройте скетч, как показано на Рис. 3.10.2, который будет являться начальной точкой при создании винтовой линии. Нажмите “OK” для выхода. Далее снова нажмите на иконку “Sketcher”  и перейдите в режим эскизирования. Постройте второй скетч, как показано на рисунке Рис. 3.10.3, который будет являться осью при создании винтовой линии. Нажмите “OK” для выхода.

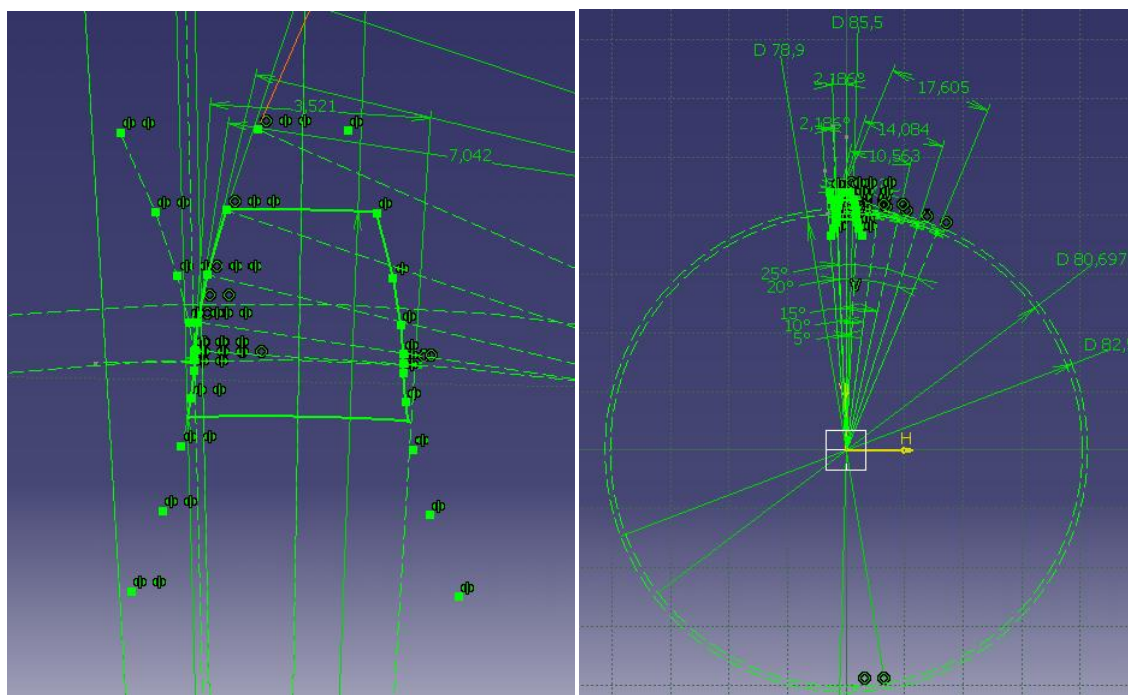


Рис. 3.10.1. Профиль зуба червячного колеса.

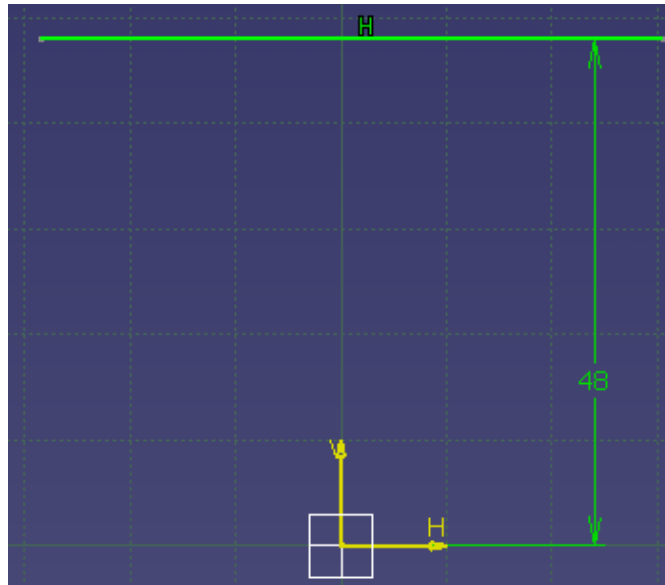


Рис. 3.10.2. Начальная точка винтовой линии.

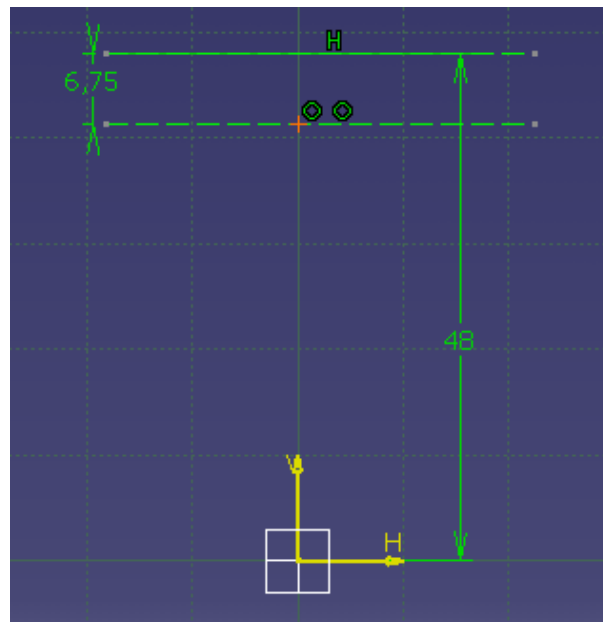




Рис. 3.10.3. Ось винтовой линии.

Перейдите в модуль поверхностного моделирования: “Start → Shape → Generative Shape Design”. Постройте винтовую линию, которая послужит нам направляющей при создании твердотельной модели зуба. Для этого нажмите иконку “Helix”  в меню “Wireframe”. Меню “Helix Curve Definition” заполните так, как показано на Рис. 3.10.4. Нажмите “OK”. Таким образом, мы получили Helix.1.

Объедините построенные участки винтовой линии Helix.1 и Helix.2. Для этого нажмите на иконке “Join”  во вкладке “Insert → Operation”. В поле “Elements To Join” диалогового окна “Join Definition” добавьте Helix.1 и Helix.2, щелкнув на них в дереве модели. Нажмите “OK”.



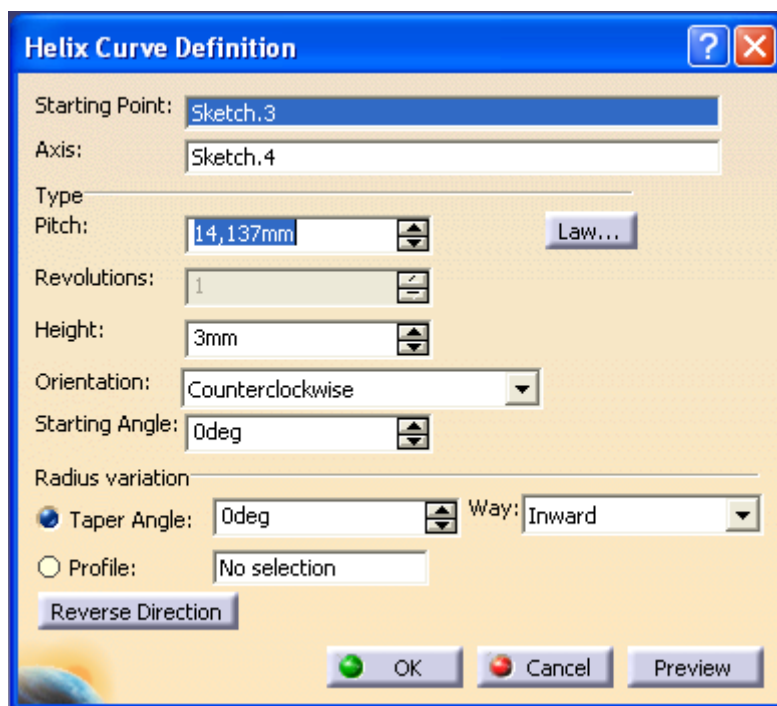



Рис. 3.10.4. Диалоговое окно команды “Helix”.

Протяните построенный контур зуба червячного колеса по винтовой линии. Для этого щелкните на иконке “Rib”  в подменю “Sketch-Based Features”. Окно “Rib Definition” заполните, как показано на Рис. 3.10.5. Нажмите “OK”. Мы получили профиль зуба червячного колеса.

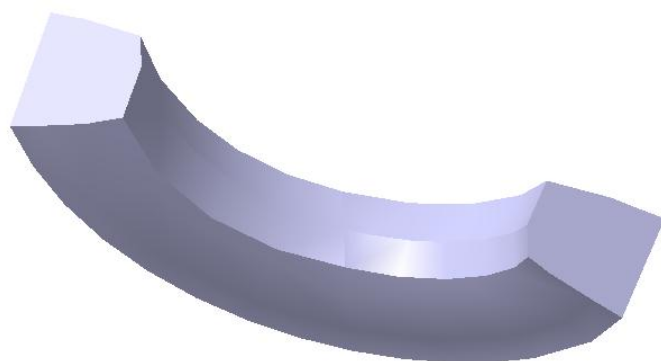
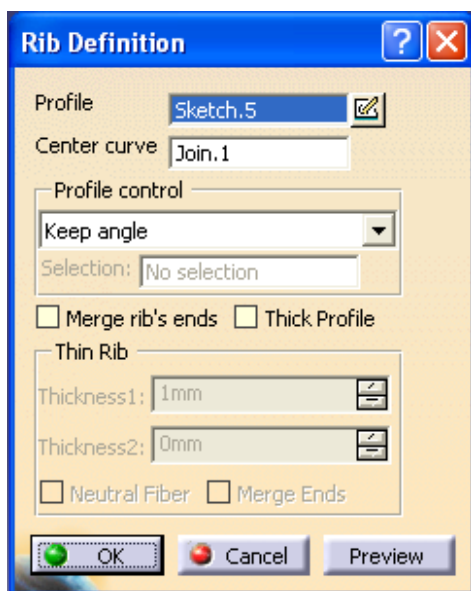



Рис. 3.10.5. Построение зуба червячного колеса.

Мы создали один зуб. По исходным данным у нас 55 зубьев. Для того чтобы создать массив зубьев воспользуемся командой “Circular Pattern”. Для этого нажмите на иконку “Circular Pattern”  в меню “In-

sert → Transformation Features”. Появившееся диалоговое окно “Circular Pattern Definition” заполните так, как показано на Рис. 3.10.6. В качестве параметров мы выбираем число элементов и общий угол, на который производится копирование (“Instance(s) & total angle”). Число элементов (зубьев) равно 55, общий угол — 360°. В качестве объекта для копирования выбираем построенный профиль зуба (Rib.1). После того, как все выбрано, нажмите “OK”.

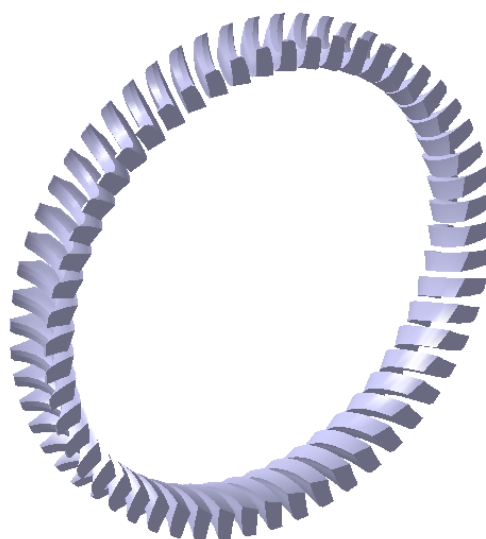
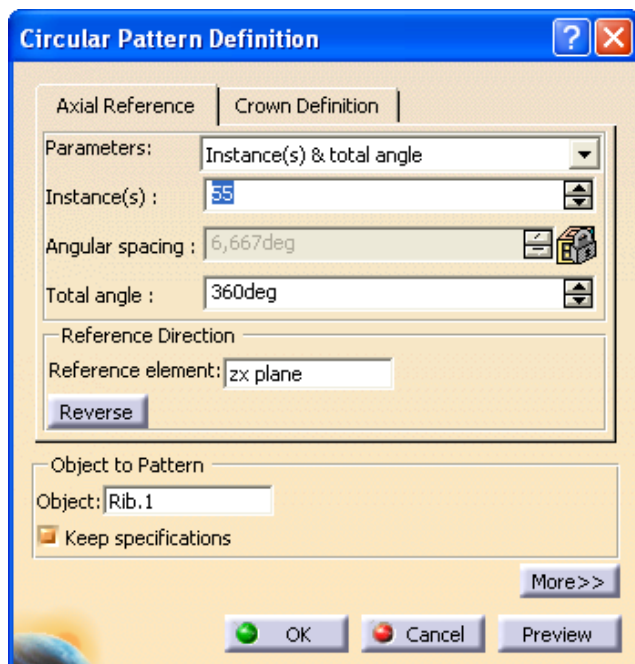


Рис. 3.10.6. Построение массива зубьев червячного колеса.

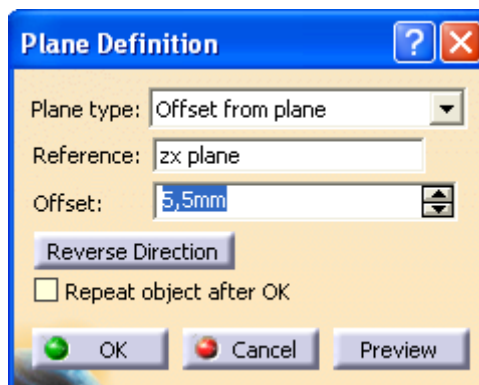




Рис. 3.10.7. Диалоговое окно команды “Plane”.

Создадим две вспомогательные плоскости. Для этого щелкните на иконке “Plane”  в подменю “Wireframe”. Диалоговое окно заполните, как показано на Рис. 3.10.7. Нажмите “OK”. При создании второй плоскости нажмите на клавишу “Reverse Direction”, чтобы создать плоскость в противоположном направлении. Таким образом, мы получили две плоскости, расстояние между которыми равно ширине

колеса (11мм). Части массива, выступающие за пределы плоскостей, будут обрезаны.

Для этого мы воспользуемся командой “Split”  в подменю “Operations”. В качестве удаляемой части выберите торцевую поверхность зубьев колеса, в качестве элемента удаления — Plane.1, нажмите “ОК”. То же самое сделайте и с противоположной поверхностью колеса, только вместо Plane.1 выберите Plane.2 (Рис. 3.10.8).

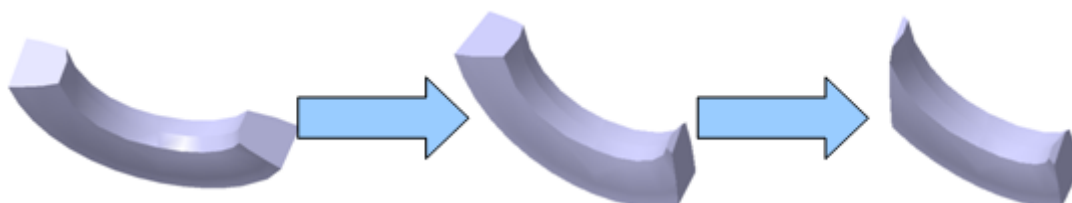




Рис. 3.10.8. Результат применения команды “Split”.

Выберите плоскость YZ, щелкните по иконке “Sketcher”  и перейдите в режим эскизирования. При помощи меню “Profile” создайте окружность так, как показано на рисунке, с центром на оси “V”, на расстоянии 48мм от оси “H”, диаметром 10,4мм. Для выхода из режима эскизирования нажмите на иконку “Exit Workbench” . Мы создали Sketch.6 (Рис. 3.10.9).

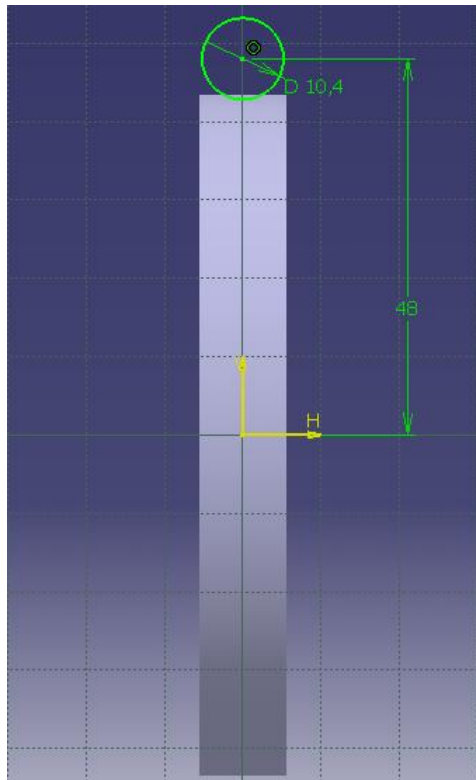



Рис. 3.10.9. Вид Sketch.6.

Щелкните по иконке “Groove Definition”  на панели инструментов “Sketch-Based Features”. В поле “First Angle” введите значение 360deg, в качестве профиля выберите Sketch.6, в качестве оси выберите ось “H”. Нажмите “OK” (Рис. 3.10.10).

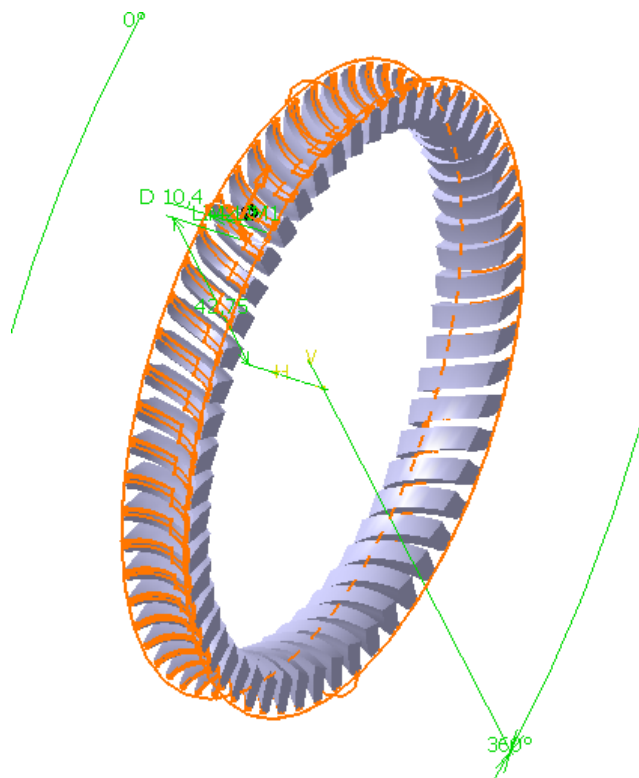


Рис. 3.10.10. Результат применения команды “Groove Definition”.

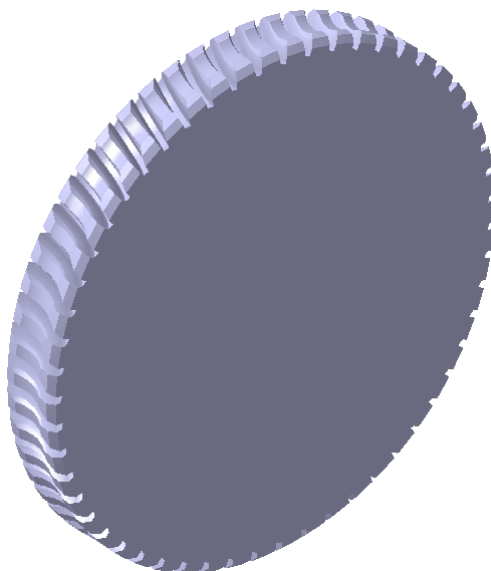





Рис. 3.10.11. Модель червячного колеса.


Для того, чтобы создать твердотельный объем колеса построим контур в плоскости XY, состоящий из построенной винтовой линии и трех прямых линий, две из которых параллельны оси “V” и симметричны относительно нее. Эти линии начинаются в окончании винто-

вой линии и заканчиваются на оси “Н”. Третья линия лежит на оси “Н” и замыкает контур. Для того чтобы построить контур воспользуйтесь командой “Line”  в подменю “Wireframe”.

Далее при помощи команды “Revolve”  проверните построенный контур вокруг оси Y. Постройте фаски по торцам колеса. Для этого щелкните на иконке “Chamfer”  и выберите два боковых ребра, построенного червячного колеса. Введите в диалоговом окне “Chamfer Definition” в поле “Angle” 45deg, а в поле “Length1” — 1мм. Нажмите “OK” (Рис. 3.10.11).

### 3.7. Методика создания сборки червячной пары в среде системы трехмерного моделирования CATIA V5

Создайте новый файл сборки. Для этого нажмите “File → New” и выберите тип файла “Product”.

Нажмите правой кнопкой мыши на вершине дерева модели и в подменю “Components” выберите вкладку “Existing Component”  и добавьте поочередно модель червячного вала и модель червячного колеса.

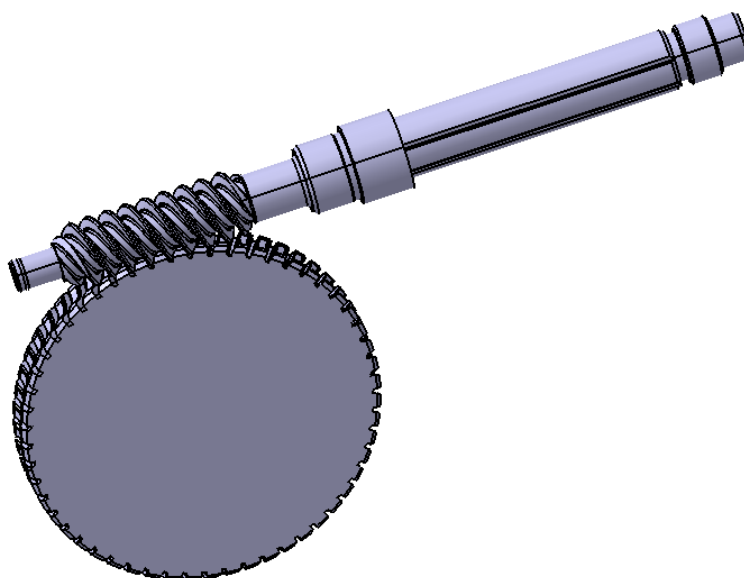






Рис. 3.11.1. Сборка червячной пары.

Нажмите на иконку “Offset”  в меню “Insert”. Установите по оси  $Z$  между осью червячного вала и осью  $X$  колеса расстояние равное  $-48\text{мм}$  (межосевое расстояние отрицательно, т. к. направлено против оси  $Z$ ).

Нажмите снова на иконку “Offset”  в меню “Insert”. Установите по оси  $Y$  между осью червячного вала и осью  $X$  колеса расстояние равное  $0$ .

Используя команду “Manipulate” , перемещайте червячный вал вдоль оси  $X$ , пока тот не займет нужного положения.

Для того, чтобы проверить пересекаются ли витки червяка с зубьями червячного колеса используйте команду “Clash” . Перемещайте червячный вал до тех пор, пока не достигнете контакта поверхностей (“Contact/Clash”).

Нажмите на иконке “Fix”  в меню “Insert”. И зафиксируйте положения червяка и червячного колеса (Рис. 3.11.1).

## Глава 4. Методика расчета червячной передачи

### 4.1. Алгоритмическая схема

Алгоритмическая схема включает в себя следующие этапы:

- создание геометрической модели;
- определение свойств материалов;
- установка атрибутов объемных тел;
- генерирование сетки конечных элементов с заданной степенью измельченности;
- определение контактных поверхностей;
- установка граничных и силовых условий;
- решение полученной системы уравнений;
- анализ полученных результатов.

В основу расчета были положены математические модели упругих свойств материала, моделирование контакта, алгоритм получения системы уравнений МКЭ и нахождения решения данной системы методом Ньютона – Рафсона.

## 4.2. Методика построения геометрической модели цилиндрической червячной передачи

### 4.2.1. Геометрический расчет эвольвентных червячных передач с использованием электронных таблиц

Рассмотрим некорригированную передачу с архимедовым червяком, имеющим в осевом сечении профильный угол  $\alpha_s = 20^\circ$ .

**Червяк.** Обозначения основных размеров червяка приведены на Рис. 4.2.1.

Связь между осевым шагом  $t_s$ , осевым модулем  $m_s$  и ходом винтовой линии червяка  $s$  выражается формулой

$$t_s = \pi m_s = \frac{s}{z_{\text{ч}}}.$$

Диаметр делительного цилиндра червяка, совпадающего в некорригированных передачах с начальным цилиндром, берут кратным осевому модулю червяка, т. е.  $d_{\text{дч}} = d_{\text{д1}} = q m_s$ , где  $q$  — число модулей в диаметре делительного цилиндра или относительная толщина червяка.

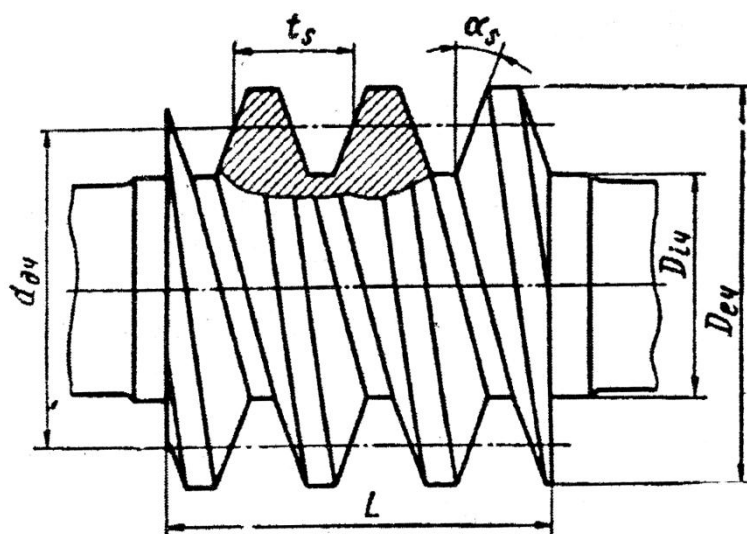


Рис. 4.2.1. Цилиндрический архимедов червяк.

Угол подъема винтовой линии  $\lambda$  на делительном цилиндре червяка связан с числом заходов  $z_{\text{ч}}$  и относительной толщиной червяка  $q$  соотношением

$$\operatorname{tg} \lambda = \frac{z_{\text{ч}}}{q}.$$

Таким образом, с увеличением  $q$  уменьшается угол подъема нарезки червяка, а, следовательно, и КПД передачи. При больших  $z_{\text{к}}$  (количество зубьев червячного колеса) расстояние между опорами чер-

вяка получается большим и для обеспечения достаточной жесткости червяка приходится увеличивать  $q$ .

Диаметр цилиндра выступов червяка (при коэффициенте высоты головки, равном единице) равен величине

$$D_{eч} = d_{дч} + 2m_s = m_s(q + 2).$$

Диаметр цилиндра впадин червяка (при радиальном зазоре по ГОСТу 2144 – 66, равном  $0,2m_s$ )

$$D_{iч} = d_{дч} - 2,4m_s = m_s(q + 2,4).$$

Длину нарезаемой части червяка (в некорригированных передачах) принимают по соотношениям:

- при  $z_ч = 1$  или  $2$ :  $L \geq (11 + 0,06z_к)m_s$ ;
- при  $z_ч = 3$  или  $4$ :  $L \geq (12,5 + 0,09z_к)m_s$ .

Для шлифуемых червяков величина  $L$ , полученная по указанным соотношениям, должна быть увеличена при  $m_s < 10$  мм на 25 мм, при  $m_s = 10 \div 16$  мм на 35 – 40 мм и при  $m_s > 16$  мм на 50 мм.

**Червячное колесо.** Сечение червяка и червячного колеса плоскостью, перпендикулярной к оси червяка, показано на Рис. 4.2.2.

Диаметр делительной окружности червячного колеса:

$$d_{дк} = d_{д2} = d_к m_s.$$

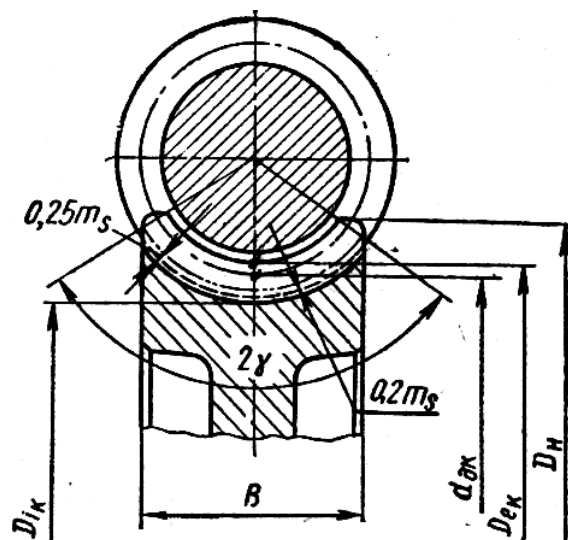


Рис. 4.2.2. Сечение червяка и колеса плоскостью, перпендикулярной к оси червяка.

Диаметр окружности выступов червячного колеса в среднем сечении (при коэффициенте высоты головки, равном единице):

$$D_{ек} = d_{дк} + 2m_s = m_s(z_к + 2).$$

Диаметр окружности впадин червячного колеса в среднем сечении (при радиальном зазоре, равном  $0,2m_s$ ):

$$D_{ик} = d_{дк} - 2,4m_s = m_s(z_к - 2,4).$$



Наружный диаметр червячного колеса  $D_H$  рекомендуется определять по формулам

- при  $z_q = 1$ :  $D_H \leq D_{ек} + 2m_s$ ;
- при  $z_q = 2$  или  $3$ :  $D_H \leq D_{ек} + 1,5m_s$ .
- при  $z_q = 4$ :  $D_H \leq D_{ек} + m_s$ .

Ширину венца колеса  $B$  рекомендуется принимать по соотношениям

- при  $z_q = 1 \div 3$ :  $B \leq 0,75D_{еч}$ ;
- при  $z_q = 4$ :  $B \leq 0,67D_{еч}$ .

Условный угол обхвата  $2\gamma$  червяка венцом колеса определяется точками пересечения дуги окружности диаметром  $D_{еч} - 0,5m_s$  с контуром венца (см. Рис. 4.2.2) вычисляется по формуле

$$\sin \gamma = \frac{B}{D_{еч} - 0,5m_s}.$$

Межосевое расстояние  $A$  определяется по формуле

$$A = \frac{1}{2}(d_{дк} + d_{дч}) = \frac{1}{2}m(z_k + q).$$

**Уравнение эвольвенты.** Зубья червячного колеса имеют эвольвентный профиль, поэтому нам необходимо знать геометрическое описание эвольвенты. Для вывода уравнения эвольвенты введем некоторые обозначения (см. Рис. 4.2.3):

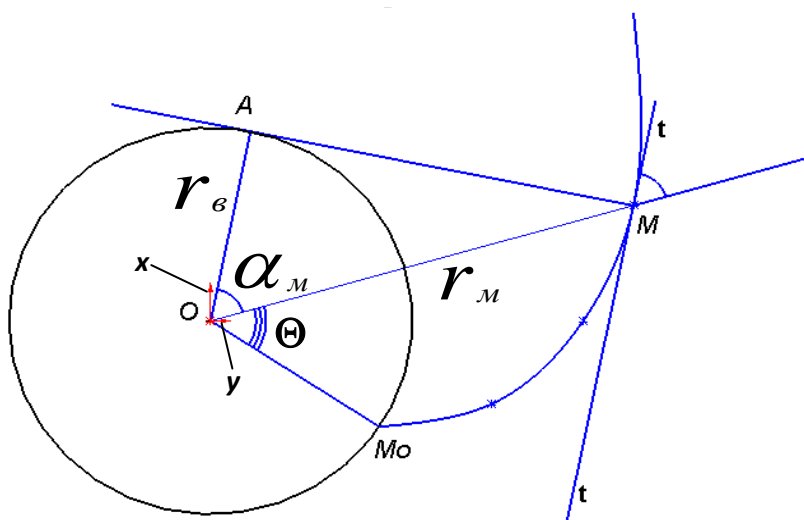


Рис. 4.2.3. Эвольвента окружности.

- $\mathbf{r}_в = \overrightarrow{OA}$  — радиус-вектор основной окружности или окружности эвольвенты, определяемый по формуле

$$r_в = r_M \cdot \cos \alpha_M; \quad (4.2.1)$$

- $\mathbf{r}_M = \overrightarrow{OM}$  — радиус-вектор точки  $M$ ;
- $\alpha_M$  — угол профиля в точке  $M$ ;

- $\Theta$  — эвольвентная функция или инволюта угла, определяемая по формуле

$$\Theta = \operatorname{tg} \alpha_M - \alpha_M = \operatorname{inv} \alpha_M; \quad (4.2.2)$$

Кроме того, на рисунке Рис. 4.2.3  $\overrightarrow{AM}$  — касательная к основной окружности в точке  $A$ ;  $t-t$  — касательная к эвольвенте в точке  $M$ . Уравнение эвольвенты строится в полярных координатах  $(\Theta, r_M)$ . Однако, нам нужны координаты точек эвольвенты в декартовой системе координат  $Oxy$  (см. Рис. 4.2.3).

Алгоритм расчета координат точек эвольвенты:

- угол профиля  $\alpha_M$  используется в качестве параметра, изменяется в модели от  $0^\circ$  до  $40^\circ$  с шагом  $2^\circ$ ;
- рассчитывается угол  $\Theta$  по формуле (4.2.2);
- по формуле (4.2.1) рассчитывается радиус основной окружности с использованием условия

$$r_M = \frac{d_{\text{дк}}}{2} \text{ при } \alpha_M = \alpha_s = 20^\circ;$$

- определяется значение  $r_M$  (по формуле (4.2.1)), с использованием полученного значения  $r_b$ , при различных значениях  $\alpha_M$ ;
- определяются координаты точек эвольвенты при помощи следующей зависимости:

$$\begin{cases} x &= r_M \cos \Theta, \\ y &= r_M \sin \Theta. \end{cases} \quad (4.2.3)$$

Весь расчет проводился с использованием электронных таблиц Microsoft Excel.

## 4.2.2. Построение геометрической модели передачи

**Создание модели червячного вала.** На основе данных, полученных при расчете геометрии, строится модель червяка по следующему алгоритму:

- создается *тело вращения* с конкретными размерами, которое имитирует заготовку правого червяка;
- создается профиль *впадины* и *правой винтовой линии*, на основе которых протягиванием получается вырез;
- левый червяк получается аналогично, при помощи замены правой винтовой линии на левую и изменения размеров тела вращения.

**Создание модели червячного колеса.** Создается профиль эвольвентного зуба колеса:

- по координатам, рассчитанным по формуле (4.2.3), строятся точки, по которым строится сплайн;

- создается направляющая кривая (небольшая часть правой винтовой линии);
- протягиванием профиля вдоль направляющей кривой получается модель зуба;
- копированием модели одного зуба вокруг оси колеса, создается массив из 30 зубьев;
- создается срединная часть колеса в виде тела вращения.

В результате реализации данного алгоритма получается геометрическая модель червячного колеса.

**Сборка полученных элементов.** Далее осуществляется сборка полученных элементов: червяк правый, червяк левый и червячное колесо (с правым направлением витка зуба). Результат показан на Рис. 4.2.4.

В дальнейшем полученная геометрическая модель импортируется в CAD/CAE систему ANSYS. Однако, для расчета деформации зубьев колеса модель частично упрощается. Рассматриваются только те зубья колеса и червяка, которые находятся в контакте. Это делается для того, чтобы сократить время расчета и уменьшить требования к вычислительным ресурсам.

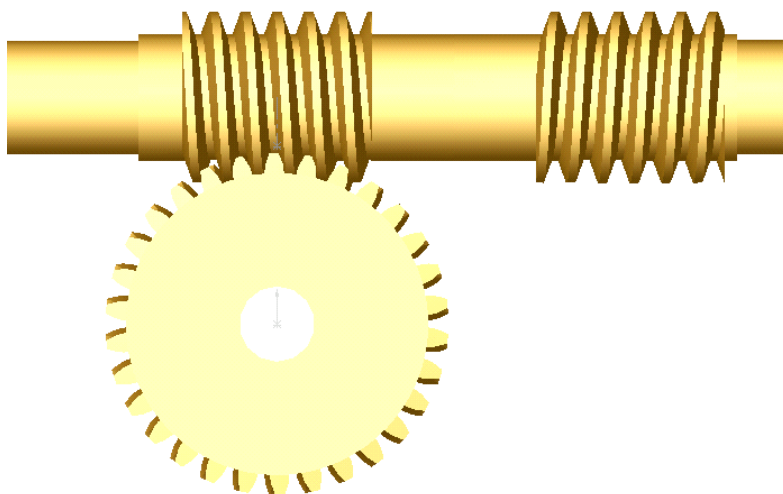


Рис. 4.2.4. Геометрическая модель червячной пары.

### 4.3. Определение свойств материалов

В данной задаче моделируются два типа материала: материал червяка и материал червячного колеса. В ходе всего моделирования оба материала подчиняются упругим соотношениям Гука. Значение модуля Юнга и коэффициента Пуассона для каждого материала можно получить из справочника [8].

## **4.4. Установка атрибутов объемных тел**

Для каждого твердого тела устанавливается свой порядковый номер моделируемого материала. Впоследствии этот атрибут переносится и на сетку конечных элементов, связанных с каждым объемом и каждой поверхностью, «натянутой» на данный объем.

## **4.5. Генерирование сетки конечных элементов**

Сначала для каждого твердого тела задаем свой размер конечного элемента, причем для зуба колеса размеры конечных элементов значительно меньше чем у червяка. Достигается это путем измельчения сетки в требуемой зоне. Затем задаются атрибуты конечных элементов: форма, порядковый номер материала, глобальная система координат. После чего генерируется сетка конечных элементов с заданными атрибутами.

## **4.6. Определение контактирующих поверхностей**

Чтобы сформировать контактную пару, определяем исходную поверхность и ответную поверхность.

Для каждой поверхности задаем свой тип контактного конечного элемента, указываем коэффициент трения и требуем, чтобы матрица жесткости была симметричной.

## **4.7. Установка граничных и силовых условий**

Здесь задаются граничные условия, имитирующие работу подшипников, что эквивалентно запрету перемещений червяка вдоль своей оси и в любом, перпендикулярном к ней, направлении и разрешению вращаться вокруг нее.

На червячное колесо в зоне его посадки на вал накладываются следующие ограничения — запрет перемещений в направлениях осей  $x$ ,  $y$ ,  $z$ .

Силовым условием является заданный крутящий момент, приложенный к червяку.

## 4.8. Установка параметров решения

Здесь устанавливаются необходимые параметры процесса вычисления решения. В качестве итерационного алгоритма решения выбирается метод Ньютона – Рафсона. Устанавливается автоматическое управление шагом по времени. Задача решается в два этапа. На первом этапе червяку задается поворот вокруг своей оси, для обеспечения начального контакта взаимодействующих поверхностей, а затем к нему прикладывается требуемый крутящий момент.

## 4.9. Анализ результатов

Проверка адекватности расчетной схемы проводится путем анализа результатов, полученных при использовании данной методики расчета и результатов, полученных при стандартном инженерном расчете. Здесь же приводятся диаграммы напряженно-деформированного состояния.

## 4.10. Выполнение расчета в ANSYS

### Исходные данные:

- $m = 8$  — модуль зацепления;
- $z_{\text{ч}} = 1$  — число заходов червяка;
- $z_{\text{к}} = 30$  — число зубьев колеса;
- $q = 15$  — число модулей в диаметре делительной окружности червяка;
- $M_{\text{ч}} = 215 \text{ Н} \cdot \text{м}$  — требуемая нагрузка на червяке;
- материал червяка — легированная сталь марки 40ХН;
- материал венца червячного колеса — бронза ОФ 10-1 отливка в кокиль.

### Свойства материалов:

- материал червяка считаем изотропным и подчиняющимся упругим соотношениям Гука:
  - модуль Юнга —  $E_{\text{ч}} = 2,1 \cdot 10^{11} \text{ Па}$ ,
  - коэффициент Пуассона —  $\nu_{\text{ч}} = 0,3$ ,
- материал червячного колеса также считаем изотропным и подчиняющимся упругим соотношениям Гука:
  - модуль Юнга —  $E_{\text{к}} = 0,9 \cdot 10^{11} \text{ Па}$ ,
  - коэффициент Пуассона —  $\nu_{\text{к}} = 0,35$ ,
- коэффициент трения —  $f = 0,04$ .

### Используемые конечные элементы.

Для аппроксимации конструкции выбран конечный элемент (КЭ) SOLID92 из библиотеки элементов ANSYS. SOLID92 — объемный четырехгранный КЭ, который определен десятью узловыми точками, имеющими по три степени свободы. Данный элемент хорошо подходит для моделирования нерегулярной сетки. Геометрия, узловые точки и система координат для этого элемента показаны на Рис. 4.10.1.

При моделировании «жестко-гибкого» контакта используются КЭ TARGE170 и CONTA174 для «целевой» (или «ответной») поверхности и поверхности контакта соответственно.

CONTA174 — КЭ, применяемый при трехмерном структурном и термическо-структурном контактном анализе. Этот элемент используется при моделировании контакта с учетом сил трения-скольжения. Элемент определен восемью узлами и расположен на поверхностях трехмерных КЭ, с которыми он связан. Он может вырождаться в шести узловой элемент в зависимости от формы основных объемных элементов. Геометрия и узловые позиции CONTA174 показаны на Рис. 4.10.2.

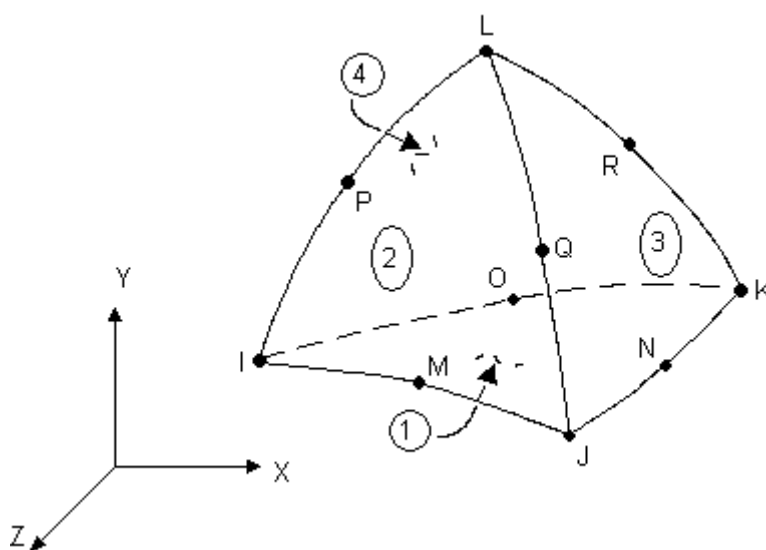


Рис. 4.10.1. Объемный конечный элемент SOLID92.

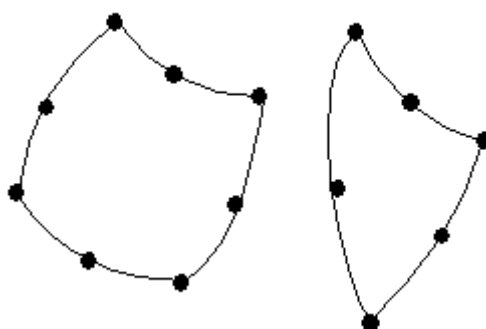


Рис. 4.10.2. Геометрическая интерпретация CONTA174.

TARGE170 используется для определения «целевой» поверхности. «Целевая» поверхность аппроксимируется набором целевых сегментных элементов (TARGE170) и связывается с поверхностью контакта при помощи вещественной константы. Выделяют восемь типов сегментных элементов, которые имеют различную форму (см. Рис. 4.10.3). Здесь используется треугольный тип.



Рис. 4.10.3. Типы целевых сегментных элементов.

### Сетка конечных элементов и граничные условия.

Сначала генерируется нерегулярная сетка КЭ. Затем на контактной поверхности зуба она измельчается, и вслед за этим определяется контактная пара. Накладываются граничные условия. Результаты разбиения трехмерной модели конструкции сеткой КЭ представлены на Рис. 4.10.4 и Рис. 4.10.5. Рис. 4.10.6 иллюстрирует расчетную схему с граничными условиями.

### Решение задачи.

Как говорилось ранее, задача решается в два этапа с использованием метода Ньютона – Рафсона. Напряженно-деформируемое состояние конструкции представлено на Рис. 4.10.7 – Рис. 4.10.19.

Рисунки с 4.10.9 по 4.10.12 отражают поле перемещений узловых точек колеса. Как видно из Рис. 4.10.12, максимальное суммарное перемещение узловых точек, расположенных на вершине зуба равно  $0,332 \cdot 10^{-3} \text{ м}$ .

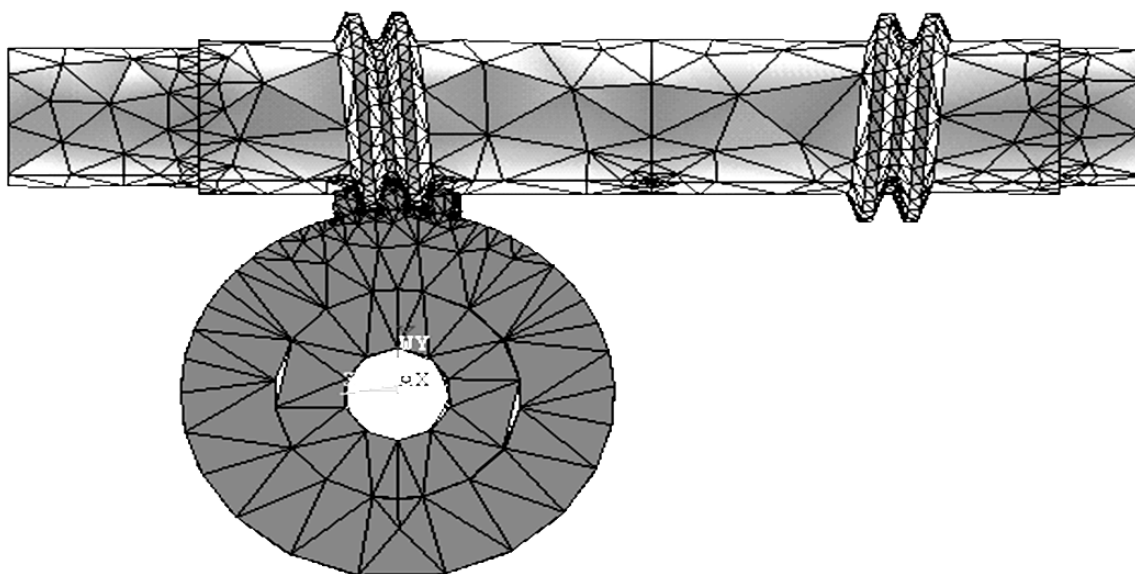


Рис. 4.10.4. Сетка трехгранных конечных элементов.

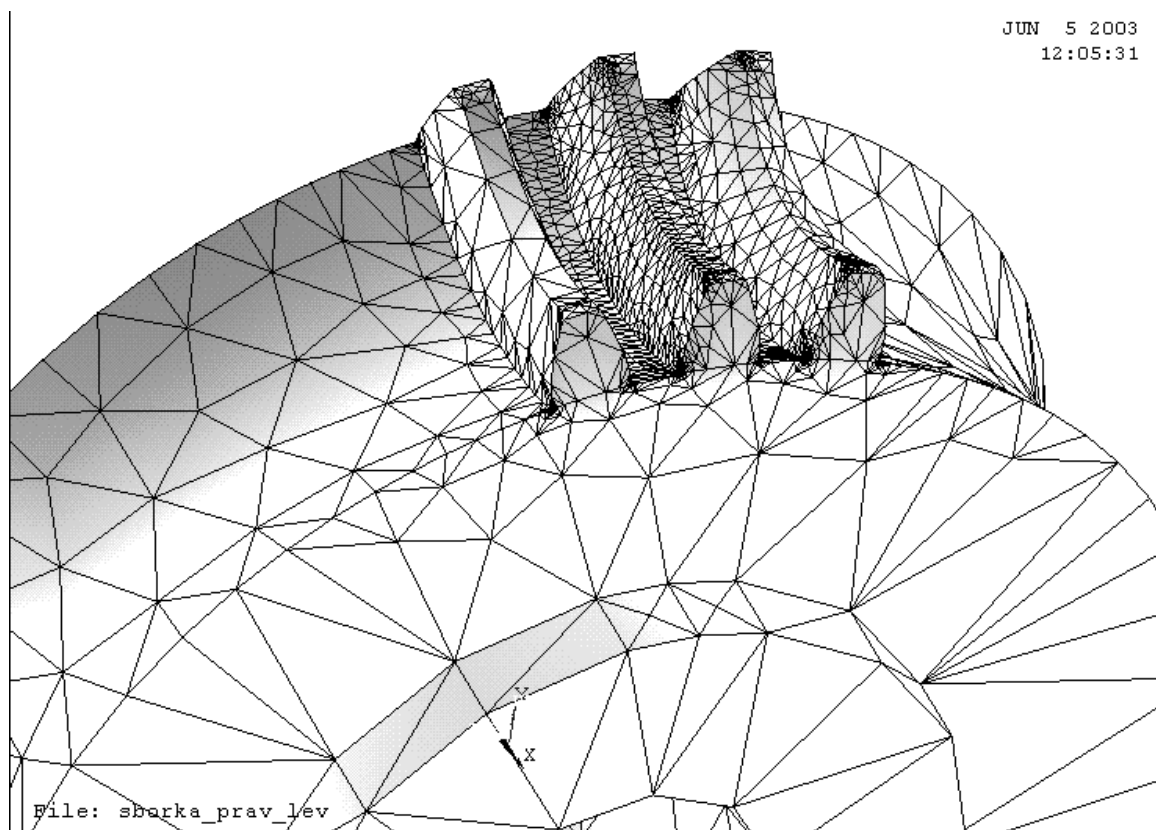


Рис. 4.10.5. Измельчение сетки КЭ в зоне контакта.



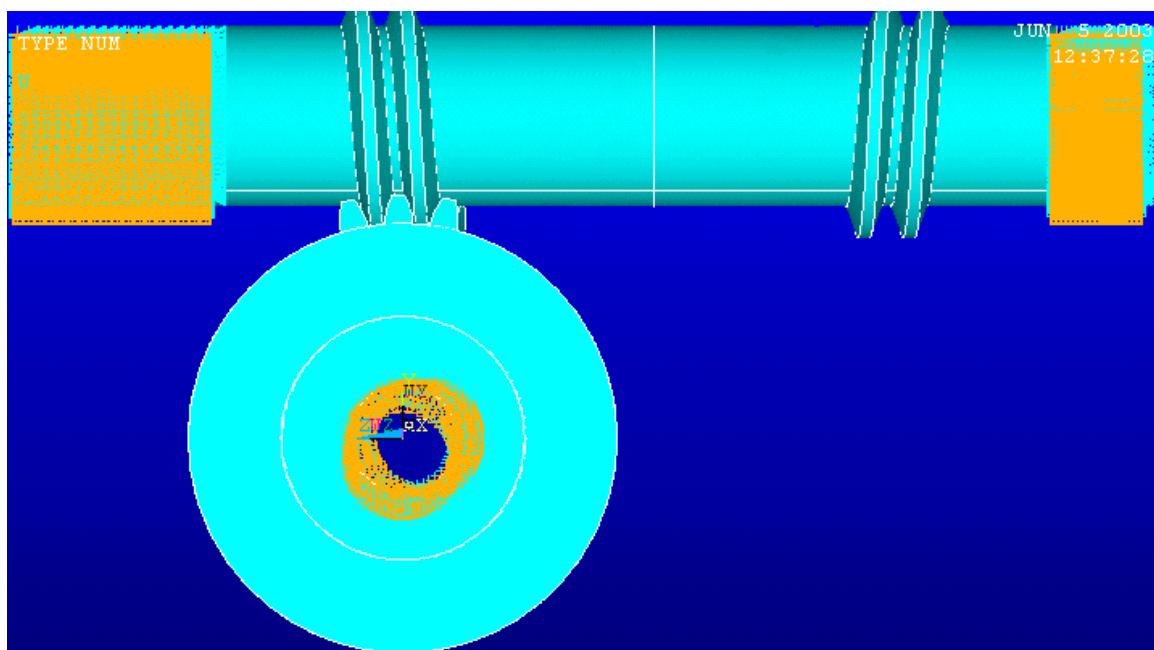


Рис. 4.10.6. Конструктивно-силовая схема.

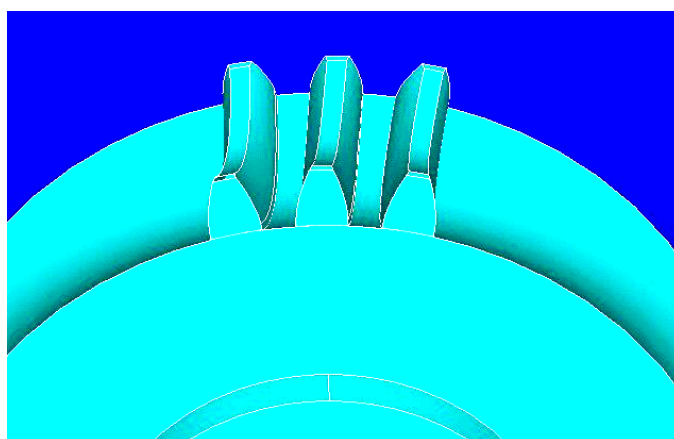


Рис. 4.10.7. Форма зуба до взаимодействия.

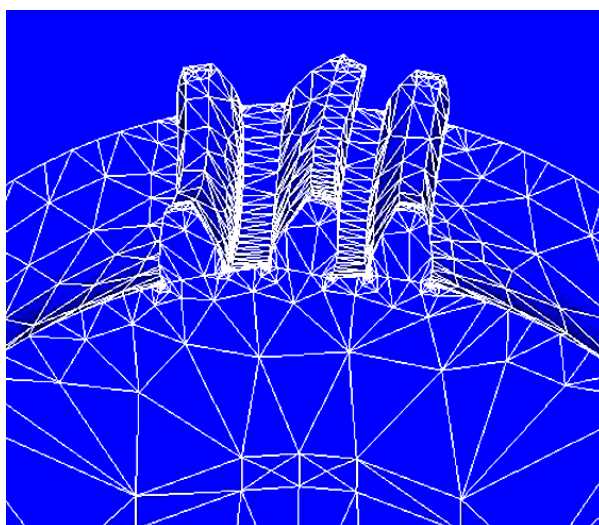


Рис. 4.10.8. Изменение формы зуба после взаимодействия.

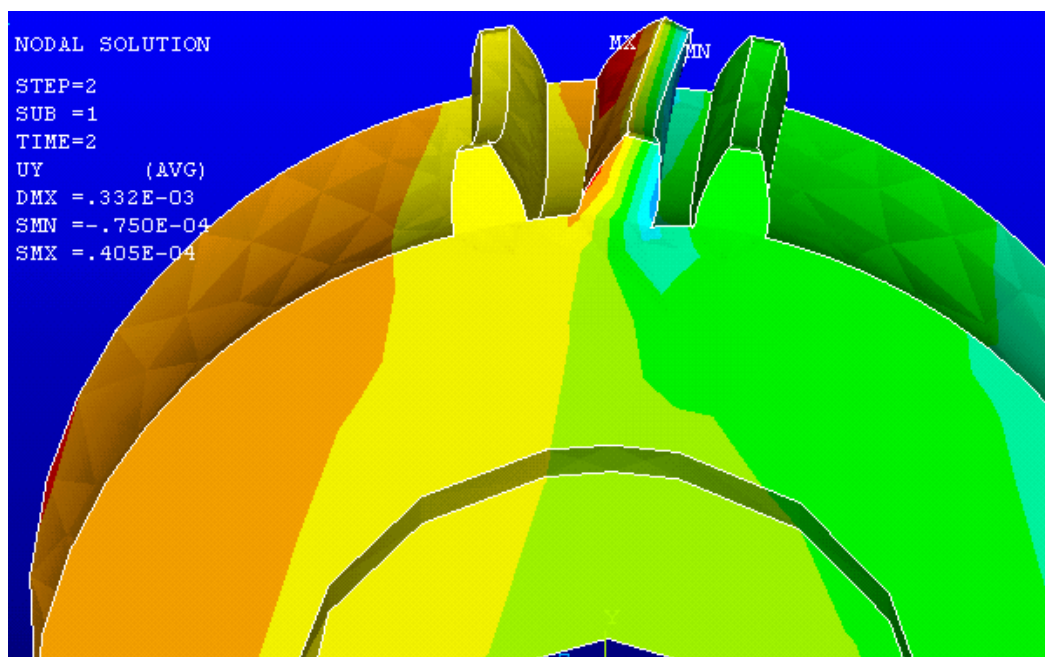


Рис. 4.10.9. Диаграмма перемещений узловых точек в направлении оси  $Y$ .

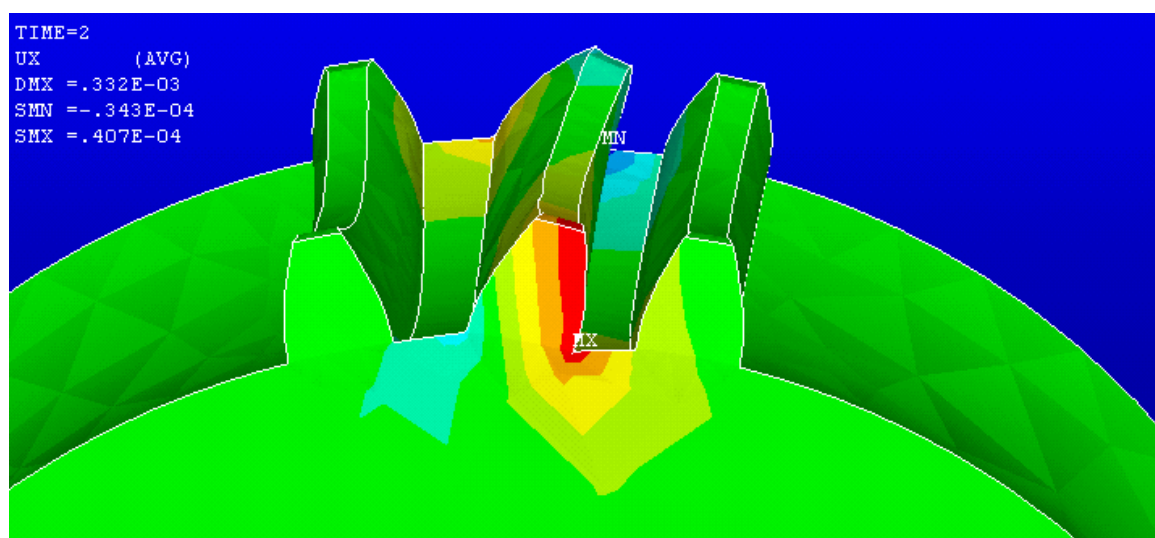


Рис. 4.10.10. Диаграмма перемещений узловых точек в направлении оси  $X$ .

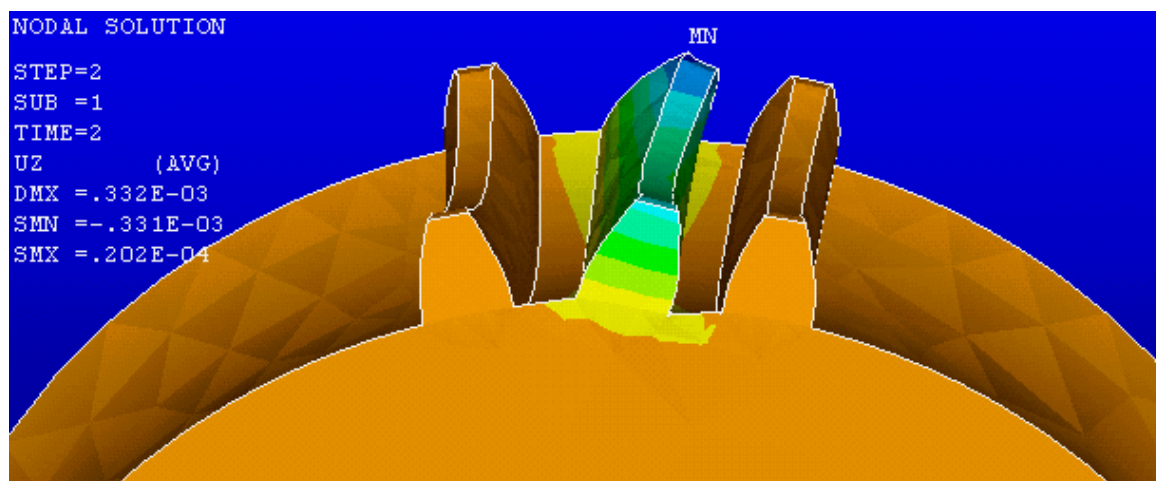


Рис. 4.10.11. Диаграмма перемещений узловых точек в направлении оси Z.

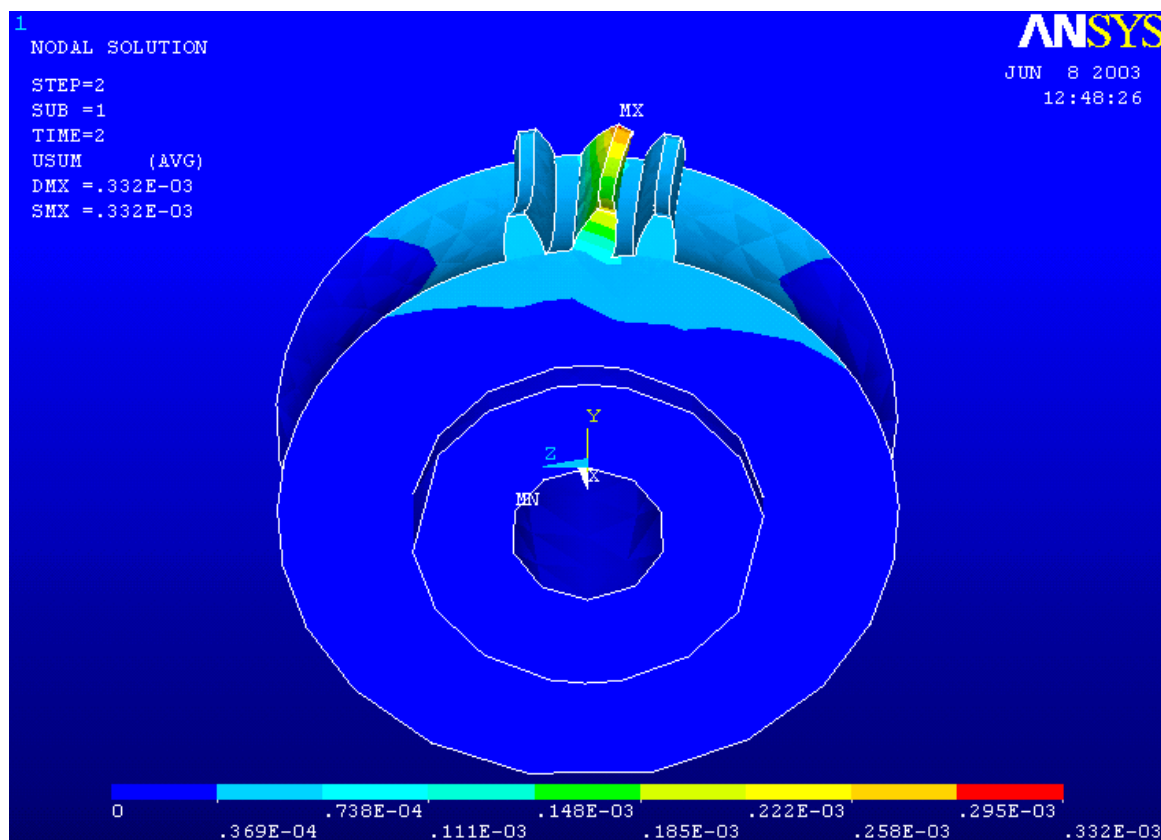


Рис. 4.10.12. Диаграмма суммарных перемещений узловых точек в процессе взаимодействия.

Рисунки с 4.10.13 по 4.10.18 показывают распределение поля деформаций, вызванных нормальными и касательными напряжениями. Деформация  $\epsilon_{xx}$  (Рис. 4.10.13) возникает благодаря наличию угла подъема винтовой линии,  $\epsilon_{yy}$  (Рис. 4.10.14) появляется из-за криволинейности профиля зуба (зуб имеет эвольвентный профиль), а  $\epsilon_{zz}$  (Рис. 4.10.18) — из-за наличия окружного усилия на колесе.

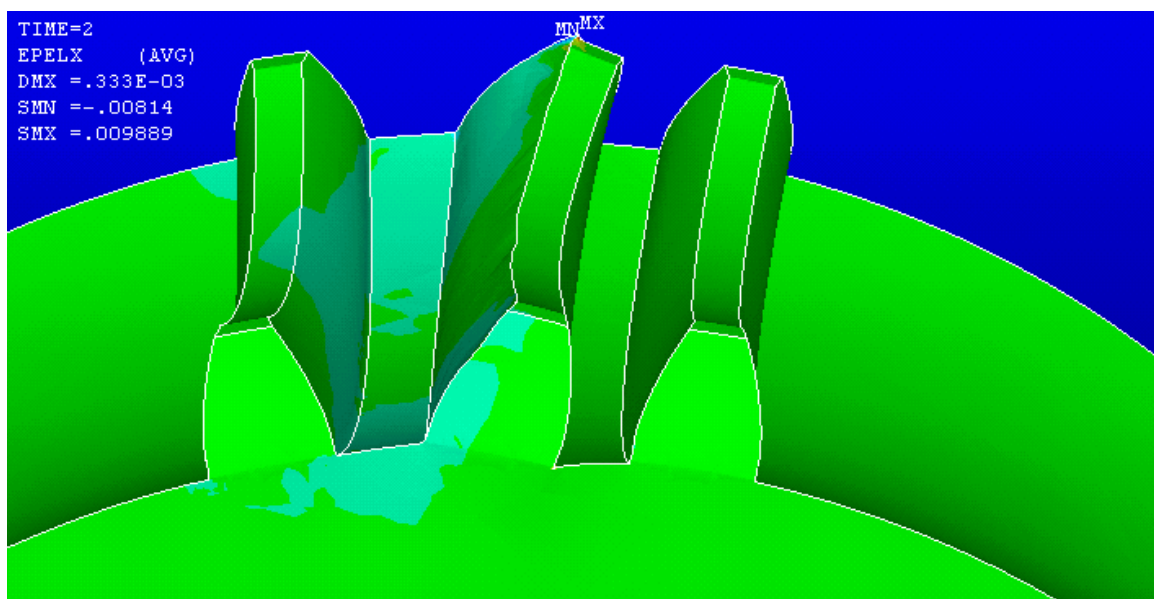


Рис. 4.10.13. Диаграмма распределения упругих деформаций  $\varepsilon_{xx}$ .

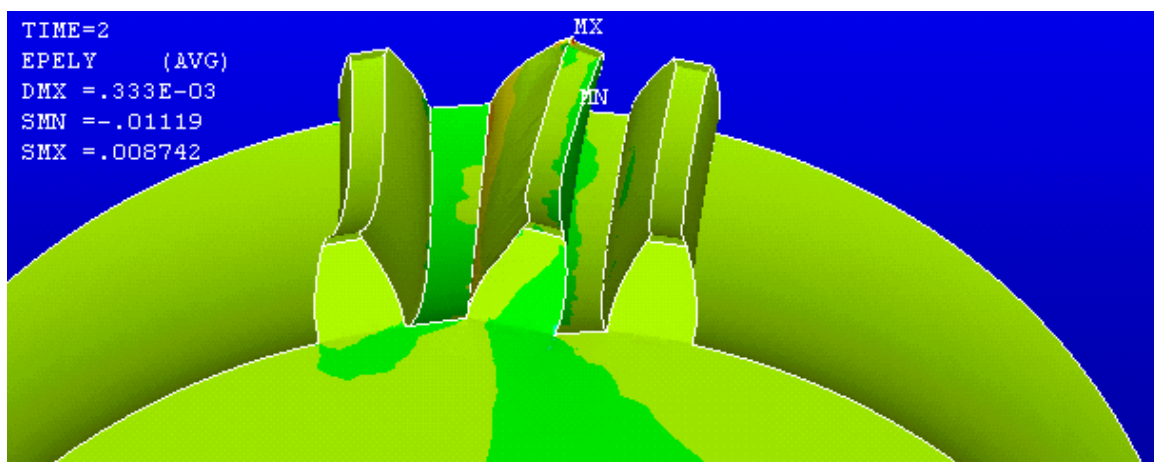


Рис. 4.10.14. Диаграмма распределения упругих деформаций  $\varepsilon_{yy}$ .

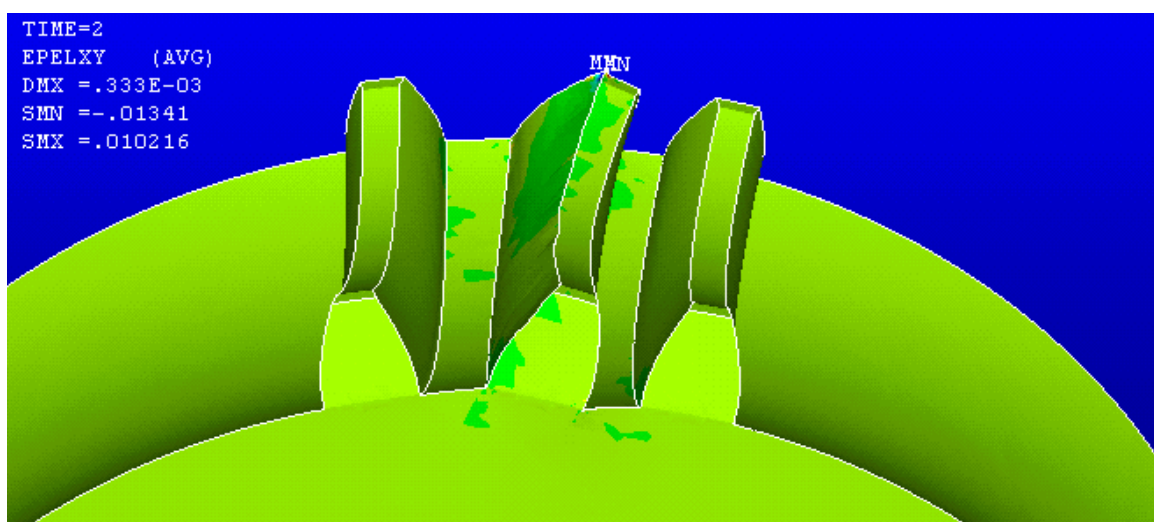


Рис. 4.10.15. Диаграмма распределения упругих деформаций  $\varepsilon_{xy}$ , вызванных касательными напряжениями.



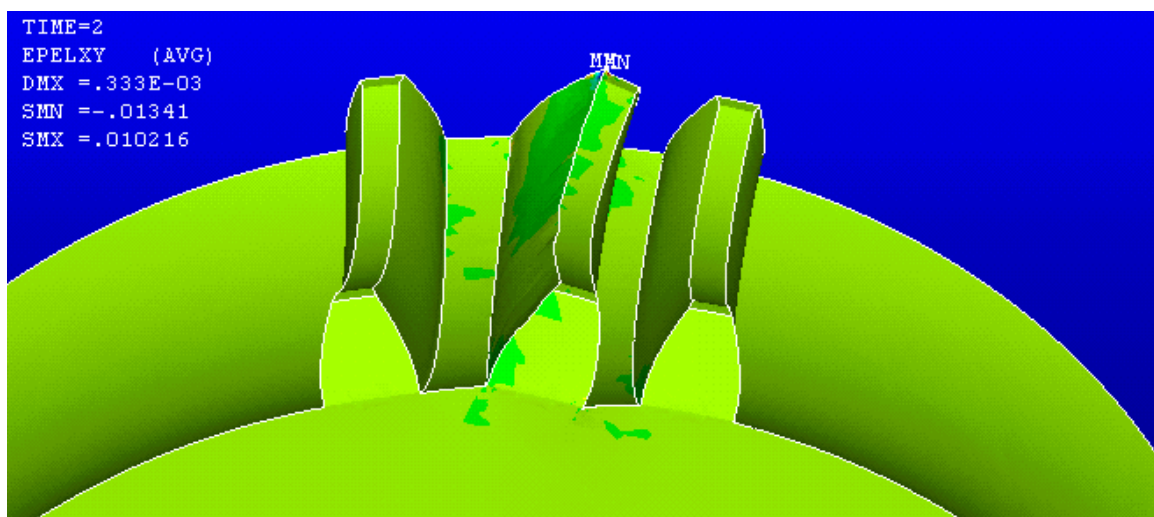


Рис. 4.10.16. Диаграмма распределения упругих деформаций  $\varepsilon_{zz}$ .

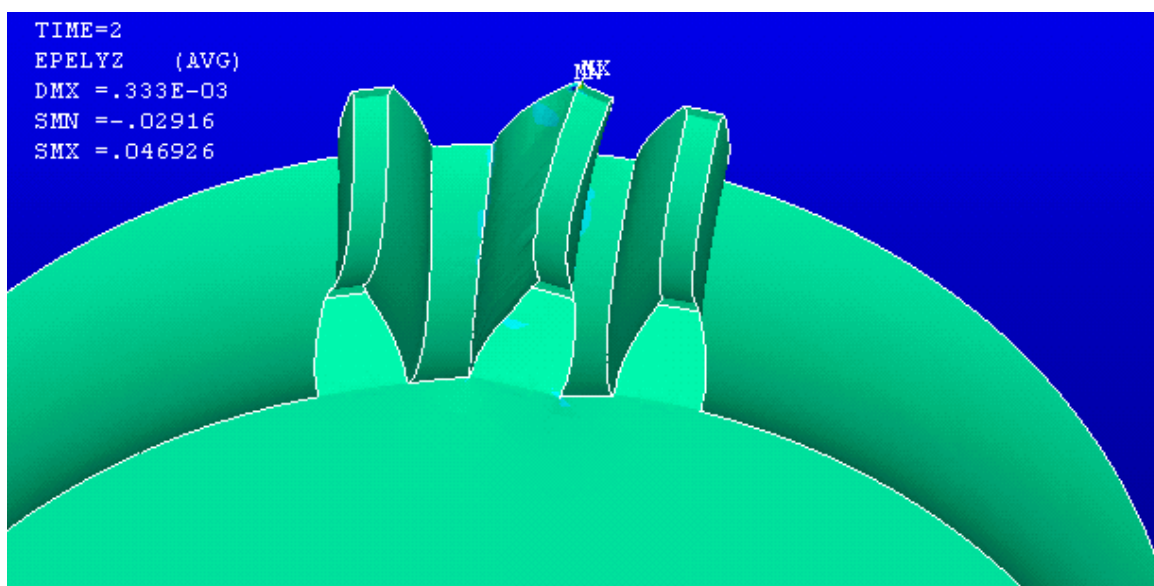


Рис. 4.10.17. Диаграмма распределения упругих деформаций  $\varepsilon_{yz}$ , вызванных касательными напряжениями.

Рис. 4.10.19 иллюстрирует распределение напряжений изгиба. Как видно из этого рисунка, напряжения возникают у ножки зуба и в небольшой области на вершине. Практика показывает, что зубья ломаются в опасном сечении, которое расположено у основания зуба, именно этот результат мы видим на данном рисунке. Наличие же напряжений на вершине зуба можно объяснить неточностью аппроксимации формы элементов конструкции. Численная величина напряжений изгиба (по Рис. 4.10.19) варьируется в следующих пределах:  $\sigma_{и(Ansys)} \approx 0,491 \cdot 10^8 \div 0,982 \cdot 10^8 \text{ Па}$ .

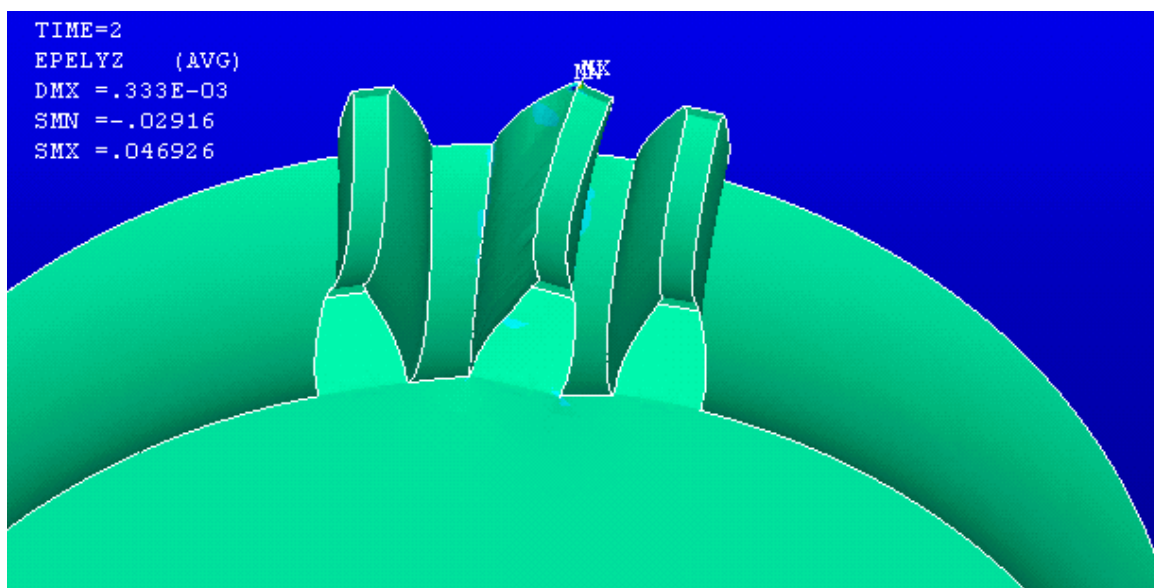


Рис. 4.10.18. Диаграмма распределения упругих деформаций  $\varepsilon_{zx}$ , вызванных касательными напряжениями.

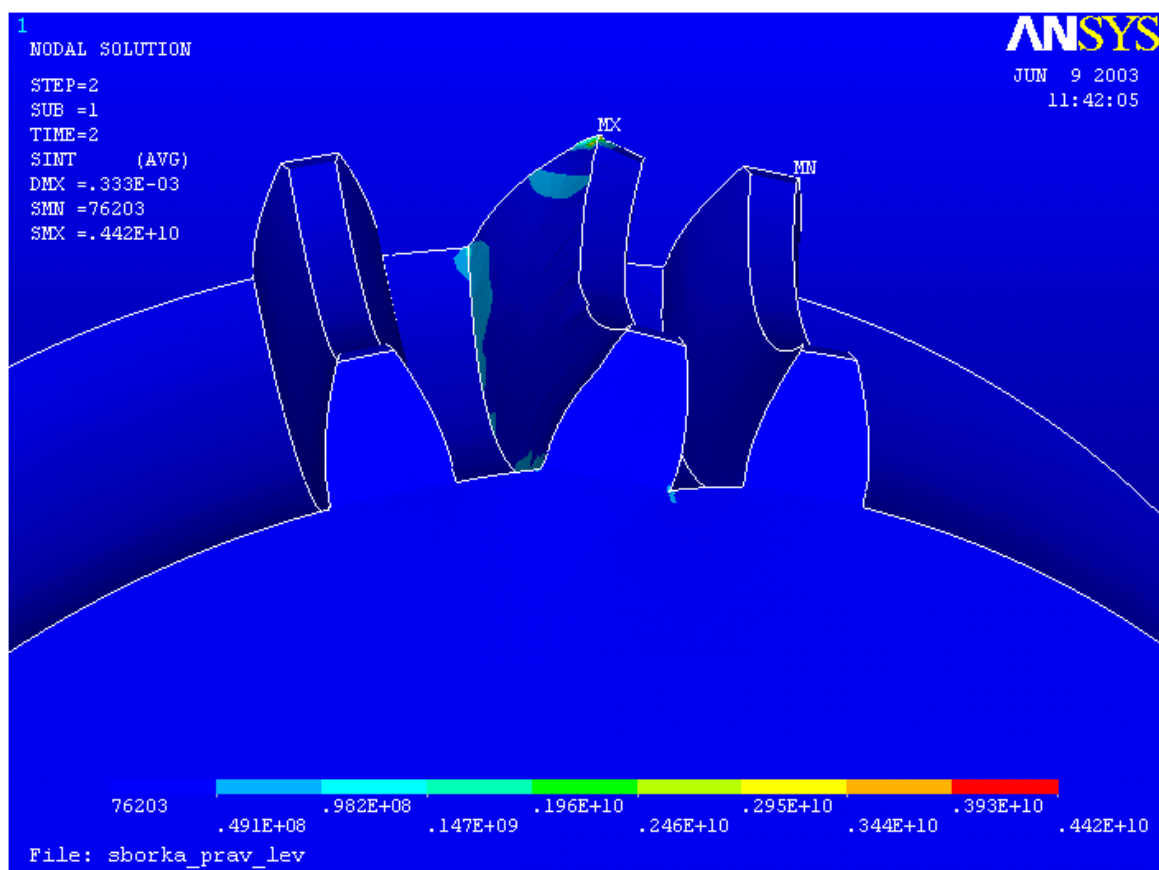


Рис. 4.10.19. Диаграмма распределения напряжений изгиба, возникающих в опасном сечении.

Для проверки адекватности разработанной расчетной схемы проведем инженерный расчет напряжений изгиба и сравним их с рас-

четными данными, полученными при использовании данной методики. Для этого запишем следующее соотношение

$$\sigma_{\text{и}} = \frac{0,64 \cdot M_{\text{к}}^p \cdot q \cdot 360}{z_{\text{ч}} \cdot i(q + 1,5) y_{\text{к}} \cdot 0,75 K_{\varepsilon} \cdot \pi d_{\text{ч}} \cdot 2\beta m^2 \gamma} \leq [\sigma_{\text{и}}],$$

в котором после подстановки числовых значений получим

$$\begin{aligned} \sigma_{\text{и}} &= \frac{0,64 \cdot 15 \cdot 360 \cdot 227 \cdot 10^4}{30 \cdot (15 + 1,5) \cdot 0,109 \cdot 0,75 \cdot 1,6 \cdot \pi \cdot 120 \cdot 64 \cdot 8^2} \left[ \frac{\text{Н} \cdot \text{мм}}{\text{мм}^3} \right] = \\ &= 78,47 \frac{\text{Н}}{\text{мм}^2} = 78,47 \cdot 10^6 \text{Па} \leq 80 \cdot 10^6 \text{Па}, \end{aligned}$$

где  $y_{\text{к}} = 0,109$  взято из [1], Табл. 44,  $2\beta = 64^\circ$ ,  $\gamma = 1$ .

Рассчитанная величина  $\sigma_{\text{и}}$  попадает в диапазон значений, полученных при расчете в системе ANSYS. Таким образом, можно говорить о правильности выбранной расчетной схемы. Из теории упругости известно, что напряжения и деформации связаны однозначной зависимостью — законом Гука. Следовательно, если напряжения рассчитаны корректно, то корректными будут и деформации. Заметим, что на вышеприведенных рисунках показанные деформации зуба, как обычно, масштабированы для обеспечения видимого визуального эффекта результата наложения соответствующих нагрузок.

# Глава 5. Modelica — язык объектно-ориентированного моделирования динамики технических систем

## 5.1. Основные понятия

В основе средств и систем унифицированного моделирования сложных гетерогенных физико-технических систем лежит реализация нескольких базовых принципов. На абстрактном проблемно независимом, уровне модель представляется стандартной системой дифференциально-алгебраических уравнений (ДАУ) вида

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{X}(t, \mathbf{x}, \mathbf{y}) \\ \mathbf{0} &= \mathbf{Y}(t, \mathbf{x}, \mathbf{y})\end{aligned}\tag{5.1.1}$$

где  $\mathbf{x} \in \mathbf{R}^m$ ,  $\mathbf{y} \in \mathbf{R}^n$ ,  $\mathbf{X}(t, \mathbf{x}, \mathbf{y}) \in \mathbf{R}^m$ ,  $\mathbf{Y}(t, \mathbf{x}, \mathbf{y}) \in \mathbf{R}^n$ . Подобное описание является универсальным инструментом, при помощи которого можно задать с той или иной степенью точности практически любые механические, электрические, магнитные, тепловые, химические и многие другие динамические процессы, встречающиеся в технике. На «прикладном» уровне, соответствующем интерфейсу языка моделирования с разработчиком модели, при представлении модели уравнениями (5.1.1) удобно визуально представлять элементы модели так, как они соотносятся между собой в физическом пространстве.

В этом состоит методика «физического» представления моделей процессов сложной природы. Физическое «мультиинженерное» моделирование обеспечивается в языке Modelica при помощи средств парадигмы объектно-ориентированного подхода.

В основе любого, подлежащего моделированию объекта, лежит, как, например, в языке C++, понятие класса. В отличие от алгоритмических языков объектно-ориентированного программирования поведение объектов класса характеризуется системой ДАУ. Более подробно объектно-ориентированные свойства языка Modelica будут рассмотрены ниже.

Спектр потенциальных приложений языка Modelica можно оценить, рассмотрев набор стандартных библиотек классов:

- `SIunits` задает около 450 типов переменных, имеющих физические размерности (угол, напряжение, масса, момент инерции, теплопроводность и др.);
- `Constants` содержит математические и физико-химические постоянные такие (число  $\pi$ , число  $e$ , число Авогадро и др.);



- `Media` содержит модели, описывающие свойства сплошных сред (идеальный газ, модели воды, модели воздуха, несжимаемые, сжимаемые, многофазные среды);
- `Math` — библиотека встроенных стандартных математических функций, в том числе функций над векторами и матрицами;
- `Blocks` — библиотека базовых блоков преобразования сигналов типа вход-выход, включающая в себя блоки непрерывных, дискретных, логических сигналов, в том числе передаточные функции, фильтры, источники сигналов и др.;
- `Electrical` содержит компоненты для построения моделей электрических цепей аналоговых и цифровых сигналов (резисторы, диоды, компоненты вентильных схем и пр.), электромеханических устройств (моторы, генераторы, трансформаторы и пр.) и отдельно компоненты для моделирования трехфазных индукционных машин;
- `Mechanics` содержит компоненты для моделирования трехмерного движения механических систем, состоящих из твердых тел (`multibody dynamics`) и отдельно компоненты для моделирования машин с одномерными поступательными и вращательными относительными движениями;
- `Thermal` — библиотека компонент для одномерных гидравлических систем (насосы, клапаны и пр.) и одномерных моделей процессов теплопередачи;
- `StateGraph` — библиотека компонент для моделирования конечных автоматов различной конструкции, реагирующих на поток дискретных событий;

Помимо перечисленных имеются всевозможные свободно распространяемые и коммерческие библиотеки классов, разработанные пользователями. Более подробно с этими библиотеками можно познакомиться на портале Ассоциации Modelica [20].

## 5.2. Базовые элементы языка

Дальнейшее краткое изложение языка Modelica следует, главным образом, его описанию в руководстве [23] и, в более развернутом варианте, в книге [19]. Для описания моделей любой сложности в языке Modelica используются следующие конструкции:

- базовые компоненты: скалярные типы `Real`, `Integer`, `Boolean`, `String`;
- структурные компоненты, применяемые для иерархического построения моделей;

- уравнения и алгоритмы;
- соединения;
- функции.

В качестве простейшего примера рассмотрим декларации скалярных объектов модели

```
Real x, y(start = 1);
parameter Real c = 1;
```

Здесь **Real** является именем предопределенного класса, задающего вещественнозначную переменную. Переменные класса **Real** имеют атрибуты (свойства). Например, атрибут **start** задает начальное значение переменной  $y$ . Обычно данный атрибут применяется для определения начальных данных в задаче Коши, автоматически формируемой компилятором перед началом процесса интегрирования системы дифференциальных уравнений модели.

Декларация переменной может иметь спецификатор **parameter** или **constant**, означающий, что переменная с такой спецификацией в процессе моделирования имеет постоянную величину (производная по времени от неё равна нулю). Спецификатор **parameter** позволяет менять значение переменной между прогонами модели без её перекомпиляции. Спецификатор **constant** позволяет изменить значение переменной только после повторной компиляции модели.

## 5.3. Классы

Поясним понятие класса на примере модели линейного осциллятора в виде груза единичной массы, скользящего без трения по горизонтальной направляющей под действием линейной пружины жесткости  $c$ . Левый «конец» пружины, соответствующий на самом деле её правому концу в недеформированном состоянии, задается переменной  $u$ , в общем случае считающейся функцией времени. На правом конце пружины находится материальная точка, положение которой задается координатой  $x$  (Рис. 5.3.1).

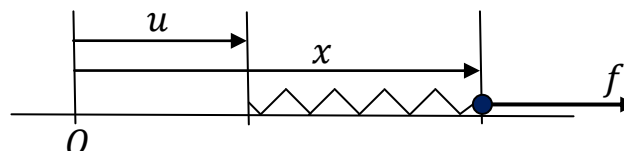


Рис. 5.3.1. Элементарный осциллятор

Движение системы задается линейным неоднородным дифференциальным уравнением второго порядка

$$\ddot{x} + cx = cu(t) + f(t). \quad (5.3.1)$$

Эту модель на языке Modelica можно представить в виде блока с входными сигналами  $u(t)$ ,  $f(t)$ , выходным сигналом  $x(t)$  и параметром  $c$ :

```
class ElementaryOscillator
  parameter Real c = 1 "Spring stiffness";
  Real x(start = 1) "Coordinate";
  Real v(start = 0) "Velocity";
  Real a "Acceleration";
  Real u "Input signal";
  Real f "External force";
equation
  der(x) = v;
  der(v) = a;
  a + c*x = c*u + f;
end ElementaryOscillator;
```

Легко видеть, что модель является неполной. Для определения движения осциллятора необходимо указать еще уравнения, задающие изменение входных переменных  $u$ ,  $f$  в зависимости от времени.

Описанную в виде класса модель осциллятора можно использовать для построения цепочки последовательно соединенных осцилляторов с пружинами, в общем случае, различной жесткости. Возьмем цепочку из двух осцилляторов и определим модель более высокого структурного уровня

```
class ChainOfOscillators
  ElementaryOscillator O1(c = 10);
  ElementaryOscillator O2(c = 100, x(start = 2));
equation
  O1.u = sin(time);
  O1.f = O2.c*(O2.x - O2.u);
  O2.u = O1.x;
  O2.f = 0;
end ChainOfOscillators;
```

Здесь **time** есть предопределённая переменная времени модели. Из уравнений модели видно, что левый конец левой пружины совершает периодические (синусоидальные) колебания возбуждая, таким образом, всю цепочку. Из этого примера видно, что параметры классов можно модифицировать при создании новых объектов, а обращения к переменным объекта класса идентифицируются обычными для структуры квалификаторами с использованием точки в качестве разделителя. Эту модель можно условно представить в виде схемы Рис. 5.3.2.

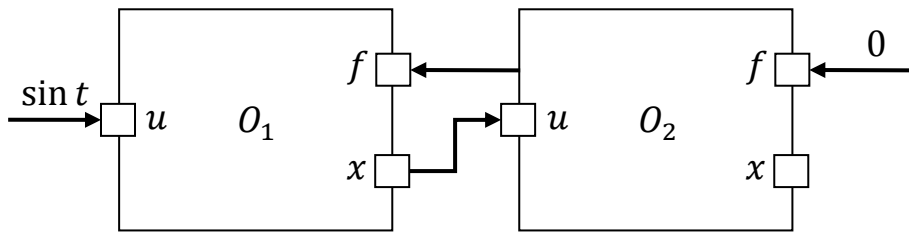


Рис. 5.3.2. Схема класса ChainOfOscillators

Теперь можно задать модель двух пар пружинных цепочек, «работающих» параллельно и независимо, перейдя к следующему структурному уровню описания классов в виде

```
class ChainsOfOscillators
  ChainOfOscillators O12(O1(c = 10), O2(c = 100));
  ChainOfOscillators O34(O1.c = 10, O2.c = 100);
end ChainsOfOscillators;
```

Рассмотренный пример иллюстрирует средства Modelica, обеспечивающие построение структурированных многоуровневых моделей. Здесь в декларациях двух идентичных объектов продемонстрированы два различных способа задания значений параметров этих объектов.

Классы в языке Modelica имеют несколько разновидностей, определяемых ключевыми словами: **model**, **connector**, **record**, **block**, **function**, **type**, **package** и накладывают различные ограничения на поведение объектов соответствующих классов. Например, класс **record**, однако замена любого из этих ключевых слов на **class** сохраняет смысл модели.

Поясним особенности некоторых из этих типов классов. Класс **record** запрещает раздел **equation**, его удобно использовать для размещения структурированной информации. Класс **record** удобно использовать как класс-запись для хранения набора значений параметров модели. В рассматриваемой модели ElementaryOscillator имеется всего один параметр  $c$  и введение дополнительного класса **record**, вообще говоря, не оправдано. Здесь же мы для иллюстрации методики применения класса **record** проведем следующую модификацию предыдущего примера:

```
record OscillatorData
  Real c;
end OscillatorData;
record ChainOfOscillatorsData
  OscillatorData O1;
  OscillatorData O2;
end ChainOfOscillatorsData;
```

```

model AnotherChainOfOscillators
  ChainOfOscillatorsData Data(O1(c = 10), O2(c = 100));
  ChainOfOscillators O12 = Data, O34 = Data;
end AnotherChainOfOscillators;

```

где присвоение (инициализация) структуры параметров приводит к тому же результату, что и в первоначальном классе `ChainOfOscillators`. Инициализация работает корректно, поскольку структура правой части этого присвоения (от класса `ChainOfOscillatorsData`) является подструктурой класса `ChainOfOscillators`. Это один из примеров реализации полиморфизма объектно-ориентированной парадигмы.

Класс **package** является типичным контейнером и предназначен для гнездования описаний классов и структурирования пространства имен. Для доступа снаружи к классу, размещенному в классе **package**, необходимо использовать квалификатор с точкой в качестве разделителя. Он играет роль, аналогичную роли **unit** в языке Pascal и может содержать константы и другие классы, в том числе и типа **package**. Классы этой категории называются также библиотеками (или пакетами) классов. Методика поиска объектов в пакетах аналогична методике поиска объектов в иерархии каталогов (папок) файловых систем.

В последних известных реализациях языка Modelica имеется возможность перегрузки операций, что соответствует одному из важнейших проявлений полиморфизма. В общем случае под полиморфизмом обычно понимают «правильную» интерпретацию операций языка в зависимости от типа операндов, задаваемых при помощи классов-наследников данного базового класса. В частности, правильная интерпретация структуры данных объектов, участвующих в операции присваивания объектов (при формировании параметров модели) также является проявлением полиморфизма языка. В таких случаях чаще всего считается, что левым операндом должен быть объект базового класса, правым — объект его класса-наследника.

Помимо полиморфизма в языке Modelica достаточно просто реализована инкапсуляция (сокрытие) реализации в классах. Для этого используется ключевое слово **protected** — аналог ключевого слова **private** в языке программирования C++. Рассмотрим, представленный выше пример цепи, состоящей из двух осцилляторов с использованием инкапсуляции

```

model ChainOfOscillators2
  parameter Real c1 = 10;
  parameter Real c2 = 100;
  input Real u;

```

```

output Real x;
protected
  ElementaryOscillator O1(c = c1);
  ElementaryOscillator O2(c = c2);
equation
  O1.u = u;
  O1.f = O2.c*(O2.x - O2.u);
  O2.u = O1.x;
  O2.f = 0;
  x = O2.x;
end ChainOfOscillators2;

```

Схема этой модели показана на Рис. 5.3.3.

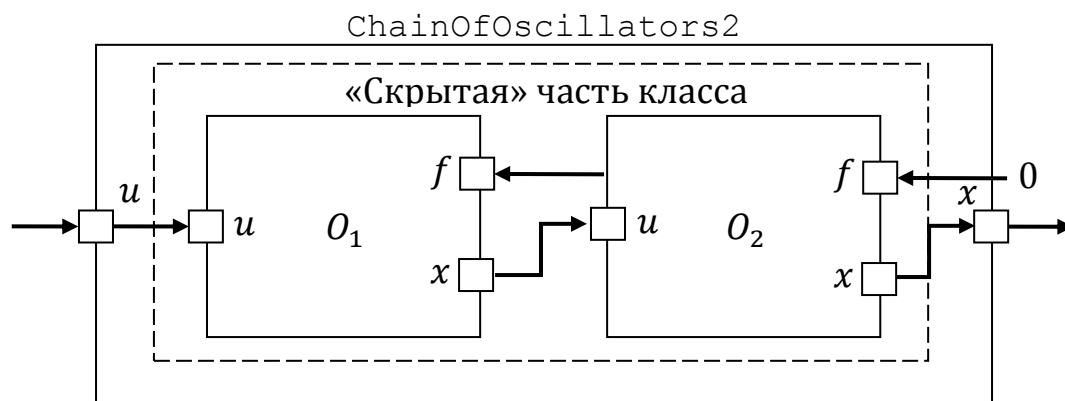


Рис. 5.3.3. Схема класса ChainOfOscillators2

## 5.4. Соединения

Для иллюстрации «работы» класса **connector** удобно рассмотреть пример из электротехники [23]. Любую точку электрической цепи можно задать при помощи виртуального контакта, задаваемого классом

```

connector Pin
  Real v "Voltage level at a Pin";
  Real i "Current";
end Pin;

```

где переменная  $v$  соответствует напряжению, а  $i$  — току, текущему через контакт.

Всюду далее предполагаем, что виртуальный электрический ток «течет» от положительного заряда к отрицательному. Элементарный резистор (сопротивление) определяется заданием двух контактов (в секции деклараций класса) и закона Ома (в секции уравнений класса), связывающего ток и падение напряжения на данном резисторе в виде класса

```

model Resistor
  parameter Real R = 1 "Resistance by default";
  Pin p "Positive pin";
  Pin n "Negative pin";
equation
  n.i = p.i;
  R*p.i = p.v - n.v; // Ohm's law
end Resistor;

```

Рассмотрим теперь (Рис. 5.4.1) простейшую (незамкнутую) электрическую цепь, состоящую из двух последовательно соединенных резисторов  $R_1$  и  $R_2$ . Модель такой цепи можно представить в виде

```

model LineOfResistors
  Resistor R1(R = 100);
  Resistor R2(R = 200);
equation
  R1.n = R2.p;
end LineOfResistors;

```

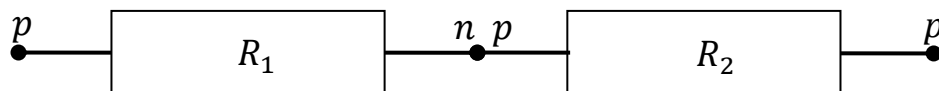


Рис. 5.4.1. Линейная цепь из двух резисторов

Можно усовершенствовать построенные компоненты электрических цепей в виде

```

type Voltage = Real(unit = "V");
type Current = Real(unit = "A");
type Resistance = Real(unit = "Ohm");
connector Pin
  Voltage v;
  flow Current i;
end Pin;
model Resistor
  parameter Resistance R = 1 "Resistance by default";
  Pin p "Positive pin";
  Pin n "Negative pin";
equation
  n.i + p.i = 0;
  R*p.i = p.v - n.v; // Ohm's law
end Resistor;

```

В случае задания переменной  $i$  со спецификатором **flow** считается, что ток через обе клеммы «втекает» в резистор. Так что, если, например,  $p.i$  положителен («втекает»), то  $n.i$  должен быть отрицательным («вытекать»). Спецификация **flow** задает так называемую потоковую переменную, составляющую бонд вместе с переменной

усилия (в данном случае это  $v$ ), теории бондграфов [24]. Это означает, что если к данному контакту подсоединены контакты нескольких электротехнических компонент цепи, то сумма всех токов в соединении (в соответствии с законами Кирхгофа) должна быть равна нулю. В теории бондграфов такое соединение бондов называется 0-узлом.

Составим теперь модель из трех резисторов, соединенных в звезду (Рис. 5.4.2)

```

model StarOfResistors
  Resistor R1(R = 100);
  Resistor R2(R = 200);
  Resistor R3(R = 300);
equation
  connect(R1.p, R2.p);
  connect(R1.p, R3.p);
end StarOfResistors;

```

Здесь **connect** является встроенным оператором языка, генерирующим уравнения, связывающими переменные в соответствии с их типами. В данном случае компилятор сгенерирует следующую систему (линейных) уравнений

$$\begin{aligned}
 R1.p.v &= R2.p.v; \\
 R1.p.v &= R3.p.v; \\
 R1.p.i + R2.p.i + R3.p.i &= 0;
 \end{aligned}$$

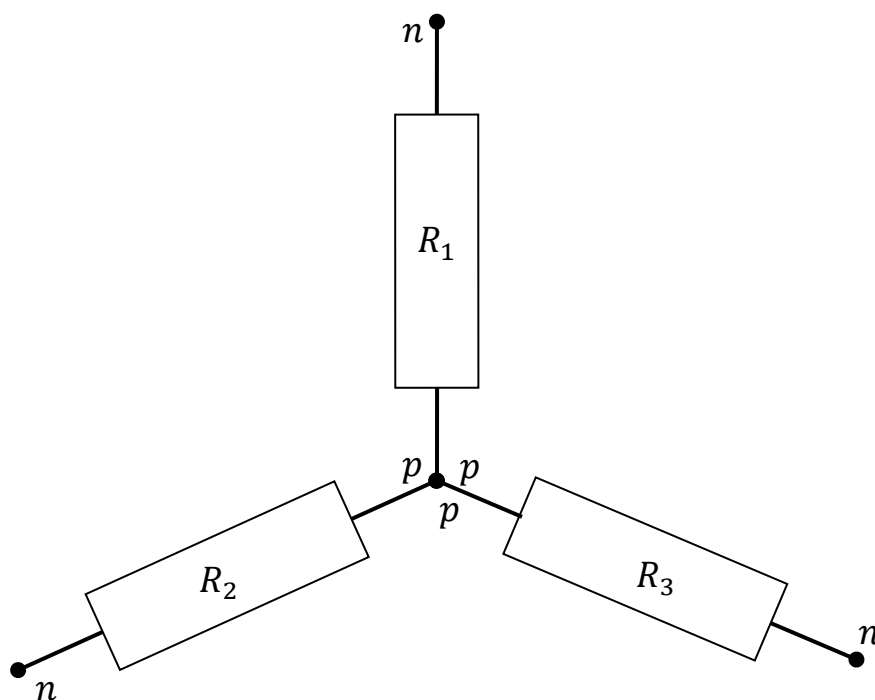


Рис. 5.4.2. Цепь из трех резисторов в виде звезды



## 5.5. Частичные модели и наследование

Важной особенностью, обеспечивающей реализацию наследования, как структурных, так и поведенческих свойств классов, является возможность задания частичной (неполной) модели. В этом случае неполный класс может многократно использоваться в качестве базового или суперкласса для классов-наследников или производных классов, разрабатываемых для различных более конкретных приложений. Для иллюстрации этого снова вернемся к электротехническим примерам. Простейшие компоненты электрических цепей (резисторы, емкости, индуктивности, источники питания) имеют общие конструктивные свойства: наличие двухконтактного (двухпинового) интерфейса между компонентом и остальной частью цепи. При этом один из контактов естественным образом считается положительным, а второй — отрицательным. Такой двухконтактный интерфейс называется портом. Для задания однопортовых компонентов электрических цепей удобно для всех этих компонент определить общего предка — базовый класс вида

```
partial model OnePort
  Pin p "Positive pin";
  Pin n "Negative pin";
  Voltage v "Voltage drop";
  Current i;
equation
  v = p.v - n.v;
  n.i + p.i = 0;
  i = p.i;
end OnePort;
```

структура которого и поведение, задающееся дифференциальными и алгебраическими уравнениями, будут автоматически повторяться в производных классах. Объекты описанного типа известны также в теории колебаний, где их принято называть двухполюсниками [3]. Сам по себе представленный класс не может породить «жизнеспособный» объект — для этого в нем не хватает уравнений, задающих поведение какого-либо из известных компонент электрических цепей. Эта ситуация аналогична абстрактным классам языка C++, которые играют роль общих базовых классов для целых категорий их наследников. Сами абстрактные классы не могут использоваться в качестве типов для создания объектов программы.

Рассмотрим модель индуктора, задаваемого при помощи дифференциального уравнения

$$L \frac{di}{dt} = v,$$

где параметр  $L$  является индуктивностью компонента, измеряемой стандартным образом в системе СИ в генри. Код компоненты индуктивности на Modelica можно представить в виде

```
model Inductor
  extends OnePort;
  parameter Inductance L = 1;
equation
  L*der(i) = v;
end Inductor;
```

где **der** является встроенным оператором языка Modelica, соответствующим дифференцированию по независимой переменной, предполагаемой по умолчанию временем.

Здесь предполагается, что где-то, в области видимости компилятора (например, сразу перед описанием класса `Inductor`), находится описание типа

```
type Inductance = Real(unit = "H");
```

задающего тип переменной размерности «индуктивность». Забегая вперед, для того, чтобы иметь возможность создавать работающие модели, скажем, что все физические размерности (как типы переменных) будут доступны компилятору, если в самом начале списка своих моделей, содержащегося обычно в контейнерном классе пользователя категории **package**, разместить директиву импорта имен вида

```
import SI = Modelica.SIunits;
```

Тогда все переменные, имеющие физические размерности можно декларировать используя имя `SI` в качестве префикса. Например, вместо наименования типа `Inductance` нужно писать `SI.Inductance`. Всюду далее будем предполагать, что такая директива имеется. Например, переопределим модель резистора в новом стиле

```
model Resistor
  extends OnePort;
  parameter SI.Resistance R = 1;
equation
  R*i = v;
end Resistor;
```

Аналогично определим модель конденсатора

```
model Capacitor
  extends OnePort;
```

```

parameter SI.Capacitance C = 1;
equation
  C*der(v) = i;
end Capacitor;

```

описываемую при помощи дифференциального уравнения

$$C \frac{dv}{dt} = i$$

с параметром емкости  $C$ .

**Замечание.** В Modelica, как и в C++, поддерживается множественное наследование. Альтернативой множественному наследованию, приводящей к более корректной конструкции результирующих моделей, является параметризация классов, рассмотренная ниже.

Здесь уместно обратить внимание на то, что порождение новых классов при помощи наследования сужает множество объектов, которые могут создаваться при помощи производного класса, в сравнении с множеством объектов, соответствующим базовому классу (если этот класс не частичный). Сужение множества объектов достигается добавлением дополнительных спецификаций, играющих роль дополнительных ограничений, структуры и поведения. Этим объясняется использование в Modelica при наследовании ключевого слова **extends**.

В качестве еще одного примера класса-наследника частичной модели OnePort рассмотрим модель резистора, зависящего от температуры его проводника,

```

model HeatResistor
  extends OnePort;
  parameter SI.Resistance R = 1 "Ref. resistance";
  parameter Real RT(unit = "Ohm/degC") = 0 "Rate";
  parameter SI.Temperature Tref = 20 "Ref. temp.";
  SI.Temperature T "Actual temperature";
equation
  v = i*(R + RT*(T - Tref));
end HeatResistor;

```

В этой модели рост текущего сопротивления зависит от температуры по линейному закону. Эту модель нельзя «расширить» из ранее рассмотренного резистора, т. к. её уравнение должно не добавляться к уже существующему уравнению закона Ома, а заменить его на новое, учитывающее фактор нагревания проводника.

## 5.6. Параметризация классов

В качестве параметров в классе могут быть не только числовые объекты, но и (другие) классы. Модифицируем ранее рассмотренную модель `StarOfResistors` добавлением в декларации резисторов спецификатора **replaceable**

```
model StarOfResistors
  replaceable Resistor R1(R = 100);
  replaceable Resistor R2(R = 200);
  replaceable Resistor R3(R = 300);
equation
  connect(R1.p, R2.p);
  connect(R1.p, R3.p);
end StarOfResistors;
```

Это означает, что в позициях объектов `R1`, `R2`, `R3` вместо объектов класса `Resistor` можно размещать объекты (с теми же именами) других классов. При этом стандартно используется полиморфизм, что означает, что замещающие классы должны иметь структуру класса `Resistor` в качестве своей подструктуры. В этом смысле замещающие классы должны быть подтипами типа `Resistor`. В частности, производный для данного класса класс является в нашем контексте его подтипом.

Теперь цепь можно усовершенствовать, заменив некоторые из резисторов на их варианты, зависящие от температуры проводника

```
model StarOfResistorsModified
  extends StarOfResistors(
    redeclare HeatResistor R1(
      R = 100, RT = 0.1, T = T - 1),
    redeclare HeatResistor R2(
      R = 200, RT = 0.1, T = T - 5));
end StarOfResistorsModified;
```

Здесь температура двух резисторов зависит от температуры окружающей среды.

Поскольку модель `HeatResistor` является подтипом, хотя и не наследником, модели `Resistor`, то в языке `Modelica` возможно заменить два объекта `R1` и `R2` на объекты более «узкого» типа `HeatResistor`. Для этого нужно использовать ключевое слово **redeclare**. При замещении в объекте замещающего класса можно переопределить его параметры и атрибуты. В отличие от числового параметра, задаваемого спецификацией **parameter**, формальный класс-параметр задается спецификацией **replaceable**, в то время как замещающий фактический класс-параметр задается спецификацией **redeclare**.

Если какой-либо из параметров замещаемого класса не должен подвергаться модификации, то модификация блокируется при помощи ключевого слова **final**. Например, размещение в модели `StarOfResistors` декларации

```
replaceable Resistor R3(final R = 300);
```

превращает параметр `R` в константу (только в контексте модели `StarOfResistors`).

Имеется еще одна возможность ограничения полиморфизма, разрешающая замещать формальный класс-параметр не произвольными фактическими классами-параметрами, а только классами-наследниками заданного суперкласса. С использованием этой возможности пример нашего параметризованного класса можно представить в виде

```
model StarOfResistors
  replaceable Resistor R1(R = 100) extends OnePort;
  replaceable Resistor R2(R = 200) extends OnePort;
  replaceable Resistor R3(R = 300);
equation
  connect(R1.p, R2.p);
  connect(R1.p, R3.p);
end StarOfResistors;
```

Тогда объект резистора `R1` можно заменить на объект конденсатора с тем же именем при помощи кода

```
model StarOfResistorsModified
  extends StarOfResistors(
    redeclare Capacitor R1(C = 0.001));
end StarOfResistorsModified;
```

Это возможно, поскольку модель `Capacitor` так же, как и `Resistor`, является наследником модели `OnePort`. Можно обобщить процедуру замещения и замещать не отдельные объекты (в нашем примере это `R1`, `R2`), а все объекты специально введенного для этой цели фиктивного (формального) класса, имя которого и будет замещаться. При этом должны соблюдаться правила полиморфизма. Например, рассмотренная цепь `StarOfResistors` может быть модифицирована следующим образом

```
model StarOfResistors
  replaceable ResistorToBeReplaced = Resistor;
  Resistor R3(final R = 300);
protected
  ResistorToBeReplaced R1(R = 100);
  ResistorToBeReplaced R2(R = 200);
```

```

equation
  connect (R1.p, R2.p);
  connect (R1.p, R3.p);
end StarOfResistors;

```

Теперь вместо указания инкапсулированных (недоступных) объектов R1 и R2 можно заместить их фиктивный класс ResistorToBeReplaced на его подтип HeatResistor при помощи кода

```

model StarOfResistorsModified
  extends StarOfResistors(
    redeclare model ResistorToBeReplaced =
      HeatResistor;
end StarOfResistorsModified;

```

В данном случае мы имеем дело с метаобъектами (роль объектов играют типы): формальный метапараметр ResistorToBeReplaced замещается фактическим метапараметром HeatResistor.

Следуя руководству [23] рассмотрим еще один поучительный пример из области гидравлики. Вместо модели электрического соединения Pin, через которое «течет» электрический ток, рассмотрим модель гидравлического соединения Stream, через которое течет поток жидкости. Соответствующий коннектор имеет код

```

connector Stream
  SI.Pressure p;
  flow SI.VolumeFlowRate v;
end Stream;

```

в то время как модель сосуда с жидкостью Tank, имеющего одну трубу для втекания жидкости и одну трубу для её вытекания, представляется на языке Modelica в виде

```

model Tank
  parameter SI.Area S = 1;
  replaceable connector PipeStream = Stream;
  PipeStream Inlet;
  PipeStream Outlet;
  SI.Height Level;
equation
  S*der(Level) = Inlet.v + Outlet.v;
  Outlet.p = Inlet.p;
end Tank;

```

где предполагается, что сосуд имеет вертикальную цилиндрическую форму с горизонтальным сечением площади S, а текущий уровень жидкости хранится в переменной Level. Так что текущий объем жидкости в сосуде равен  $S \cdot \text{Level}$ , и скорость «прибывания» жидкости в

силу постоянства  $S$  удовлетворяет тождеству (в обозначениях языка Modelica)

```
der (S*Level)  $\equiv$  S*der (Level)
```

Приток жидкости обеспечивается за счет двух труб. Он равен алгебраической сумме притоков, обеспечиваемых каждой трубой. Ситуация аналогична рассмотрению алгебраической суммы электрических токов, «втекающих» в звездообразное разветвление проводников электрической цепи. Первое уравнение в секции поведения, начинающейся после ключевого слова **equation**, соответствует балансу массы жидкости. Второе уравнение, задает равенство давлений в трубах.

Доопределим модель Tank до модели, учитывающей вместе с переносом массы также и перенос тепла

```
connector HeatStream
  extends Stream;
  SI.Temperature T;
end HeatStream;
model HeatTank
  extends Tank(
    redeclare connector PipeStream = HeatStream);
  SI.Temperature T;
equation
  S*der (Level*T) = Inlet.v*Inlet.T + Outlet.v*Outlet.T;
  Outlet.T = T;
end HeatTank;
```

Здесь предполагается, что жидкость в сосуде «мгновенно» подвергается идеальному перемешиванию и имеет во всех точках одну и ту же температуру. То же самое относится и к трубам, доставляющим жидкость в танк. В общем случае все три температуры могут быть различными, однако второе уравнение модели обеспечивает равенство температур в сосуде и в выходной трубе.

Первое же уравнение в секции поведения представляет уравнение баланса тепла. Здесь принято еще одно упрощающее предположение: теплоемкость жидкости считается одинаковой во всех точках объема. Слева в уравнении стоит производная от величины, пропорциональной теплу, накопленному в сосуде. Справа стоит алгебраическая сумма величин, пропорциональных теплу, притекающему в сосуд через трубы.

Если не использовать механизм наследования, а в явной форме, как это сделал бы компилятор, полностью описать последнюю построенную модель, то после работы компилятора по «раскручиванию» схемы наследования мог бы получиться следующий код

```

model HeatTank1
  parameter SI.Area S = 1;
  connector PipeStream
    SI.Pressure p;
    flow SI.VolumeFlowRate v;
    SI.Temperature T;
  end PipeStream;
  PipeStream Inlet;
  PipeStream Outlet;
  SI.Height Level;
  SI.Temperature T;
equation
  S*der(Level) = Inlet.v + Outlet.v;
  Outlet.p = Inlet.p;
  S*der(Level*T) = Inlet.v*Inlet.T + Outlet.v*Outlet.T;
  Outlet.T = T;
end HeatTank1;

```

где «сливаются» все декларации и уравнения базовых и производных классов.

## 5.7. Массивы

Массивы предназначены для хранения объектов, имеющих матричную структуру, и задаются в виде

```

SI.Position[3] Point "Vector of coordinates";
SI.Velocity[3] PointVel "Linear velocity";
SI.Acceleration[3] PointAcc "Linear acceleration";
Real[3, 3] Rot "Matrix of transformation";
Real[3, 3, 3] Tens "3D tensor";

```

или в виде, привычном для языков программирования

```

SI.Position Point[3];
SI.Velocity PointVel[3];
SI.Acceleration PointAcc[3];
Real Rot[3, 3];
Real Tens[3, 3, 3];

```

При этом, используя методику, описанную выше, можно создавать и использовать новые матричные типы. Например, можно создать общий тип матрицы преобразования трехмерного вещественного пространства

```

type Transformation = Real[3, 3];

```

и использовать его для декларации



```
Transformation Rotation;
```

В качестве менее тривиального примера рассмотрим цепочку определений, задающих типы объектов, используемых в динамике трехмерного движения систем тел

```
type Position = Real(unit = "m");  
type Position3 = Position[3];  
type Force = Real(unit = "N");  
type Force3 = Force[3];  
type Torque = Real(unit = "N.m");  
type Torque3 = Torque[3];
```

Теперь можно задать связь типа, например, жесткой заделки одного твердого тела в другом в виде коннектора

```
connector MBSCut  
Transformation R;  
Positions r0;  
flow Force3 f;  
flow Torque3 t;  
end MBSCut;
```

Этот коннектор задает срез-сечение, по которому тела «заделаны» друг в друга (твердотельно приклеены друг к другу). Здесь объект  $R$  определяет  $3 \times 3$ -матрицу поворота, задающую ориентацию площадки среза относительно некоторой общей координатной системы отсчета  $O_0x_0y_0z_0$  так, что система координат  $Oxyz$ , жестко связанная со срезом, имеет ось  $Ox$  перпендикулярной площадке среза. Объект  $r0$  определяет вектор, задающий положение начала  $O$  координатной системы  $Oxyz$  относительно системы  $O_0x_0y_0z_0$ .

Пара объектов  $f$  и  $t$  образует силовой винт, в общем случае направленный произвольно, или моттор [5], действующий на тело через заданный срез и состоящий из вектора силы  $f$  и вектора крутящего момента  $t$  пары сил. Так же, как электрический ток в коннекторе  $Pin$  и объемный расход жидкости в коннекторе  $Stream$ , моттор в коннекторе  $MBSCut$  (MBS — аббревиатура от MultiBody System) определен в виде объекта потокового типа. В самом деле, сумма мотторов, действующих в срезе со стороны каждого из связанных тел по третьему закону Ньютона равна нулевому моттору. Заметим, что в контексте теории бондграфов если в качестве потоковой переменной используется моттор, то в качестве переменной «усилия» нужно использовать кинематический винт относительного движения связанных тел, состоящий из вектора относительной линейной скорости и вектора относительной угловой скорости. Ясно, что кинематический винт в контексте механической связи имеет физический (а не формальный) смысл,

если коннектор задает не связь типа заделки (как в нашем случае), а связь, допускающую относительные перемещения тел.

Развивая этот пример, определим модель невесомой жесткой перемычки с двумя срезами для жестких заделок в соседние с перемычкой тела (Рис. 5.7.1)

```

model Bar
  parameter Position3 r = {1, 0, 0};
  MBSCut a "Left cut";
  MBSCut b "Right cut";
equation
  b.R = a.R;
  b.r0 = a.r0 + a.R*r;
  zeros(3) = a.f + b.f;
  zeros(3) = a.t + b.t - cross(r, a.f);
end Bar;

```

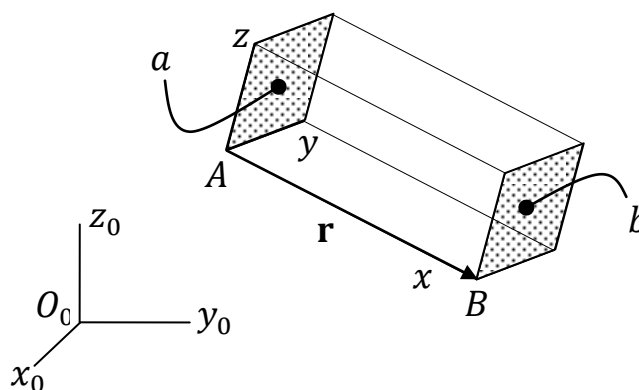


Рис. 5.7.1. Перемычка между двумя сечениями  $a$  и  $b$ .

Перемычка имеет два параллельных среза: левый —  $a$  и правый —  $b$ . В силу первого уравнения модели матрицы поворота от базовой системы координат  $O_0x_0y_0z_0$  к системе координат  $Axuz$  левого среза и  $Bxuz$  правого среза совпадают. Таким образом, срез  $b$  получается из среза  $a$  поступательным перемещением на вектор  $\mathbf{r}$ . Это свойство реализовано во втором уравнении модели. Здесь объекты  $a.r0$  и  $b.r0$  задают координаты точек  $A$  и  $B$  соответственно.

Третье уравнение модели задает баланс сил, действующих на перемычку. Здесь **zeros** является встроенным оператором языка, генерирующим вектор, состоящий из трех нулевых компонент. Четвертое, последнее, уравнение модели задает баланс крутящих моментов, действующих на перемычку и приведенных к центру сечения  $b$ . В самом деле, главный момент всех, действующих на перемычку, сил относительно центра сечения  $b$  составляется из двух моментов пар сил, приложенных к сечениям и хранящихся в объектах  $a.t$  и  $b.t$ , и мо-

мента силы, приложенной в центре сечения  $a$  и хранящейся в объекте  $a.f$ . Вектор положения центра сечения  $a$  относительно центра сечения  $b$  равен  $-\mathbf{r}$ . Поэтому момент силы в последнем уравнении вычисляется по формуле  $-\mathbf{r} \times \mathbf{f}_a$ . В этом уравнении встроенный оператор **cross** вычисляет векторное произведение (трехмерных векторов).

Уравнения балансов сил и моментов являются векторными. Каждое порождает по три скалярных уравнения (в проекциях на связанные с телом оси  $Axyz$  или  $Bxyz$ ). Первое уравнение модели — матричное. Оно порождает девять скалярных уравнений. Второе уравнение — также векторное. Оно генерирует еще три скалярных уравнения. Всего в модели получается восемнадцать скалярных алгебраических уравнений.

Операции с матрицами и векторами в Modelica во многом аналогичны соответствующим операциям в известных вычислительных средах Mathematica, Matlab или Mathcad. Операторы  $+$ ,  $-$ ,  $*$ ,  $/$  над массивами можно понимать как векторизации в Mathcad. Одномерный массив ( $n$ -вектор) может быть построен при помощи конструкции вида

```
a = {expr1, expr2, ..., exprn};
a двумерный массив ( $m \times n$ -матрица) — конструкции вида
M = {expr11, expr12, ..., expr1n;
     expr21, expr22, ..., expr2n;
     ...
     exprm1, exprm2, ..., exprmn};
```

где под  $\text{expr}_i$  или  $\text{expr}_{ij}$  понимается какое-либо из допустимых скалярных выражений языка Modelica. В отличие от Mathcad и C/C++ индексация элементов массивов начинается не от нуля, а от единицы. Как и в Mathcad, в Modelica имеется операция взятия подматрицы  $M[i_1:i_2, j_1:j_2]$ , где при помощи индексных пар компилятор устанавливает поддиапазоны строк и столбцов данной матрицы, используемые для построения подматрицы. В языке также имеются встроенные матричные операторы **zeros**, **ones**, **identity**, **transpose** и другие.

Класс на этапе моделирования может не иметь информации о размерностях матрицы. Эта информация может изменяться между отдельными прогонами модели. На этот случай имеется функция **size**( $M, i$ ) для определения фактического максимального значения  $i$ -ого индекса матрицы. При этом сама матрица должна задаваться без размеров в виде  $M[:, :]$ .

В качестве иллюстрации рассмотрим одну из моделей блока преобразования сигналов типа «вход-выход» [23], используемую в те-

ории систем автоматического регулирования (Рис. 5.7.2). Подобная модель имеется в стандартной библиотеке классов `Blocks` языка.

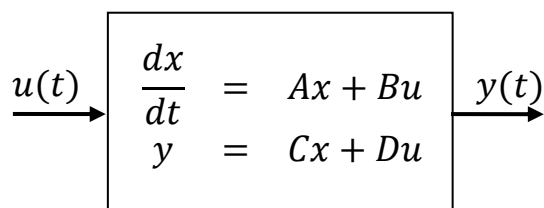


Рис. 5.7.2. Линейная управляемая система

В этой модели  $u(t)$  — входной сигнал, а  $y(t)$  — вырабатываемый линейной системой выходной сигнал. Код модели имеет вид

```
block StateSpace "Linear state space system"
  parameter Real A[:, size(A, 1)];
  parameter Real B[size(A, 1), :];
  parameter Real C[:, size(A, 1)];
  parameter Real D[size(C, 1), size(B, 2)] =
    zeros(size(C, 1), size(B, 2));
  input Real u[size(B, 2)];
  output Real y[size(C, 1)];
protected
  Real x[size(A, 1)](start = zeros(size(A, 1)));
equation
  der(x) = A*x + B*u;
  y = C*x + D*u;
end StateSpace;
```

Матрица  $A$  предполагается квадратной. Поэтому в её декларации количество столбцов полагается равным количеству строк `size(A, 1)`. Все остальные размерности также согласованы в соответствии с системой ДАУ Рис. 5.7.2. В данном частном случае матрица  $D$ , используемая при формировании выходного вектора  $y$ , полагается равной нулю. Двоеточие в позиции размерности означает, что величина соответствующей размерности будет взята из объекта, порожденного классом `StateSpace`, как это сделано, например, в следующей модели

```
block LinearSystemExample
  StateSpace LinSys(A = [0.12, 2; 3, 1.5],
                    B = [2, 7; 3, 1],
                    C = [0.1, 2]);
equation
  LinSys.u = {time, sin(time)};
end LinearSystemExample;
```

Классы библиотеки `Blocks` соответствуют блокам, используемым для построения моделей систем в Simulink. Применение таких

моделей упрощает разрешение проблемы причинности, локализуя эту проблему в пределах одного объекта, созданного из класса категории **block**. С другой стороны, применение только этих средств не может обеспечить уровень сложности потенциальных задач, для которых требуется применение техники так называемого аказуального моделирования.

## 5.8. Циклы

В любом языке программирования циклические конструкции играют существенную роль, обеспечивая возможность реализации повторяющихся вычислений различного типа. Хотя Modelica является декларативным (а не императивным) языком в нем также реализована возможность циклов. Однако циклические конструкции здесь интерпретируются двояко.

Во-первых, циклы можно реализовывать в традиционном императивном стиле. Для этого в Modelica имеются классы специальной категории **function**, которые будут рассмотрены ниже. Во-вторых, применение циклов в декларативных конструкциях означает построение периодически организованных систем уравнений так, как это реализуется в макропроцессорах. Фактически это приводит к интенсивному использованию символьного алгебраического препроцессора, применяемого в компиляторах с Modelica для построения системы ДАУ модели при её подготовке к прогону.

Такого рода применение циклов предпочтительно тогда, когда желательно каким-нибудь образом запараллелить вычислительный процесс, состоящий из однотипных вычислений, или требуется реализовать каскады типовой обработки информации в модели. Используем накопленный запас моделей для иллюстрации этих соображений.

Рассмотрим сначала задачу исследования динамики цепочки связанных последовательно элементарных осцилляторов, рассмотренных выше, в зависимости от количества грузов в цепи. Само это количество будем считать параметром задачи, изменяемым от одного прогона модели к другому. Модель представим в виде, иллюстрирующем применение циклической конструкции **for**

```
model Chain
  parameter Integer n = 2 "Number of masses";
  parameter Real[n] c = {1, 2} "Spring stiffnesses";
  ElementaryOscillator[n] Node;
initial equation
  for i in 1:n loop
```

```

    Node[i].c = c[i];
  end for;
equation
  Node[1].u = sin(time);
  for i in 2 : n loop
    Node[i - 1].f = Node[i].c * (Node[i].x - Node[i].u);
    Node[i].u = Node[i - 1].x;
  end for;
  Node[n].f = 0;
end Chain;

```

В другом примере используем компоненту Tank, представляющую модель сосуда с трубой для втекания и трубой для вытекания жидкости. Поведение этого класса достаточно тривиально. В технических системах хранения и обработки жидкостей применяются значительно более сложные модели. Типичная конфигурация емкостей в таких системах — линейные цепи объектов, соединенные трубами. Такие конфигурации используются, например, в системах очистки сточных вод, где последовательно расположенные биореакторы обеспечивают поэтапную очистку воды от загрязнений различного типа. Аналогичная методика может быть применена при разработке моделей химических и биохимических процессов, в которых узлами обработки потоков реагентов могут быть химические реакторы различной конструкции. Характер взаимосвязи объектов класса Tank изображен на Рис. 5.8.1.

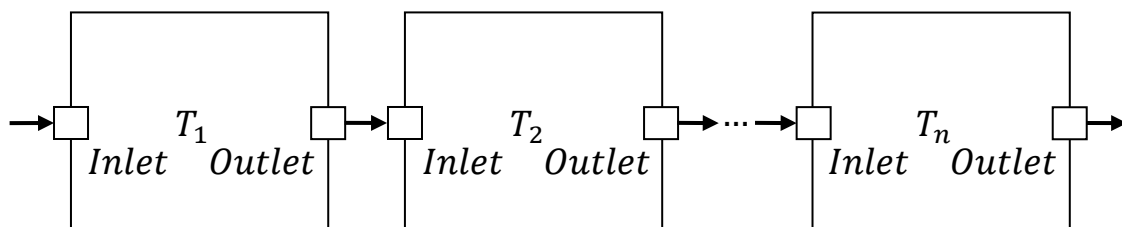


Рис. 5.8.1. Линейная цепь связанных емкостей

Соответствующий фрагмент кода на Modelica имеет вид

...

```

parameter Integer n = 2 "Number of tanks";

```

```

    Tank[n] Reservoir;
    ...
equation
    ...
    for i in 1 : n - 1 loop
        connect(Reservoir[i].Outlet,
                Reservoir[i + 1].Inlet);
    end for;
    ...
end Chain;

```

К этому фрагменту нужно добавить ДАУ, описывающие процессы химической кинетики в реакторах и уравнения баланса жидкости в соединениях.

## 5.9. Императивное кодирование

При моделировании непрерывно протекающих процессов не всегда удастся обойтись одними лишь уравнениями. Типичным примером являются гибридные динамические системы, в которых для описания динамики требуется применить аппарат управления событиями. В случаях же моделирования работы дискретных устройств часто желательно вместо уравнений использовать операторы присваивания.

В языке Modelica предусматривается использование внешних функций, подготовленных на языках программирования C и Fortran 77. Для обеспечения естественного и надежного интерфейса с такими функциями из моделей пользователя также желательно иметь в арсенале разработчика средства обычного языка программирования. Этого можно достичь, используя вместо секции **equation** секцию **algorithm**. Эта секция используется в классах специально для этого предназначенной категории **function** (допускается также использование секции **algorithm** в одной модели — классе категории **model** — с секцией **equation**). В качестве стандартного примера рассмотрим функцию вычисления полинома по схеме Горнера

```

function Poly
    input Real[:] c "Array of coefficients";
    input Real x "Argument";
    output Real f "Result";
algorithm
    f := c[1];

```

```

for i in 1 : size(c, 1) loop
  f := f*x + c[i + 1];
end for;
end Poly;

```

В языке Modelica функции применяются в стандартном математическом смысле: в них нет внутренней памяти для запоминания состояния вычислительного процесса и нет возможности создать какие-либо побочные эффекты, которые стандартным образом применяются, например, в C/C++.

Таким образом, в секции **algorithm** можно использовать общую циклическую конструкцию вида

```

for <Итератор> loop
  <Алгоритм>
end for;

```

допускающую произвольное гнездование. Под конструкцией <Алгоритм> здесь подразумевается произвольный фрагмент императивного кода на языке Modelica. Кроме уже рассмотренного, имеется также еще один вид цикла

```

while <Условие> loop
  <Алгоритм>
end while;

```

В императивном варианте Modelica допускает также стандартные конструкции ветвления вида

```

if <Условие> then
  <Алгоритм1>
[
else
  <Алгоритм2>
]
end if;

```

Ветвления можно использовать, как в секциях **equation**, так и в секциях **algorithm**. Условное выражение определяется стандартно

```

if <Условие> then <Выражение1> else <Выражение2>

```

Возвращаясь к примеру функции Poly заметим, что функции могут вызываться как в императивном, так и в декларативном коде. Объекты, декларированные в функции при помощи спецификации **output**, должны вычисляться внутри неё и возвращаются в выражение в том месте, где стоит идентификатор имени вызываемой функции. Входные параметры можно передавать внутрь функции одним из двух способов: либо по имени переменной входного параметра, ис-



пользуя это имя в качестве ключевого слова, либо по порядку расположения значений передаваемых параметров, совпадающему с порядком перечисления входных переменных в описании функции. Функцию `Poly` можно вызвать либо в виде

```
y { := | = } Poly(c = {-1, 2, 100.2, 0.002}, x = 0.25);
```

либо в виде

```
y { := | = } Poly({-1, 2, 100.2, 0.002}, 0.25);
```

Конкретная реализация общих конструкций языка *Modelica*, в том числе циклических и условных конструкций, представленных выше, подробно описана в документе [22] спецификаций стандарта языка.

Заметим, что декларативный (уравнения) и императивный (операторы присваивания) коды языка по-разному выполняются во времени функционирования модели. Напомним, что модельное время всегда может быть получено при помощи встроенной переменной **time**. Уравнения декларативного кода действительны во все время функционирования модели, если только они не отключаются при помощи конструкции

```
if <Условие> then
  <Уравнения1>
else
  <Уравнения2>
end if;
```

позволяющей альтернировать отдельные уравнения или целые группы уравнений в моменты времени, динамически вычисляемые в процессе прогона модели. Императивный код алгоритмов выполняется всегда мгновенно по отношению ко времени модели.

## 5.10. Гибридные модели

В языке *Modelica* можно использовать непрерывные, дискретные или смешанные модели в пределах одного класса или одного объекта, в частности, в пределах объекта, соответствующего модели верхнего уровня. Можно применять разрывные функции. При этом желательно, чтобы в моменты перехода через разрывы производилось переключение уравнений. В противном случае солверы (численные решатели систем уравнений, задающих неявные функции) аварийно завершают вычислительный процесс с соответствующим диагностическим сообщением. При переключении систем уравнений монитор времени выполнения пытается заново сформировать задачу

Коши для следующего отрезка регулярности. Если это не удастся сделать, система также аварийно завершает процесс симуляции

Регулярность функций может нарушаться не только в случаях разрывов (первого рода), но и при наличии разрывов (тоже первого рода) у их производных. Заметим, что наличие разрывов второго рода у производных также обычно приводит к аварийному завершению вычислительного процесса.

Негладкие задачи, требующие при моделировании переключения между режимами гладкого поведения, встречаются в самых разнообразных технических приложениях. Достаточно вспомнить проблемы, связанные с сухим трением, где для задания различных режимов относительного проскальзывания и покоя тел применяется функция знака вида

$$\text{sgn}(x) = \begin{cases} -1 & \text{при } x < 0, \\ 0 & \text{при } x = 0, \\ 1 & \text{при } x > 0. \end{cases}$$

В реальных вычислениях, особенно в многомерных задачах, случай  $x = 0$  идентифицировать невозможно. Однако в задачах, связанных с сухим трением, оказалось возможным с приемлемой точностью [13] приближенно заменить функцию  $\text{sgn}(x)$  на её аппроксимацию (не в равномерной, а в интегральной функциональной метрике) вида

$$\text{sgn}_\delta(x) = \begin{cases} -1 & \text{при } x < -\delta, \\ x/\delta & \text{при } -\delta \leq x \leq \delta, \\ 1 & \text{при } x > \delta, \end{cases}$$

где параметр  $\delta > 0$  предполагается достаточно малым, характеризующим близость моделируемого движения к точному. Функция, аналогичная  $\text{sgn}_\delta(x)$ , часто используется в технических приложениях. В стандартной библиотеке `Blocks` классов языка `Modelica` имеется модель `Limiter`, реализующая преобразование сигнала  $u(t) \mapsto y(t)$  по формуле

$$y(u) = \begin{cases} u_{\min} & \text{при } u < u_{\min}, \\ u & \text{при } u_{\min} \leq u \leq u_{\max}, \\ u_{\max} & \text{при } u > u_{\max}. \end{cases}$$

Данная формула преобразования может быть реализована при помощи уравнения

```
y = if u < u_min then u_min
    else if u <= u_max then u else u_max;
```

или при помощи эквивалентной конструкции

```
if u < u_min then
  y = u_min;
else if u <= u_max then
  y = u;
```

```

else
    y = u_max;
end if;

```

в которой в зависимости от использованных условий альтернируются три уравнения.

## 5.11. Дискретные события и модели с дискретным временем

В языке Modelica реализованы две модели времени: а) непрерывная, неявно обеспечиваемая при помощи интеграторов дифференциальных уравнений и б) дискретная, реализуемая на фоне непрерывной при помощи встроенного оператора дискретизации **sample**, более подробно описываемого ниже. Оператор дискретизации генерирует события в последовательные моменты времени модели. Эти события могут быть обработаны при помощи специальной конструкции языка

```

when <Условие> then
    <Уравнения> | <Алгоритм>
end when;

```

предназначенной для асинхронной обработки событий. Событие наступает, если значение логического выражения в условии приобретает значение **true** (меняется от **false** к **true** — как в таких случаях говорят в схемотехнике — по переднему фронту сигнала, генерируемого выражением условия). Конструкция **when** по своим свойствам напоминает средства обработки во времени модели асинхронных событий, так называемых исключительных ситуаций (exceptions), применяемых в современных системах разработки программного обеспечения или аппаратных прерываний. Ни уравнения, ни алгоритмический код в этом случае не занимают времени модели. С точки зрения процесса моделирования они работают мгновенно.

Конструкция **when** может также эффективно использоваться при реализации гибридных автоматов. В общем случае гибридные автоматы могут быть достаточно сложными. При наступлении события в таком автомате изменения могут претерпеть не только правые части дифференциальных уравнений, но и структура и размерность динамической системы. Примеры подобных гибридных автоматов можно построить в динамике механических систем с неудерживающими связями, в случае контакта твердых тел с трением, когда связь может находиться в одном из трех состояний: а) отсутствие контакта, б) относительное проскальзывание тел, в) относительное качение тел.

Построим простейший пример гибридного автомата — динамическую модель подпрыгивающего над горизонтальной плоскостью упругого шарика массой  $m = 1$  и коэффициентом восстановления  $k \leq 1$ . Направим ось  $x$  вертикально вверх. Ограничение односторонней связи и дифференциальное уравнение движения шарика имеют соответственно вид

$$x \geq 0, \quad \ddot{x} = -1,$$

где величина ускорения свободного падения предполагается равной единице.

В рассматриваемой задаче механическая система может «находиться» в одном из двух возможных состояний: «Полет» с повторяющимися ударами и «На связи» (шарик покоится). Таким образом, в нормальной форме Коши систему дифференциальных уравнений движения шарика можно представить в виде

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= u(x_1, x_2), \end{aligned}$$

где в зависимости от состояния, которое, в свою очередь, зависит от точки в фазовом пространстве, сила  $u(x_1, x_2)$  может принимать одно из двух значений:  $-1$  в состоянии «Полет» и  $0$  в состоянии «На связи». При переходе из состояния в состояние необходимо также позаботиться о корректном присвоении начальных данных. Так что в «идеальной» модели автомата следует положить

$$u(x_1, x_2) = \begin{cases} -1 & \text{при } x_1 > 0 \text{ или } x_2 \neq 0, \\ 0 & \text{при } x_1 \leq 0 \text{ и } x_2 = 0, \end{cases}$$

Такая идеальная постановка приводит к бесконечной серии подскоков с асимптотически уменьшающейся высотой

$$h = \frac{x_2^2}{2},$$

что неизбежно приведет к аварийному завершению процесса моделирования в силу невозможности бесконечного измельчения шага интегрирования. Поэтому в качестве приближенного критерия выхода на связь уместнее было бы взять условие достаточной малости высоты подпрыгивания шарика

$$u(x_1, x_2) = \begin{cases} -1 & \text{при } x_1 > 0 \text{ или } x_2^2 \geq 2\varepsilon, \\ 0 & \text{при } x_1 \leq 0 \text{ и } x_2^2 < 2\varepsilon, \end{cases}$$

где  $\varepsilon$  является достаточно малым пороговым значением высоты подскока, ниже которой считается, что шарик «лег» на связь.

Код описанной модели автомата на языке Modelica имеет вид

**model** Hybrid

**parameter** Real Tol = 10<sup>^</sup>(-7) "Tolerance";

**parameter** Real k = 0.9 "Restitution coefficient";

Real[2] x "State vector";

```

Real u "Force";
Integer State;
initial equation
  State = if x[1] <= 0 and x[2]^2 < 2*Tol
    then 2 else 1;
algorithm
  when x[1] <= 0 and x[2]^2 >= 2*Tol then
    State := 1;
  end when;
  when x[1] <= 0 and x[2]^2 < 2*Tol then
    State := 2;
  end when;
equation
  der(x[1]) = x[2];
  der(x[2]) = u;
  u = if State == 1 then -1 else 0;
  when x[1] <= 0 and x[2]^2 >= 2*Tol then
    reinit(x[2], -k*x[2]);
  end when;
  when x[1] <= 0 and x[2]^2 < 2*Tol then
    reinit(x, {0, 0});
  end when;
end Hybrid;

```

Проанализируем более подробно получившуюся модель. Целочисленная переменная *State* предназначена для хранения информации о текущем состоянии гибридного автомата. Значение 1 соответствует состоянию «Полет», а значение 2 — состоянию «На связи». Секция **initial equation** предназначена для формирования начального значения переменной состояния в зависимости от начальных данных задачи Коши.

Входной алфавит автомата состоит из двух символов: «Удар» и «Посадка на связь». Символ «Удар» поступает на «считывающую головку» автомата при возникновении условия

$$x[1] \leq 0 \text{ and } x[2]^2 \geq 2 \cdot Tol,$$

обрабатываемого одновременно в двух конструкциях **when**. Одна конструкция расположена в секции императивных операторов **algorithm** и обеспечивает корректное формирование состояния после обработки входного символа. В случае удара сохраняется состояние «Полет». Вторая конструкция **when**, выполняемая синхронно с первой, расположена в секции поведения **equation** и обеспечивает корректный пересчет начальных данных (скорости после удара) с учетом модели Ньютона.

Символ «Посадка на связь» так же, как и символ «Удар», поступает на вход автомата при выполнении условия

$$x[1] \leq 0 \text{ and } x[2]^2 < 2 \cdot Tol,$$

когда скорость сближения с поверхностью оказывается меньше заданного порогового значения, что в рамках данной модели считается достаточным для выхода системы на связь. В секции **algorithm** переменная состояния принимает значение, соответствующее состоянию «На связи», а в секции поведения начальные данные подправляются так, чтобы шарик после этого покоился на плоскости.

На рассматриваемом примере мы видим, что при обработке событий можно использовать встроенный оператор

```
reinit (<Переменная состояния>, <Значение>);
```

для смены (мгновенной) значений непрерывно меняющихся переменных при моделировании.

На этом примере демонстрируется построение на языке Modelica не только гибридного, но и любого конечного автомата на основе реализации при помощи конструкции **when** матрицы переходов состояний и блоков обработки очередного символа. В этом стиле можно реализовать модели драйверов устройств и любых процессов асинхронной обработки событий (подсистем операционных систем, контроллеров, сетевых протоколов и др.).

На языке Modelica по умолчанию время модели «течет» непрерывно. Однако имеется возможность реализовать также модель дискретного времени. Для этого служит уже упоминавшийся выше встроенный оператор **sample**. Его использование имеет формат

```
sample (<Стартовый момент>, <Интервал>),
```

где <Стартовый момент> — это выражение, вычисляющее момент времени первого перехода значения булевозначного оператора **sample** из состояния **false** в состояние **true** (переднего фронта соответствующего логического сигнала). При обработке сигнала в конструкции **when** момент обратного перехода значения этого оператора не играет роли. Считается, что оператор **sample** порождает короткие логические импульсы. Последующие импульсы происходят через промежутки времени длительности <Интервал>. Так что оператор **sample** порождает временную сетку (дискретную последовательность  $\{t_n\}_{n=0}^{n=\infty}$  временных меток) вида

$$t_n = \text{Стартовый момент} + n \cdot \text{Интервал}, \quad n \geq 0, \quad n \in \mathbb{Z}_+.$$

Для иллюстрации рассмотрим пример подсчета количества импульсов оператора **sample**, приходящихся по отдельности на отрицательные и на неотрицательные значения входного сигнала  $u(t)$

```
model Count
```

```
  parameter SI.Time t0 = 1 "Start time";
```

```

parameter SI.Time dt = 1 "Time interval";
parameter Real Shift = 0.3;
Integer m(start = 0) "Number of negatives";
Integer n(start = 0) "Number of non-negatives";
Real Negs(start = 0) "Negatives percentage";
Real NNegs(start = 0) "Non-negatives percentage";
Real u;
equation
  u = sin(time) + Shift;
  when sample(t0, dt) and u < 0 then
    m = pre(m) + 1;
    Negs = m / (m + n);
  end when;
  when sample(t0, dt) and u >= 0 then
    n = pre(n) + 1;
    NNegs = n / (m + n);
  end when;
end Count;

```

Помимо подсчета числа «отрицательных» и «неотрицательных» импульсов вычисляется также доля каждого из этих выпадений. Графики этих долей показывают, что соответствующие кривые стремятся к пределам, соответствующим, очевидно, мере множеств отрицательности и неотрицательности заданной (периодической) функции. При подсчете числа импульсов в блоках **when** обработки событий использовался еще один встроенный оператор: **pre**. Этот оператор вычисляет левый предел переменной его аргумента в момент обрабатываемого события

$$\mathbf{pre}(x) = \lim_{t \rightarrow t'_-} x(t),$$

где  $t = t'$  предполагается моментом наступления события.

Таким образом, в Modelica по умолчанию предполагается, что для переменной во времени величины  $x$  значения  $x(t)$  и  $\mathbf{pre}(x)$  могут в общем случае различаться, что соответствует разрыву первого рода. Однако в согласии с требованиями теорем существования и единственности дифференциальных уравнений с разрывными правыми частями [15] требование непрерывности справа для функций переменных состояния сохраняется.

Конструкцию **when** можно эквивалентным образом описать помощи встроенного булевозначного оператора **edge** (<Условное выражение>). Этот оператор можно определить в виде

$$\mathbf{edge}(x) \equiv x \text{ and not } \mathbf{pre}(x);$$

Таким образом,  $\mathbf{edge}(x) = \mathbf{true}$  только в момент изменения значения логической переменной (выражения)  $x$  от значения **false** к

**true**. В самом деле, в точке непрерывности  $x = \text{pre}(x)$ . Поэтому  $\text{edge}(x) = \text{false}$ . При переходе от **true** к **false** имеем  $\text{pre}(x) = \text{true}$  и  $x = \text{false}$ . Поэтому снова получаем

```
edge(x) = false and not true = false;
```

И, наконец, при  $\text{pre}(x) = \text{false}$  и  $x = \text{true}$  имеем

```
edge(x) = true and not false = true;
```

Поэтому в языке Modelica конструкция вида

```
when <Условие> then
  . . .
  x = <Выражение>;
  . . .
end when;
```

эквивалентна конструкции

```
if edge(<Условие>) then
  . . .
  x = <Выражение>; // переменная может иметь скачок
  . . .
else
  . . .
  x = pre(x); // переменная сохраняет непрерывность
  . . .
end if;
```

Таким образом, в секции поведения (**equation**) конструкция **when** не имеет логической альтернативы в виде конструкции типа **else**. Однако в секции операторов (**algorithm**) возможна конструкция вида

```
when <Условие1> then
  <Операторы1>
elseif <Условие2> then
  <Операторы2>
elseif <Условие3> then
  <Операторы3>
  . . .
end when;
```

В связи с использованием в языке Modelica механизма моделирования дискретного времени и дискретных событий помимо переменных типов **Boolean**, **Integer**, **String**, имеющих естественную дискретность, существует также возможность использовать дискретный аналог вещественных переменных. Для этого в декларации таких



переменных следует добавить спецификацию **discrete**, например в виде

```
discrete Real x;
```

Такая спецификация означает, что вещественная переменная может меняться только в дискретные моменты времени, и соответствующая функция времени является кусочно-постоянной. Это свойство может быть полезно при построении моделей динамических систем, реализуемых в виде итерируемых отображений (каскадов).

Заметим в заключение, что упоминание в конструкциях **if** и **when** условных выражений автоматически генерирует внутреннюю переменную модели, для которой отслеживаются события перемены знака. Такое отслеживание состоит в выполнении процедуры дополнительной локализации момента перемены знака, что приводит к расходованию (может быть значительному) вычислительных ресурсов. Если же конструкция **if** применяется только для вычисления значений функций и её условное выражение не предназначено для генерации каких-либо событий, то расходования упомянутых ресурсов можно избежать, применяя функцию **noEvent**, запрещающую процедуру локализации для условных выражений, оказавшихся в пределах аргумента этой функции. Данная мера может заметно повысить быстродействие разрабатываемой модели. Например, если мы хотим задать переменную  $y$  как функцию модуля от переменной  $x$ , то лучше всего это сделать в виде

```
y = if noEvent(x < 0) then -x else x;
```

## 5.12. Библиотеки классов

Разрабатываемые пользователем динамические модели технических систем обычно образуют наборы классов со сложно переплетенными отношениями наследования, инстанциации и пр. В этих наборах можно выявить одну или несколько иерархий, которые позволяют структурировать моделируемую задачу или класс задач. По традиции объектно-ориентированного программирования такой набор принято называть библиотекой классов.

Структурированной библиотекой моделей легче управлять, её легче модифицировать и использовать разделение труда специалистов различных направлений, разрабатывающих модели в рамках различных технических дисциплин. Для организации таких библиотек предназначена специальная разновидность класса **package** — модель контейнерного типа, предназначенная для хранения других моделей, в том числе вложенных контейнеров. Начинать проектиро-

вание моделей рекомендуется с создания личных библиотек требуемой для нового проекта структуры.

Ситуация с контейнерными классами **package** напоминает файловую систему, где папки (каталоги) содержат документы (файлы) и другие папки, образуя, таким образом, иерархию папок файловой системы. При этом аналогично файловой системе в библиотеке классов возникает проблема видимости имен моделей (и их объектов). Именно так устроена стандартная библиотека языка Modelica. Например, структура библиотеки аналоговых компонент электрических цепей имеет вид

```
package Modelica
. . .
package Blocks
. . .
end Blocks;
package Constants
. . .
end Constants;
package Electrical
package Analog
package Examples
. . .
end Examples;
package Basic
. . .
model Resistor
. . .
end Resistor;
. . .
end Basic;
. . .
end Analog;
. . .
end Electrical;
. . .
end Modelica;
```

Извне библиотеки (пакета) компонента может быть достигнута при помощи квалифицированного имени, с указанием имен подклассов, разделенных точками. Например, компонента электрического со-

противления извне стандартной библиотеки будет найдена по имени `Modelica.Electrical.Analog.Basic.Resistor`. Если же класс, в котором нужно установить (инстанцировать) объект класса `Resistor` находится в пакете `Analog`, то к имени этого класса следует обращаться при помощи имени `Basic.Resistor`.

Рассмотрим различные частные случаи видимости классов и объектов и доступа к ним в библиотеках языка. Как уже было замечено, доступ можно осуществлять по полному имени с подробной иерархической квалификацией. В качестве альтернативы можно использовать простые (несоставные) имена в специально построенном виртуальном «плоском» пространстве имен. Для этого все потенциально используемые в проекте имена можно присоединить к списку имен, видимых напрямую. Напрямую же видны имена, расположенные в текущем пакете и в пакетах, его содержащих. Для операции присоединения имен используется оператор `import` так, как это уже было показано выше в примере, в котором имя библиотеки `Modelica.SIunits` типов физических размерностей объявляется эквивалентным короткому имени `SI`, что позволяет объявлять типы физических размерностей при помощи короткого префикса `"SI."` вместо длинного `"Modelica.SIunits."`. Различные способы использования оператора импортирования имен в случае размещения объекта класса `Resistor` рассмотрены в следующем примере.

Вариант 1:

```
import Modelica.Electrical.Analog.Basic;  
.  
.  
.  
Basic.Resistor R;
```

Вариант 2:

```
import Bas = Modelica.Electrical.Analog.Basic;  
.  
.  
.  
Bas.Resistor R;
```

Вариант 3:

```
import Modelica.Electrical.Analog.Basic.*;  
.  
.  
.  
Resistor R;
```

## 5.13. Трехмерная анимация моделей

Средства трехмерной анимации моделей подвижных объектов, описанные в данном разделе, не относятся к стандарту языка `Modelica` — это средства компилятора `Dymola`. Однако для удобства данный

материал размещен в разделе языка, поскольку реализация анимации существенно опирается на свойства соответствующей компоненты, применяемой в упомянутом компиляторе.

Будем считать, что в трехмерном пространстве движется твердотельный объект, с которым жестко связана подвижная система координат  $Oxyz$  (Рис. 5.13.1). Пространственная ориентация этой координатной системы задается при помощи ортогональной матрицы  $S$ , столбцы которой состоят из координат её базисных ортов. Кроме того, предполагается, что положение начала координат (точки  $O$ ) системы  $Oxyz$  задается вектором  $\mathbf{R}$ , а характерная точка самого геометрического трехмерного объекта смещена относительно начала подвижной системы на вектор  $\mathbf{r}_0$ . Вектор  $\mathbf{R} = (X, Y, Z)^T$  задан относительно неподвижной координатной системы  $O'x'y'z'$ , а вектор  $\mathbf{r}_0$  — относительно подвижной системы  $Oxyz$ .

Неподвижная система координат совпадает с координатами сцены окна анимации. Произвольная точка движущегося тела, задаваемая в подвижной системе  $Oxyz$  радиусом-вектором  $\mathbf{r} = (x, y, z)^T$ , в неподвижной системе координат сцены будет задаваться радиусом-вектором  $\mathbf{r}' = (x', y', z')^T = \mathbf{R} + S\mathbf{r}$ .

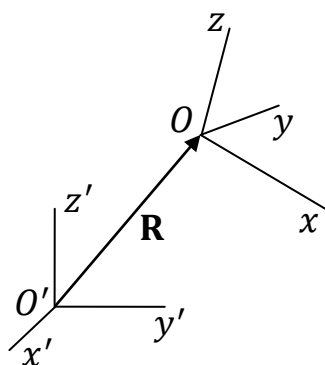


Рис. 1.13.1. Конфигурация координатных систем.

В компиляторе Dymola для визуализации движущихся тел или их фрагментов применяются библиотека геометрических примитивов, из которых и составляются сложные геометрические формы. Список этих примитивов имеет вид: “box” (параллелипипед), “sphere” (трехосный эллипсоид), “cylinder” (цилиндр с эллиптическим основанием), “cone” (конус с эллиптическим основанием), “pipe” (труба эллиптического сечения), “beam” (балка), “gearwheel” (зубчатое колесо), “arrow” (объемная стрелка) и другие примитивы [17, 18]. Эллиптичность перчисленных геометрических объектов получается при помощи растяжения/сжатия фигур вдоль собственных координатных осей.

С каждой геометрической формой ассоциирована система координат, которая может двигаться (в соответствии с уравнениями модели) относительно базовой системы координат, движущейся заданным образом относительно координат сцены или покоящейся относительно неё. Эта ассоциированная координатная система предполагается жестко связанной с одним из виртуальных твердых тел модели. В этом случае графический образ формы будет перемещаться на сцене анимационного окна. В качестве примера рассмотрим модель физического маятника, имеющего форму качающегося «бруса». Код этой модели можно представить в виде

```

model Pendulum
  parameter SI.Mass m = 1;
  parameter SI.Length L = 1;
  parameter SI.Acceleration g = 9.81;
  parameter SI.Inertia J = m*L^2/3 "Thin beam";
  SI.Angle phi;
  SI.AngularVelocity omega;
  VisualShape Bar(
    r0 = {0, 0, 0},
    Shape = "box",
    Length = L,
    Width = 0.05,
    Height = 0.05,
    LengthDirection = {0, -1, 0},
    WidthDirection = {1, 0, 0},
    Material = {1, 0, 0, 0});
equation
  der(phi) = omega;
  J*der(omega) = -m*g*sin(phi);
  Bar.S = [cos(phi), -sin(phi), 0;
           sin(phi),  cos(phi), 0;
           0,          0, 1];
  Bar.r = {0, 0, 0};
end Pendulum;

```

Класс VisualShape имеет много параметров. Для описания кинематики трехмерного объекта важны: текущий радиус-вектор точки начала собственной системы координат тела (по отношению к системе координат сцены) Bar.r и ортогональная матрица текущего поворота объекта Bar.S. Расположение параллелепипеда, представляющего физический маятник, по отношению к собственной системе координат тела показано на Рис. 5.13.2.

В соответствии с кодом модели Pendulum длина параллелепипеда направлена в сторону, противоположную оси Оу. Ширина и высота приняты достаточно малыми и направлены вдоль двух других

осей собственной системы координат. Цвет формы кодируется при помощи параметра `Material`, первые три параметра кодируют цветовые компоненты объекта в порядке: красный, зеленый, синий. Интенсивность каждого цвета определяется действительным числом, задающим долю интенсивности в сумме долей всех трех цветов. Долей от единицы задается четвертый параметр, характеризующий степень отражения падающего на грань света: 0 — полное отсутствие отражения (полное поглощение), 1 — зеркальное отражение.

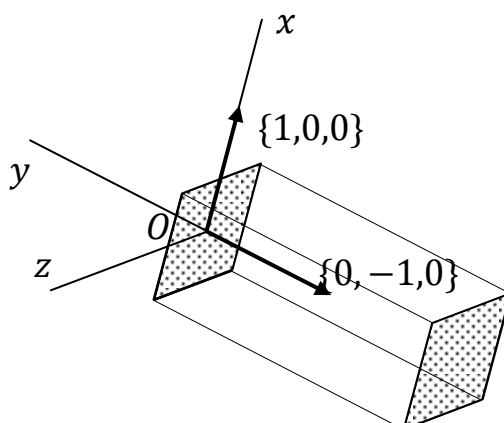


Рис. 5.13.2. Геометрическая конфигурация формы "box".

Движение маятника, подвешенного при помощи цилиндрического шарнира с осью, параллельной оси  $Oz$ , здесь задается при помощи двух переменных:  $\phi$  — угла отклонения центральной продольной линии бруса от вертикали и  $\omega$  — угловой скорости поворота бруса. Брус для простоты считается однородным стержнем. Поэтому момент инерции относительно оси  $Oz$  вычисляется по простой формуле  $J_z = mL^2/3$ .

# Глава 6. Dymola — визуальный компилятор с языка Modelica

## 6.1. Возможности Dymola

Для языка Modelica в настоящее время существует несколько компиляторов:

- Dymola (<http://www.3ds.com/products/catia/portfolio/dymola/>),
- MapleSim (<http://www.maplesoft.com/products/maplesim/>),
- MathModelica (<http://www.mathcore.com/products/mathmodelica/>),
- OpenModelica (<http://www.ida.liu.se/labs/pelab/modelica/OpenModelica.html>).

Рассмотрим свойства первого из перечисленных компиляторов. Перевод аббревиатуры Dymola — **D**ynamic **m**odeling **l**aboratory означает «Лаборатория динамического моделирования». Название соответствует назначению данного программного продукта. Его интерфейс обеспечивает инструментарий для поддержания работы виртуального испытательного стенда, на котором «размещается» строящаяся модель динамического процесса и на котором затем удобно анализировать всевозможные физические переменные так же, как если бы эти переменные измерялись при помощи датчиков.

Компилятор Dymola предназначен для моделирования поведения различных физических систем. В нем поддерживается построение иерархических моделей при помощи библиотек компонент. Эти компоненты визуально соединяются друг с другом в соответствии с законами взаимодействия моделируемых объектов. Модели таких объектов составляют богатый набор библиотек, обеспечивающих возможность симуляции процессов из многих областей техники. Выше были перечислены библиотеки стандартного набора языка Modelica. Помимо этого имеется значительное семейство библиотек, разработанных независимыми поставщиками. Часть библиотек находится в открытом доступе в Сети. Значительная же часть из них относится к коммерческим библиотекам и распространяется независимо. Эти библиотеки, предназначенные для решения отдельных прикладных задач, представляют собой наиболее ценное наполнение приложений языка Modelica.

При реализации компилятора были использованы две фундаментальные методологии моделирования динамических процессов: а) объектно-ориентированный подход к проектированию моделей; б) применение систем ДАУ, описывающих поведение моделей. К основным свойствам Dymola, выделяющим компилятор на фоне других

программных продуктов для моделирования динамических процессов, относятся:

- обеспечение управления сложными моделями технических систем;
- применение визуальной техники проектирования, ускоряющей разработку проектов;
- применение символьного препроцессора, обеспечивающего оптимальную редукцию исходной системы ДАУ;
- открытость при создании собственных моделей пользователя;
- открытый интерфейс с другими программами;
- трехмерная анимация;
- обеспечение симуляции процессов с привязкой к реальному масштабу времени.

Компилятор может работать в двух режимах:

1. разработки/подготовки к выполнению модели, этот режим в интерфейсе Dymola активизируется при помощи закладки Modeling в правом нижнем углу окна приложения (Рис. 6.1.1);

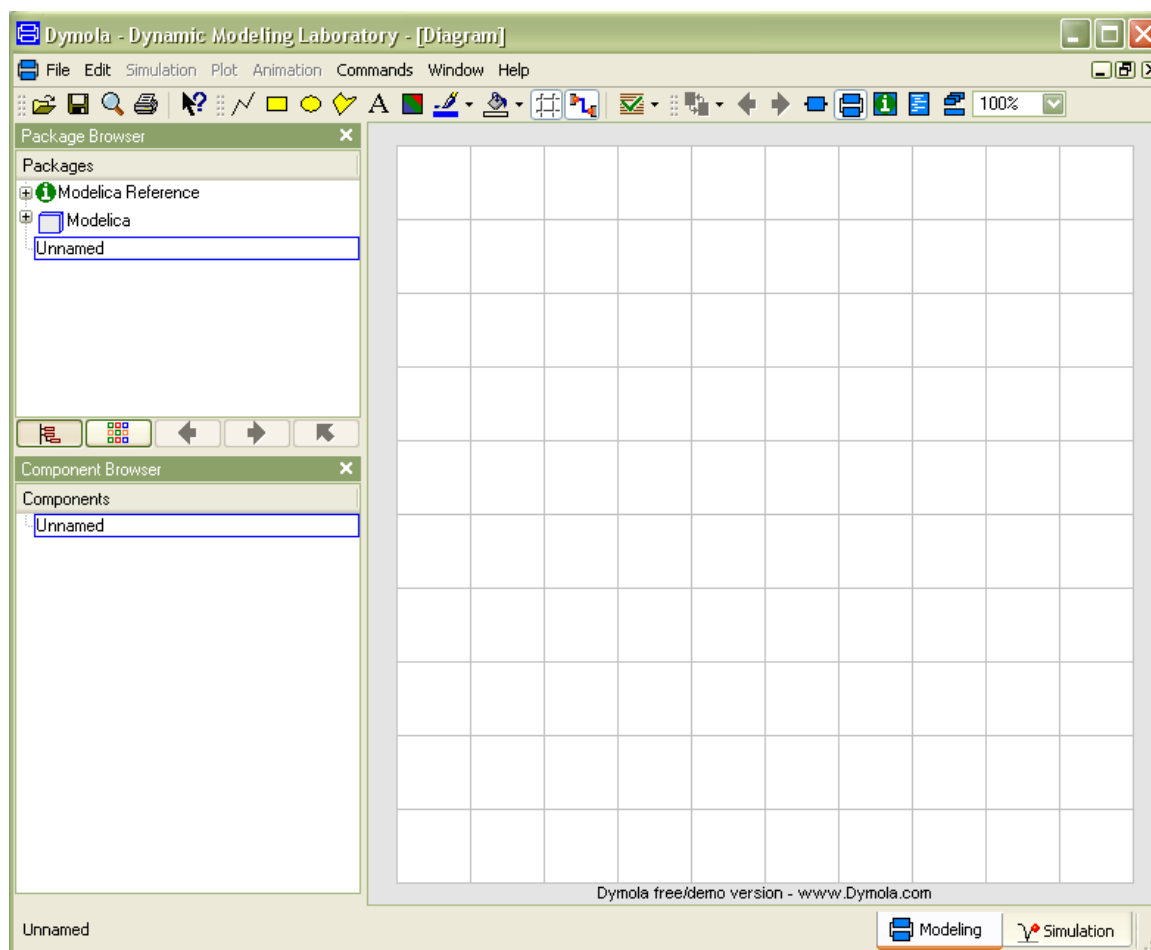


Рис. 6.1.1. Общий вид интерфейса Dymola в режиме Modeling.



2. трансляции/симуляции модели, этот режим активизируется при помощи закладки Simulation в правом нижнем углу окна Dymola (Рис. 6.1.2).

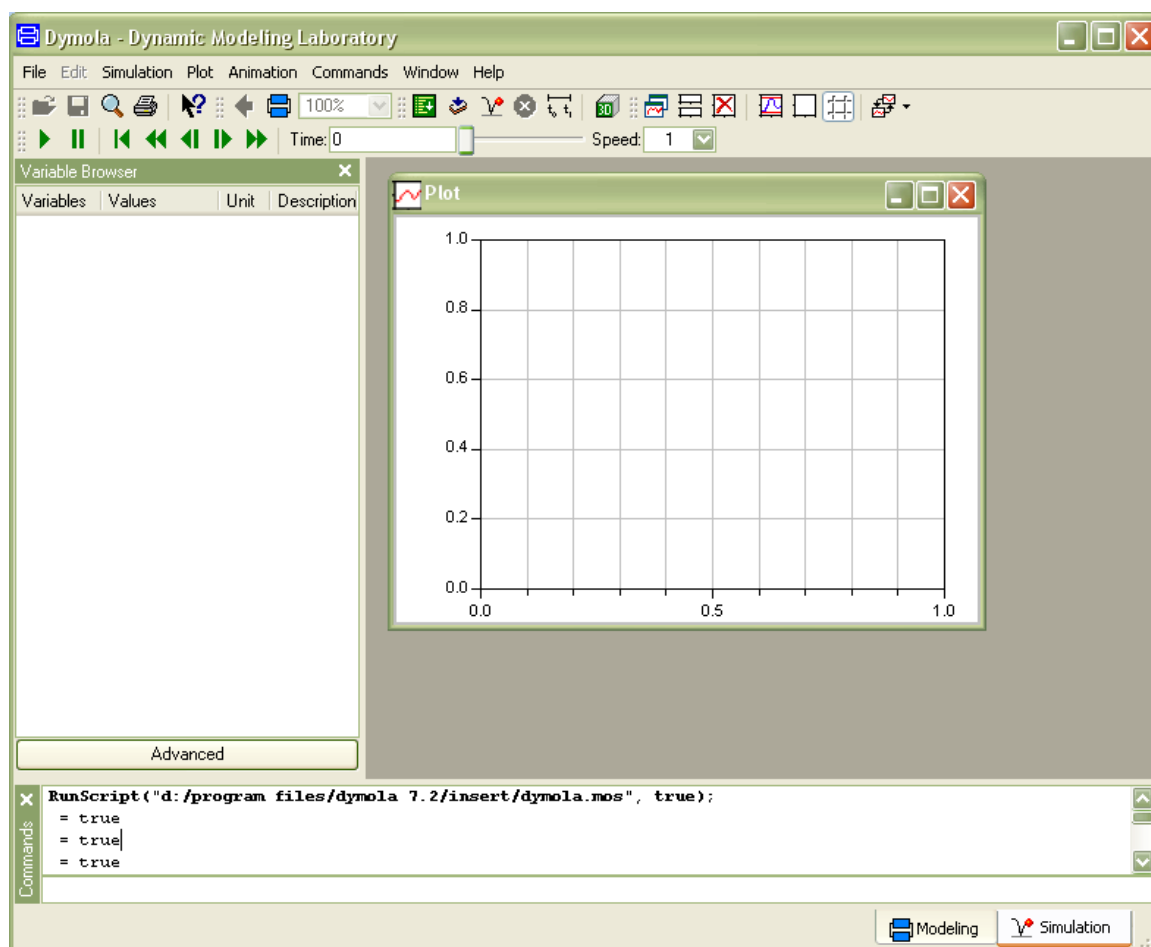


Рис. 6.1.2. Общий вид интерфейса Dymola в режиме Simulation.

Первый из двух режимов будем называть режимом разработки модели, а второй — прогона модели. Все дальнейшее описание будет посвящено изложению основных сценариев работы с компилятором. Этих сценариев всего два:

1. выполнение существующей модели;
2. построение (и последующее выполнение) новой модели.

Перед подробным описанием упомянутых выше режимов работы перечислим некоторые особенности реализации Dymola, обеспечивающие её функциональность:





- помимо развитого текстового редактора, обеспечивающего структурное и многоцветное представление текущей модели на языке Modelica, имеется графический редактор, позволяющий готовить графический материал для визуального представления разрабатываемых моделей (вторая слева панель инструментов на Рис. 6.1.1);

- все разрабатываемые модели имеют символьное (текстовое) представление на языке Modelica в виде файлов со стандартным расширением имени .mo;
- для обеспечения процесса разработки и выполнения моделей Dymola может импортировать графические файлы из других источников, причем для визуализации трехмерных объектов на сцене окна анимации импортируются трехмерные модели, созданные при помощи инструментов геометрического моделирования;
- существенной особенностью компилятора Dymola является наличие, помимо синтаксического и семантического анализатора языка Modelica, символьного алгебраического препроцессора, обеспечивающего оптимизирующую реорганизацию тотальной системы ДАУ компилируемой модели с одновременной редукцией индекса так, чтобы обеспечить генерацию (промежуточного) кода (на языке C) решения задачи Коши при помощи одного из имеющихся в арсенале (не менее шестнадцати) компилятора интеграторов;
- сгенерированный компилятором C-код модели может быть экспортирован и использован непосредственно в других приложениях;
- Dymola имеет развитые средства численного экспериментирования: трехмерную анимацию подвижных объектов модели и двумерные графики для анализа поведения переменных в зависимости от времени модели и друг от друга;
- обеспечение автоматической генерации документации для разрабатываемой модели.

Рассмотрим более подробно интерфейс Dymola для разработки моделей (Рис. 6.1.1). Он стандартен и напоминает интерфейс Проводника — файлового менеджера операционной системы Windows. Как и в Проводнике, слева мы видим браузер объектов. Только не объектов файловой системы (папок и документов), а объектов стандартной библиотеки языка, подключенных в данной установке компилятора. Этими объектами являются библиотеки и подбиблиотеки (аналоги папок Проводника), соответствующие описанной выше разновидности классов **package**, и модели остальных разновидностей, наполняющие упомянутые библиотеки. В нижней части этого браузера всегда располагаются личные библиотеки пользователя, «подгруженные» из его собственных mo-файлов.

Ниже подокна браузера библиотек расположено окно браузера объектов текущей активной модели. Текущая (сфокусированная) модель из браузера библиотек всегда отображается (как и в Проводнике) в основном окне справа. Каждая модель имеет четыре уровня

(слоя) представления (аналогично многослойному представлению ячеек Excel):

- Icon — пиктограмма, изображающая объект данного класса в визуальном представлении класса, содержащего этот объект («крышка» для «ящика» визуальной модели). Активизируется кнопкой  на инструментальной панели режима разработки (справа на Рис. 6.1.1);
- Diagram — визуальная модель данного класса, состоящая из графических икон объектов, соединенных коннекторами различного типа, («содержимое» «ящика» визуальной модели). Активизируется кнопкой  на инструментальной панели режима разработки;
- Documentation — окно документации данной модели, активизируется кнопкой  на инструментальной панели режима разработки;
- Modelica Text — невизуальное описание данной модели на языке объектно-ориентированного моделирования Modelica, представляющее самый «низкий», базовый слой описания модели, активизируется кнопкой  на инструментальной панели режима разработки;

Все подокна интерфейса компилятора, в том числе и отдельные инструментальные панели, могут свободно перемещаться в пределах его окна, менять свои размеры, удаляться/появляться и «пристыковываться» друг к другу в разных комбинациях. Подобные операции легко реализуются при помощи мыши или при помощи контекстного меню.

## 6.2. Создание проекта

Разработка любого проекта обычно начинается с создания новой личной библиотеки пользователя. Для этого нужно воспользоваться меню Dymola в виде следующей последовательности его пунктов: “File → New... → Package”, после чего откроется диалог, изображенный на Рис. 6.2.1. В первое поле вводим имя новой личной библиотеки (MyProject) и нажимаем кнопку “OK”.

Затем компилятор выдает предупреждение в виде окна на Рис. 6.2.2, в котором Dymola напоминает о возможности создания подбиблиотеки (если пользователь случайно об этом забыл). В данном случае следует нажать кнопку “Ассерт”, после чего пиктограмма созданной библиотеки появится в окне браузера библиотек так, как это показано на Рис. 6.2.3.

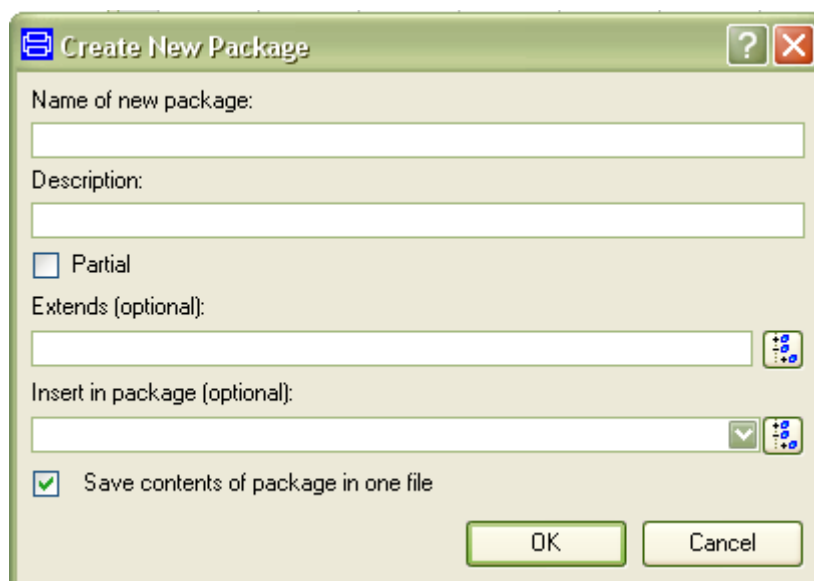


Рис. 6.2.1. Интерфейс создания новой библиотеки классов.

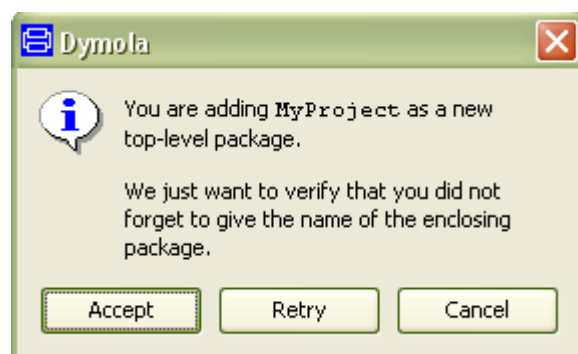


Рис. 6.2.2. Предупреждение о подбиблиотеке.

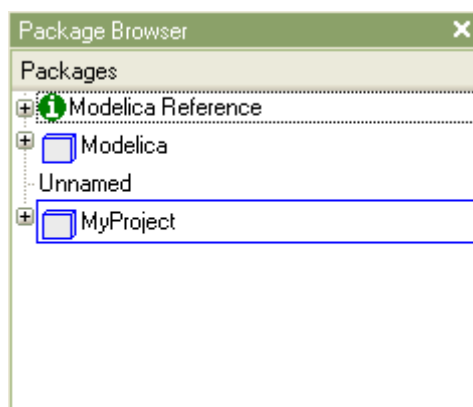


Рис. 6.2.3. Браузер библиотек.

Для создания новой модели и размещения её в только что созданной личной библиотеке нужно выполнить последовательность действий, аналогичную той, что была проделана только что при создании библиотеки `MyProject`: выполнить команду меню “File → New... → Model” и заполнить соответствующее окно диалога так, как это показано на Рис. 6.2.4, не забыв указать имя контейнера `MyProject` для вновь создаваемого класса `Model1`. После этого окно браузе-

ра библиотек и классов будет выглядеть так, как показано на Рис. 6.2.5, а в окне текущей модели справа будет представлено (пока еще пустое) содержимое этой модели.

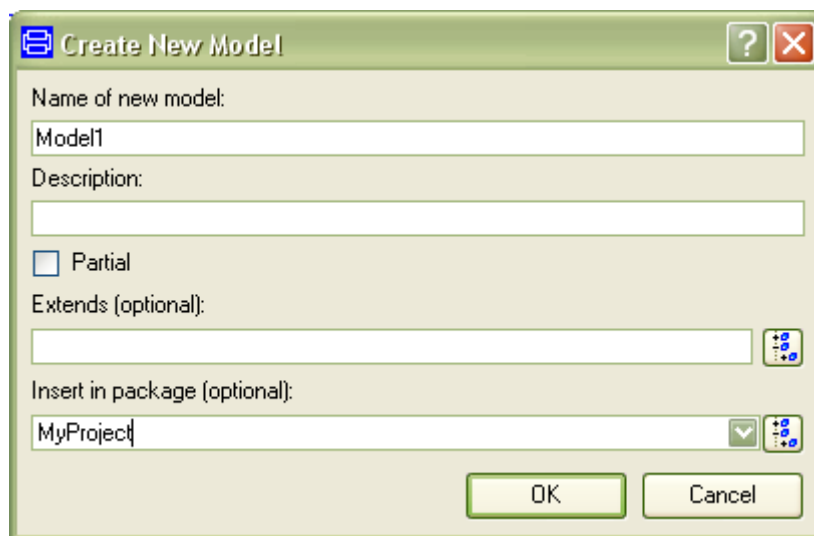


Рис. 6.2.4. Интерфейс создания новой модели.

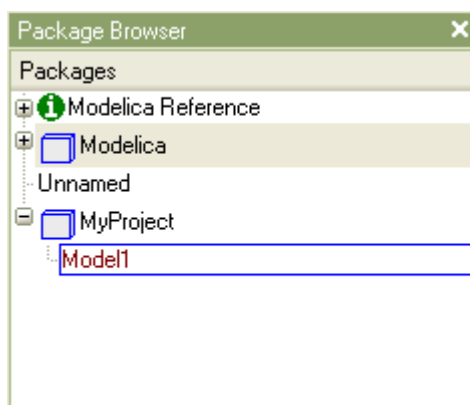


Рис. 6.2.5. Браузер библиотек с новой моделью.

Для обеспечения дальнейшей работы разработчика удобно сразу же сохранить (пока еще не наполненный) проект в папке пользователя. Для этого после установки фокуса на корневую библиотеку проекта используем команду меню “File → Save As...”. Получим диалог, изображенный на Рис. 6.2.6. После подтверждения имени проекта появится диалог Рис. 6.2.7, который также нужно подтвердить, после чего появится диалог сохранения из Рис. 6.2.8, в котором следует указать папку хранения проекта. Проект становится «связанным» с этой папкой, и все последующие сохранения проекта или его частей будут автоматически производиться в эту папку.

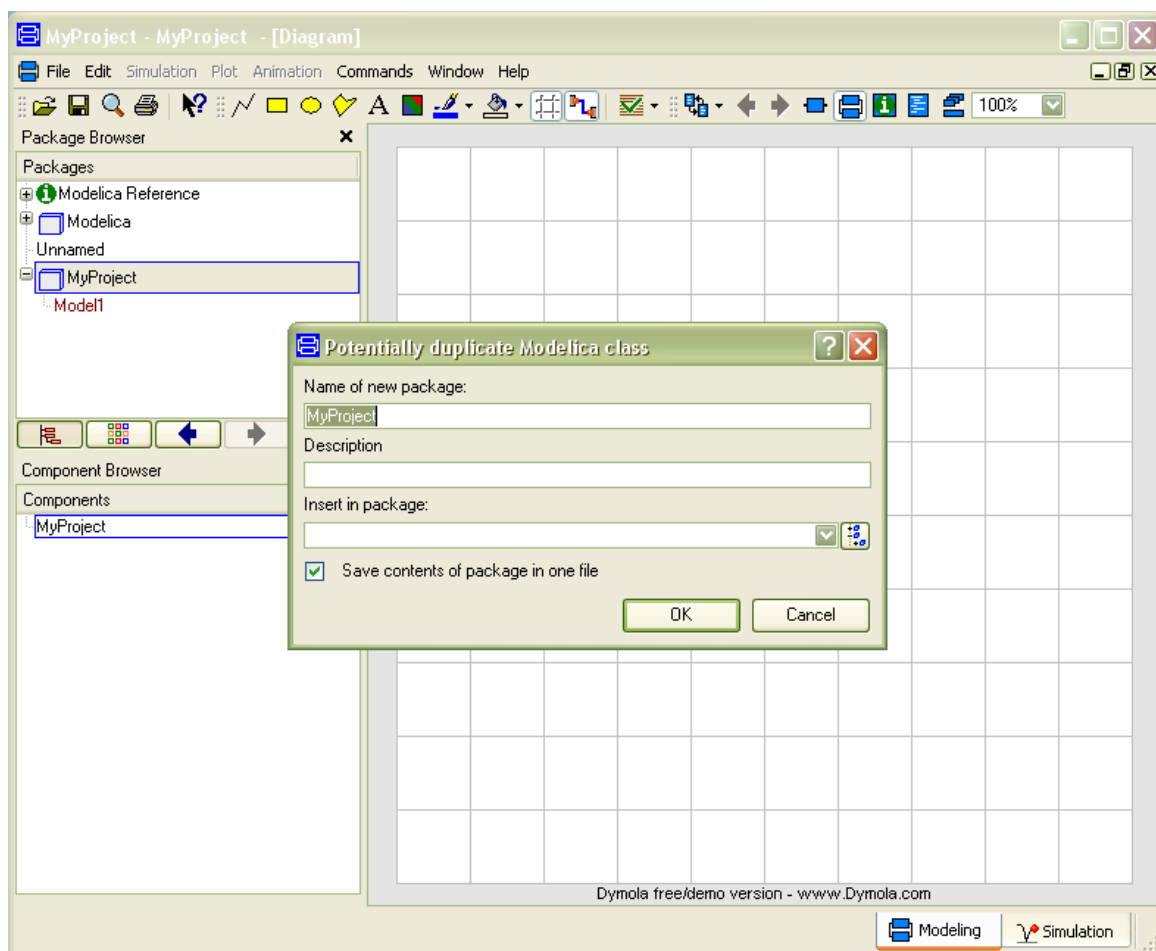


Рис. 6.2.6. Начало диалога сохранения проекта.

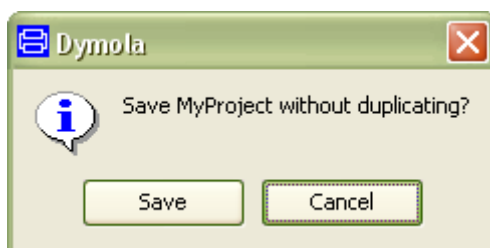


Рис. 6.2.7. Диалог подтверждения.

Для иллюстрации методики работы с Dymola частично позаимствуем из пособия [9] пример из области электротехники. Построим модель электрической цепи, показанной на Рис. 6.2.9 и состоящей из трех резисторов  $R_1, R_2, R_3$ , конденсатора  $C$ , индуктора  $L$ , источника питания  $E$  и заземления  $G$ . Для «сборки» из всех перечисленных элементов визуальной модели используем стандартную библиотеку `Modelica.Electrical.Analog.Basic` (для резисторов, конденсатора, индуктора и заземления) и библиотеку `Modelica.Electrical.Analog.Sources` (для источника питания) цепей электрических аналоговых сигналов.

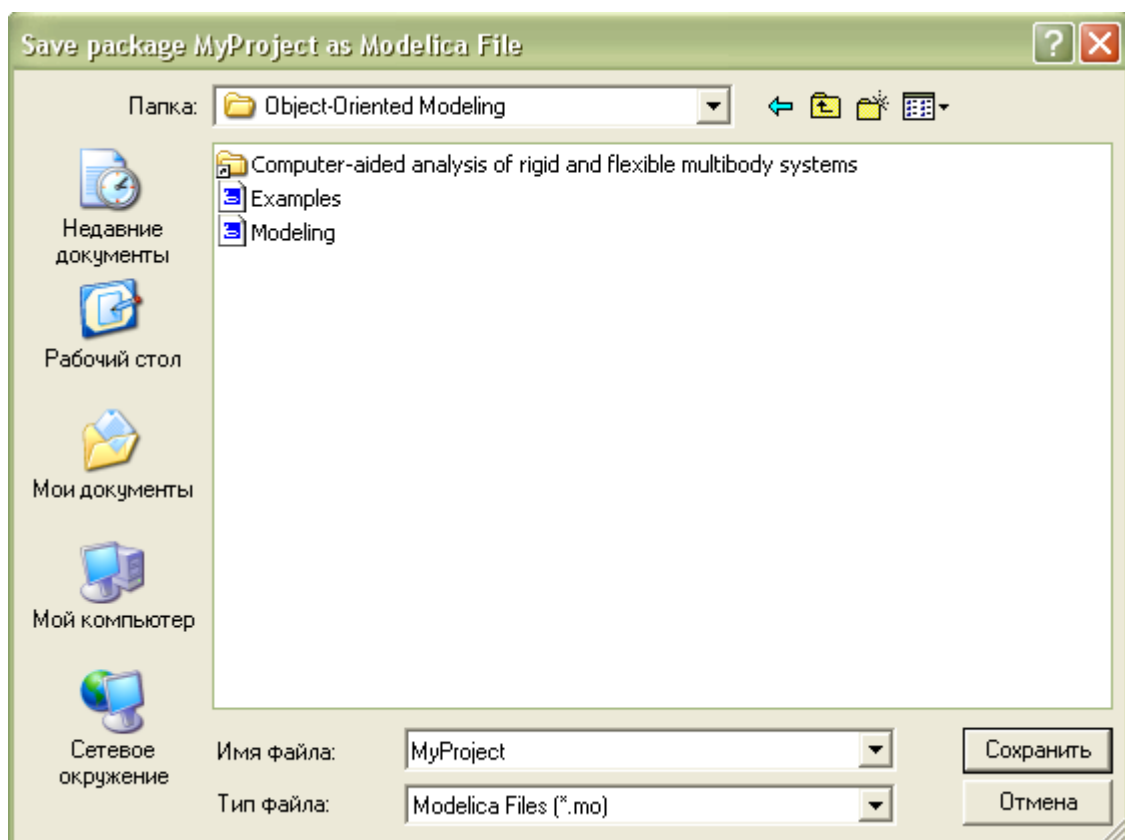


Рис. 6.2.8. Диалог указания папки хранения проекта.

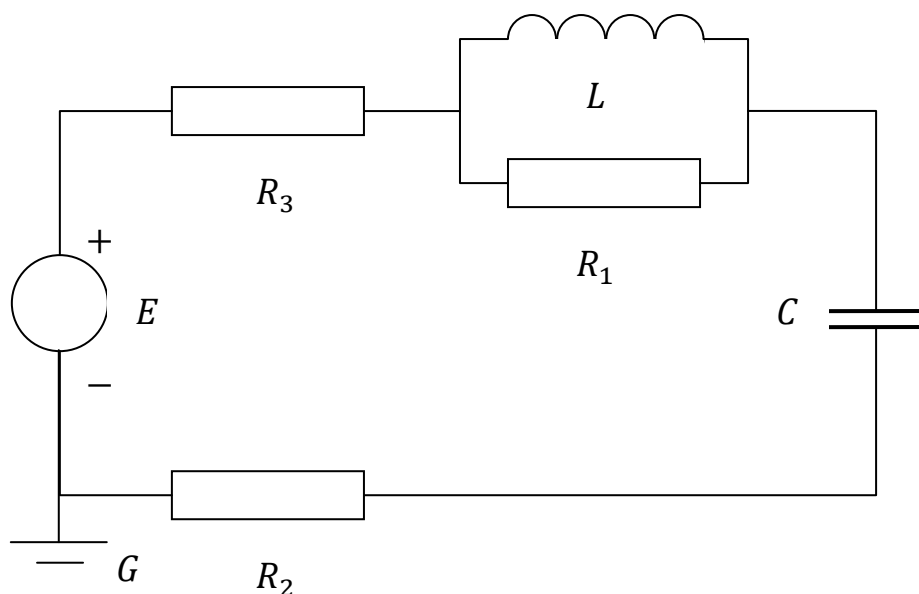


Рис. 6.2.9. Пример электрической цепи.

При помощи технологии перетаскивания компонент из окна библиотек в окно визуальной модели создадим в ней необходимый набор объектов из классов: Resistor, Capacitor, Inductor, Ground, SineVoltage, так, как это показано на Рис. 6.2.10.

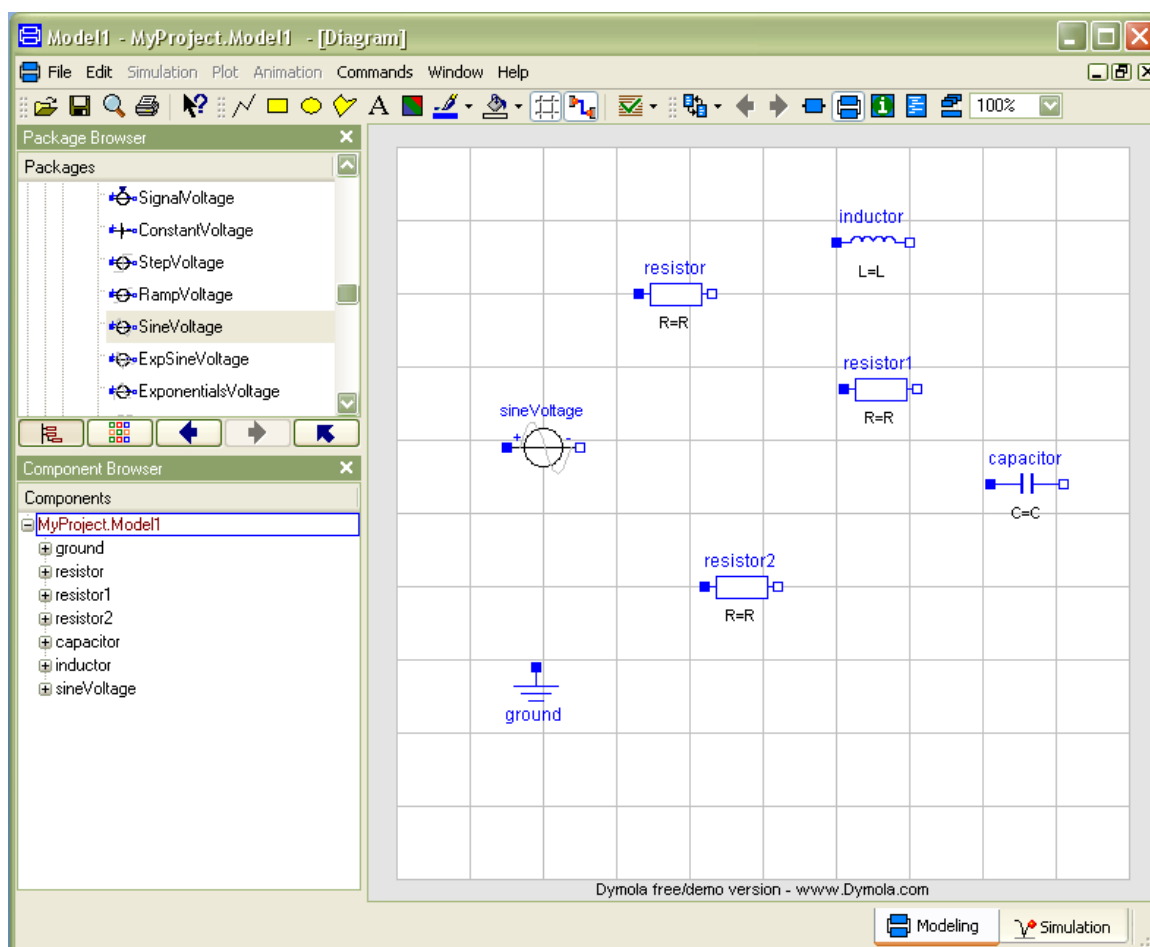


Рис. 6.2.10. Результат «превращения» классов в объекты.

Для соединения элементов электрической цепи «проводами» ориентируем некоторые из объектов визуальной модели в соответствии с Рис. 6.2.9 и предполагая, что электрический ток «течет» от положительного заряда к отрицательному. На визуальной модели Рис. 6.2.10 положительная контактная площадка закрашена синим цветом, а отрицательная — не закрашена. Итак, повернем икону конденсатора по часовой стрелке на прямой угол. Для этого сначала установим фокус на объект `capacitor`, после чего он будет выделен «габаритными» маркерами красного цвета, а затем выполним команду меню “Edit → Rotate –90”. Аналогичную операцию выполним с объектом источника питания `sineVoltage`. Объект `resistor2` нужно отразить зеркально относительно вертикальной оси (геометрической схемы). Для этого после установки фокуса нужно выполнить команду “Edit → Flip Horizontal”. В результате получим визуальную модель, представленную на Рис. 6.2.11.



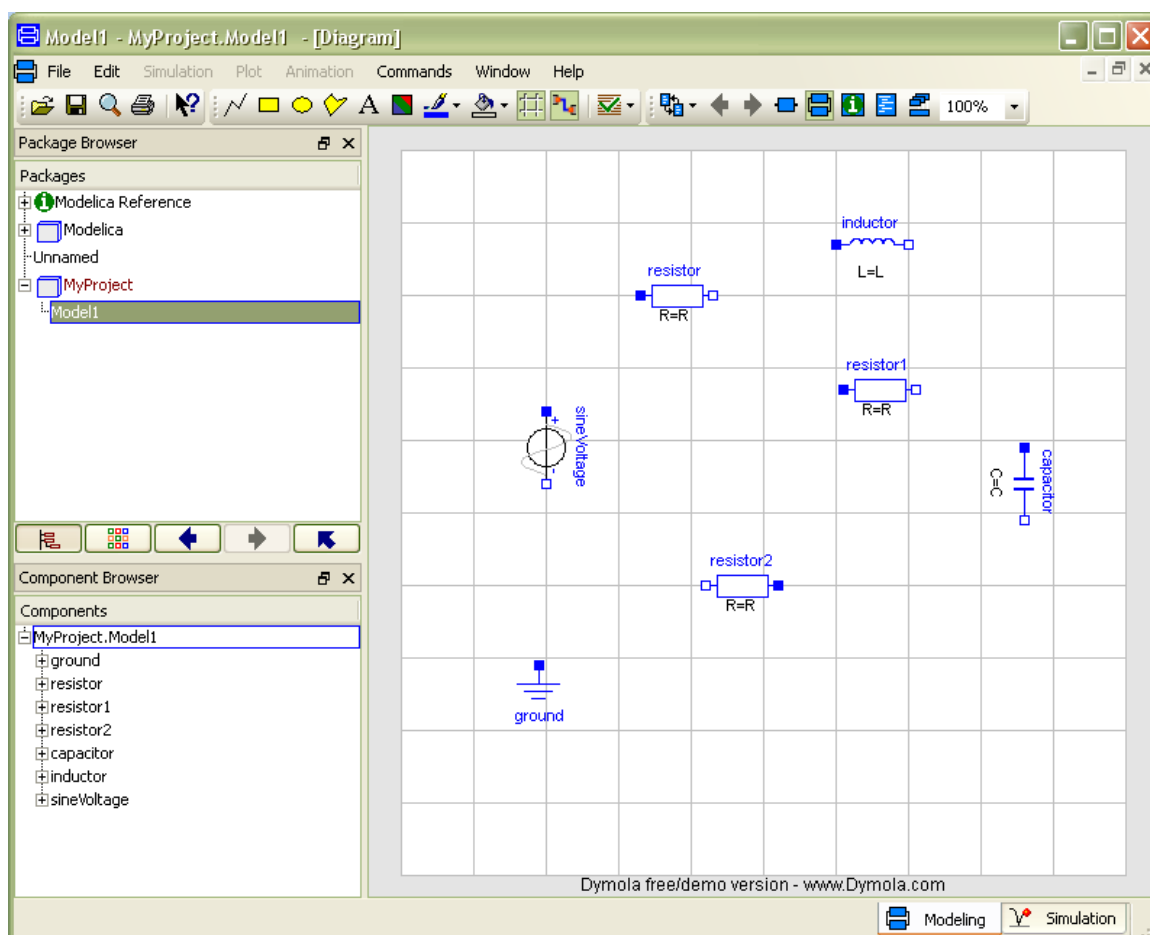


Рис. 6.2.11. Результат ориентирования объектов модели.

На следующем шаге следует выравнивать объекты визуальной модели и соединить их коннекторами протаскиванием мышью. При этом поворот коннектора на прямой угол обеспечивается разовым нажатием левой клавиши мыши вниз или вверх в зависимости от фактического номера поворота. Нажатие/отжатие левой клавиши мыши на значке порта компонента (объекта) означает начало/завершение построения коннектора. Чтобы отменить построение текущего коннектора, следует воспользоваться контекстным меню (нажать правую клавишу мыши и выбрать команду “Cancel connection”). Результирующая визуальная модель, полученная после выполнения всех операций соединения, представлена на Рис. 6.2.12.

Этап подготовки модели завершается присвоением каждому объекту его имени и значений его параметрам, указанным в задании. Для этого, после установления фокуса на соответствующий объект, нужно использовать пункт “Parameters...” контекстного меню или просто двойной щелчок левой кнопки мыши на иконе модели.

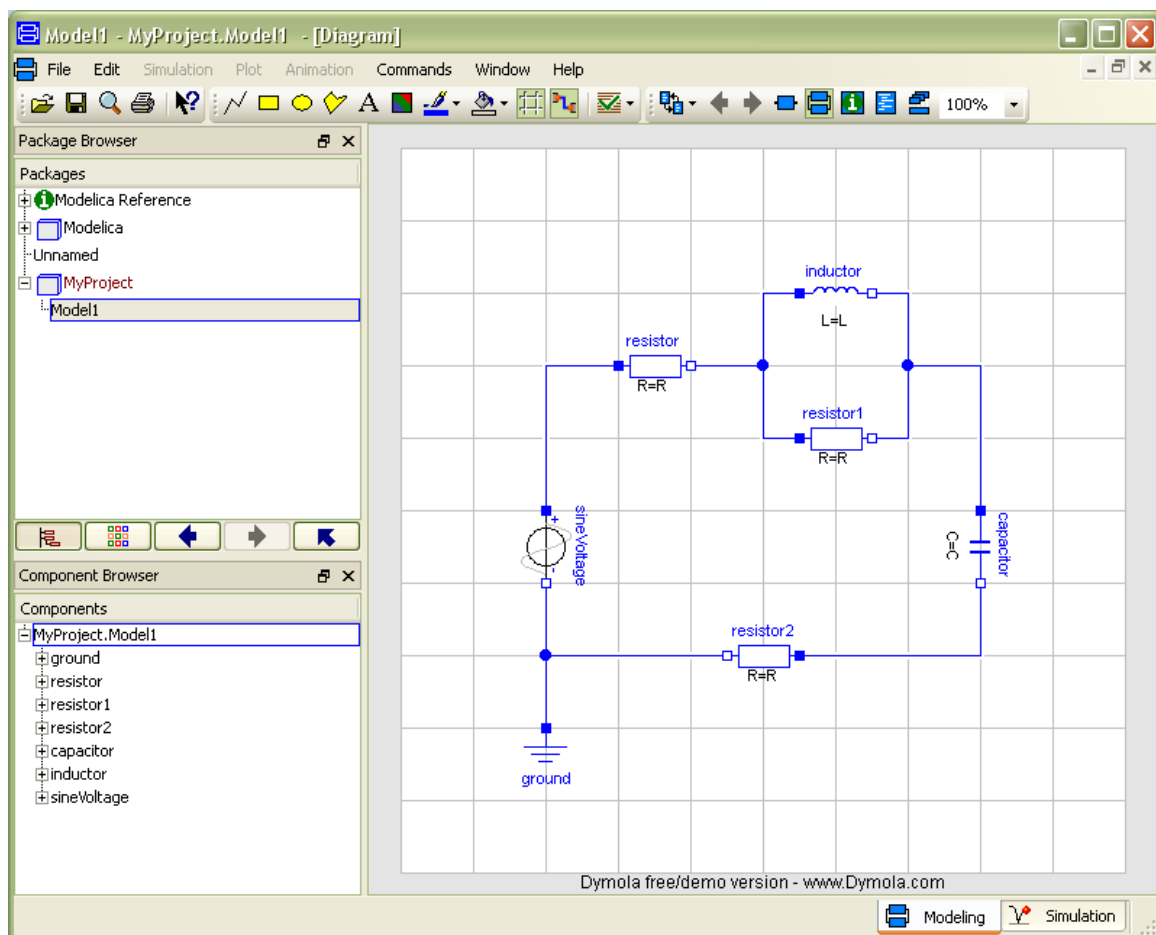


Рис. 6.2.12. Результат соединения объектов модели.

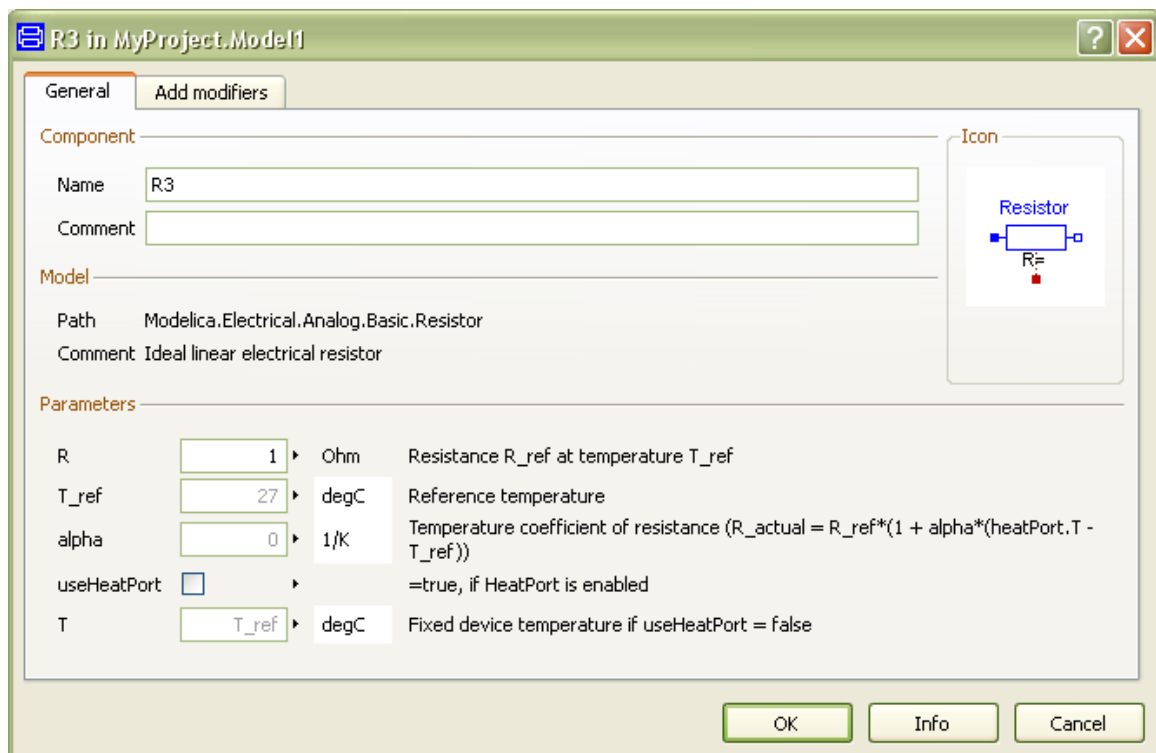


Рис. 6.2.13. Окно диалога модификации свойств объекта.

В появившемся диалоге (Рис. 6.2.13 для объекта `resistor`) установить требуемое имя в поле “Name” и значение параметра в требуемом поле (на Рис. 6.2.13 это поле “R” значения базового сопротивления для базового значения температуры в 27°C). Параметры источника переменного синусоидального питания устанавливаются следующими: амплитуда колебаний — 220В, частота колебаний напряжения — 50Гц. Результирующая визуальная модель показана на Рис. 6.2.14.

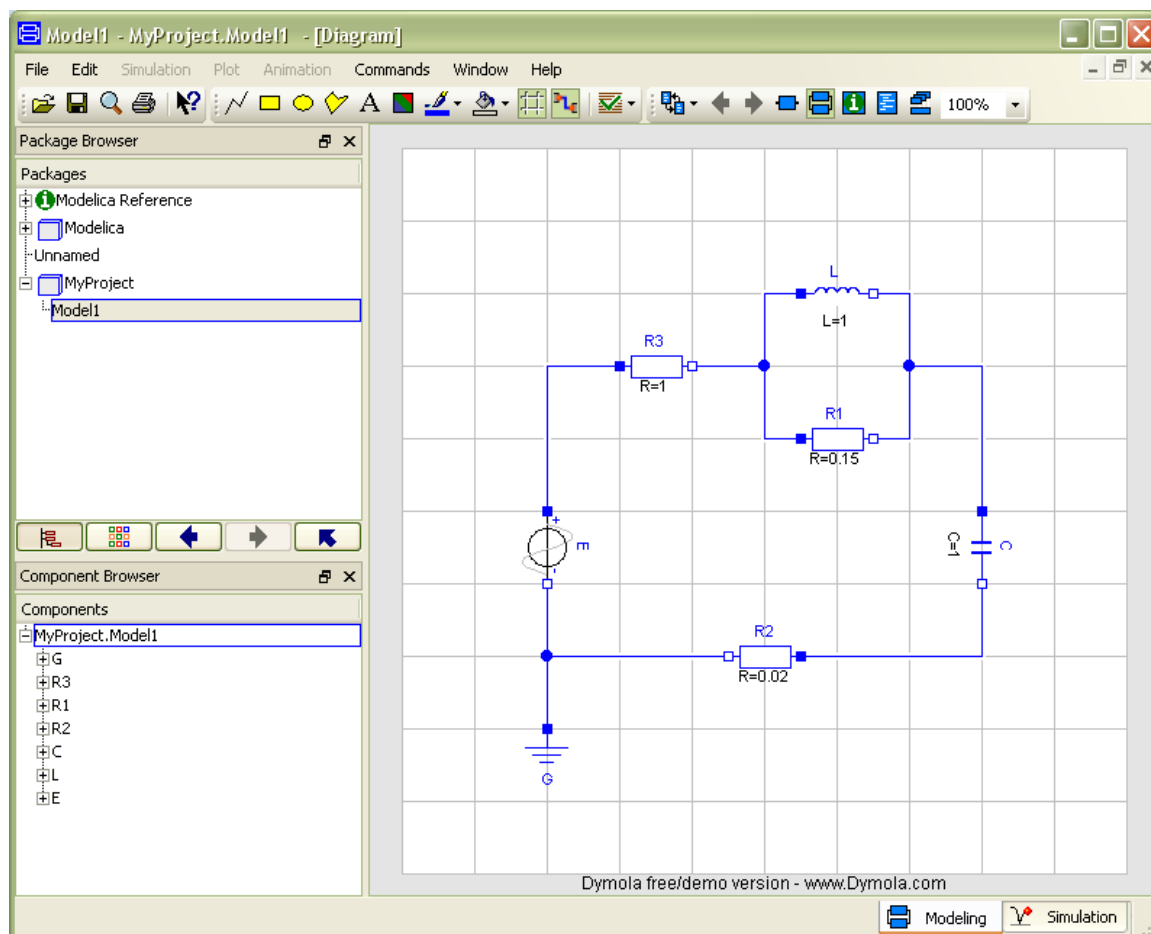


Рис. 6.2.14. Готовая визуальная модель электрической цепи.

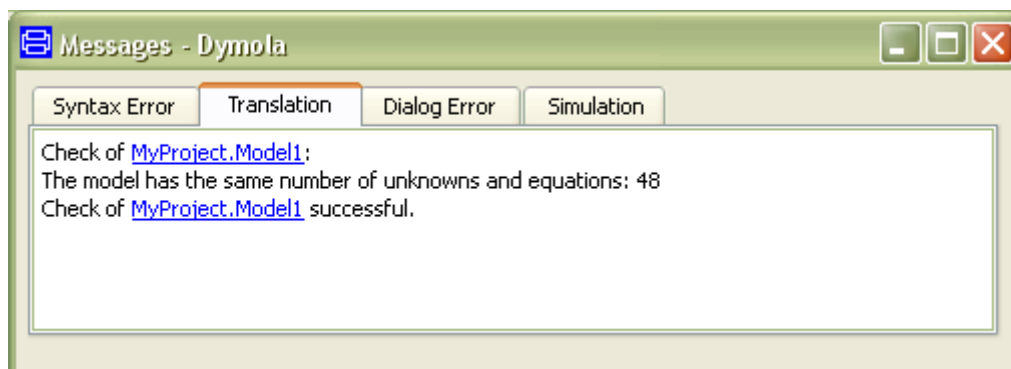




Рис. 6.2.15. Диагностическое сообщение компилятора.

Для синтаксического (и частично семантического) контроля правильности построенной модели можно воспользоваться инструментом со значком , результатом работы которого является диагностическое сообщение вида представленного на Рис. 6.2.15.

Для проведения испытаний построенной модели следует перейти в режим её прогона (закладка “Simulation”). После смены режима работы Dymola откроется окно интерфейса симуляции модели, представленное на Рис. 6.1.2. Опишем кратко структуру этого интерфейса. Слева, как обычно, расположено подокно браузера. На этот раз это браузер (всех) переменных модели. Эти переменные могут быть объявлены как в корневой модели, так и внутри её объектов. Они могут быть отдельными скалярами или быть элементами массивов.



Правее окна (на самом деле, как и в режиме подготовки, все подокна могут свободно перемещаться в пределах области основного окна компилятора) браузера переменных расположена область для окон трехмерной анимации и (двумерных) графиков. Количество окон графиков не ограничивается. Каждое новое окно может быть открыто либо при помощи команды меню “Plot → New Plot Window”, либо при помощи кнопки  на панели инструментов. Окно анимации, если оно по какой-либо причине оказалось закрытым, всегда может быть открыто при помощи команды меню “Animation → New Animation Window”.

В нижней части интерфейса расположено окно оболочки командной строки Dymola. Компилятор имеет достаточно развитый командный язык, допускающий выполнение в интерактивном режиме одиночной команды, либо целой последовательности команд (скрипта), предварительно записанной в файл с расширением .mos. Возможность создания скриптов обеспечивает эффективный инструмент для автоматизации различных операций с компилятором. Верхняя часть окна оболочки командной строки является многострочным протоколом исполнения команд, а нижняя её часть является собственно командной строкой для ввода команд или имен скриптов. Простой пример работы командного процессора (сложение двух чисел: 2 и 2) показан на Рис. 6.2.16.



Рис. 6.2.16. Окно оболочки командного процессора.

В фазе прогона подоперации трансляции и симуляции модели можно выполнить вместе или раздельно. В первом случае следует

сразу нажать икону инструмента  (или выполнить команду меню “Simulation → Simulate”), что в данном случае будет означать запуск операции трансляции (если исходный код модели был изменен, а трансляция еще не проводилась). Во втором случае вначале при помощи инструмента с кнопкой , или команды меню “Simulation → Translate” проводится только трансляция. При этом вначале работает аналитический препроцессор, выполняющий оптимизацию системы ДАУ (в том числе редукцию индекса). Затем генерируется код модели на языке C (в данном случае работает Visual Studio 2008/Visual C++ 2008 Express Edition), и, наконец, генерируется исполнимый модуль модели dymosim.exe. В результате генерируется диагностическое сообщение вида изображенного на Рис. 6.2.17.

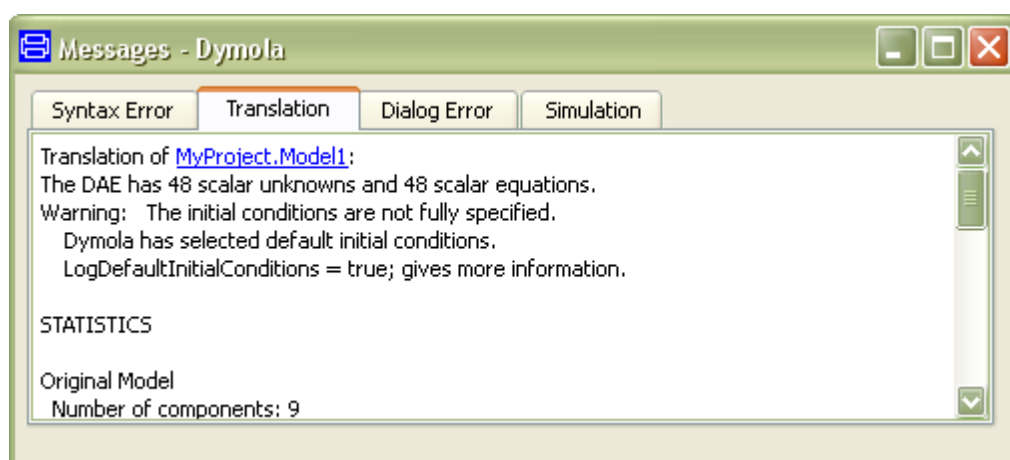



Рис. 6.2.17. Диагностическое сообщение фазы трансляции модели.

Затем, после первой трансляции модели, удобно провести настройку аппарата симуляции. Для этого используется инструмент настройки, запускаемый при помощи кнопки  (или команды меню “Simulation → Setup...”) и изображенный на Рис. 6.2.18. Закладка “General” позволяет задать общие настройки такие, как: а) моменты начала (обычно это нуль) и конца симуляции в масштабе времени модели; б) густоту точек вывода результатов на графики в зависимости от времени (можно задать в виде длины интервала — “Interval length” — между соседними моментами, либо в виде количества интервалов — “Number of intervals”); в) свойства процесса интегрирования (метод интегрирования — “Algorithm” и используемая при этом точность — “Tolerance”). Для решения рассматриваемого примера выберем количество интервалов, равным 5000, и точность —  $10^{-6}$ .

Поясним кратко смысл остальных закладок. “Translation” задает различные режимы трансляции модели. “Output” определяет категории переменных, информация о которых поступает в выходной файл симуляции модели. Результирующая информация численного экспе-

римента помещается в файл с расширением `.mat`. Размер этого файла может быть очень большим, что может, в свою очередь, существенно замедлить процесс симуляции. Отменяя вывод категорий переменных, не представляющих интереса для экспериментатора, можно значительно ускорить процесс вычислений. “Debug” включает/отключает вывод различной диагностической информации, полезной при отладке модели. “Compiler” обеспечивает подключение нужного компилятора с языка C для генерации исполнимого кода модели. Закладка “Realtime” предназначена для настройки режима привязки времени модели к реальному времени. Это средство, полезное при отладке модели во взаимодействии с реальными процессами, поставляется за отдельную плату.

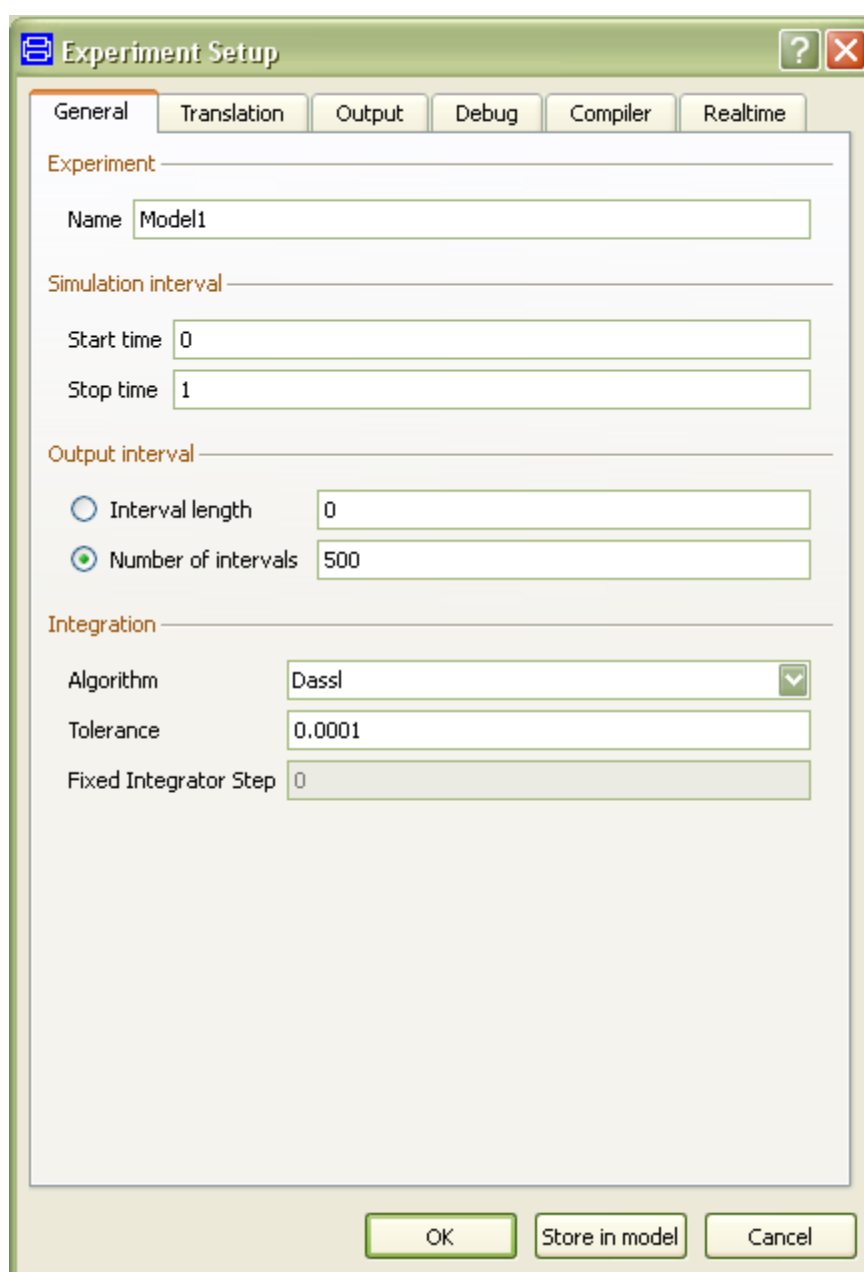


Рис. 6.2.18. Настройка вычислительного процесса.

После настройки запускается симуляция модели. Для примера построим график зависимости от времени тока, текущего по индуктору. Результат представлен на Рис. 6.2.19.

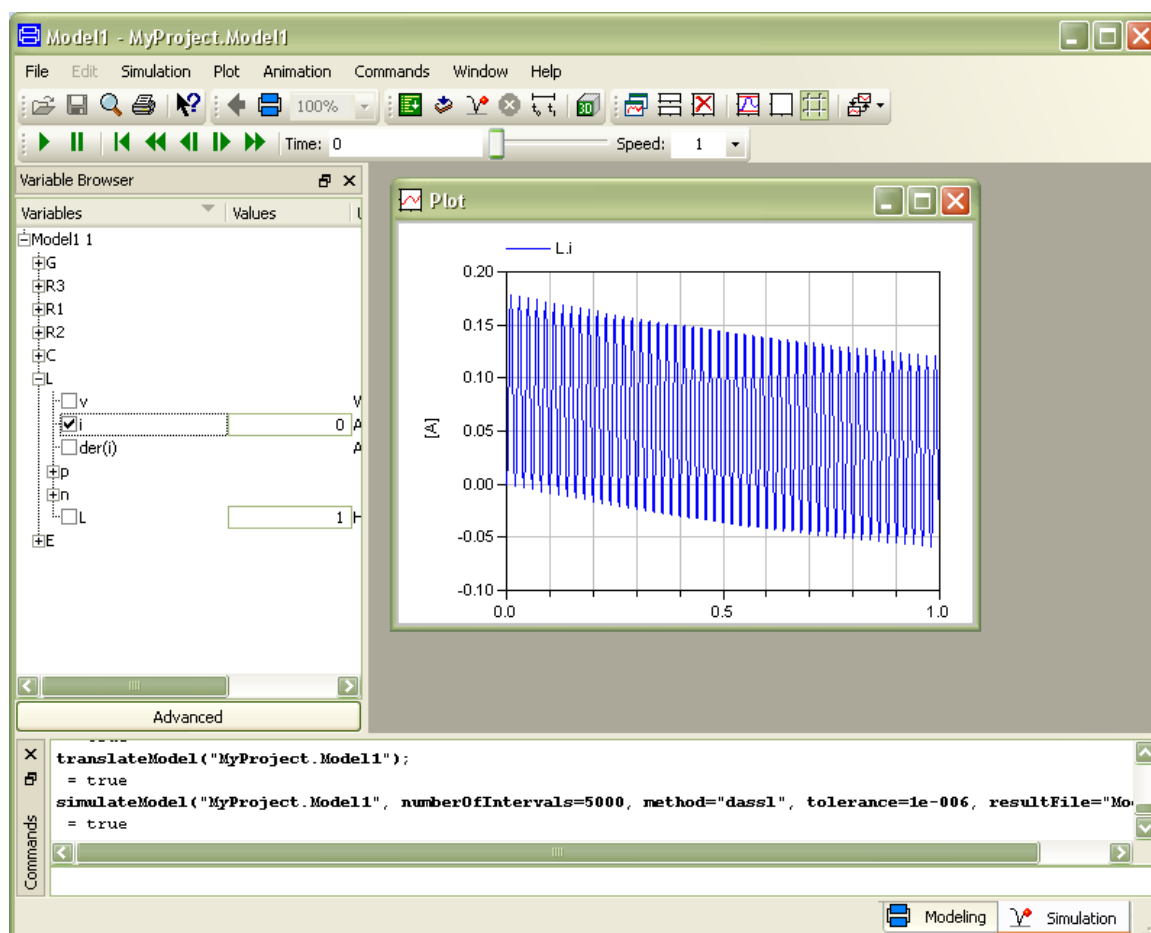


Рис. 6.2.19. Результат тестового прогона модели.

## 6.3. Отладка проекта

При создании модели исключительно важное значение имеет возможность независимой проверки правильности её работы. Для этого обычно лучше всего иметь какую-нибудь другую модель, уже прошедшую проверку. Очень часто в качестве эталона используется какая-либо аналитическая модель, поддающаяся проверке при помощи визуального контроля. Процесс верификации состоит в одновременном прогоне проверяемой и эталонной моделей и последующем сравнении некоторых контрольных переменных, имеющих в сравниваемых моделях одинаковый смысл.

В случае рассмотренной в предыдущем разделе задачи в качестве эталонной модели используем аналитическую модель электрической цепи. Для этого введем следующие переменные задачи:

- $v$  — падение напряжения во всей цепи, снимаемое на контактах источника питания  $E$ ;
- $v_1$  — падение напряжения на контактах резистора  $R_3$ ;
- $v_2$  — падение напряжения на контактах резистора  $R_1$ ;
- $v_3$  — падение напряжения на контактах индуктора  $L$ ;
- $v_4$  — падение напряжения на контактах конденсатора  $C$ ;
- $v_5$  — падение напряжения на контактах резистора  $R_2$ ;
- $i$  — ток, протекающий по источнику питания  $E$ ;
- $i_1$  — ток, протекающий по резистору  $R_3$ ;
- $i_2$  — ток, протекающий по резистору  $R_1$ ;
- $i_3$  — ток, протекающий по индуктору  $L$ ;
- $i_4$  — ток, протекающий по конденсатору  $C$ ;
- $i_5$  — ток, протекающий по резистору  $R_2$ .

На визуальной модели Рис. 6.2.14 «закрашенные» порты объектов соответствуют контактам положительной полярности, а незакрашенные порты соответствуют контактам отрицательной полярности. Предполагается, что электрический ток «течет» от положительно заряженных контактов к отрицательно заряженным.

Всего получилось двенадцать переменных задачи. Наша задача: построить двенадцать (независимых) ДАУ. Прежде всего запишем уравнения Кирхгофа для напряжений. Имеем

$$\begin{aligned} v_2 &= v_3, \\ v &= v_1 + v_2 + v_4 + v_5. \end{aligned}$$

Уравнения Кирхгофа для токов могут быть, очевидно, записаны в виде

$$\begin{aligned} i &= i_1, \\ i_1 &= i_2 + i_3, \\ i_2 + i_3 &= i_4, \\ i_4 &= i_5. \end{aligned}$$

Таким образом, законы Кирхгофа доставляют нам шесть линейных алгебраических уравнений. Далее, три сопротивления в соответствии с законом Ома доставляют нам три уравнения для резистивных ( $R$ -элементов в терминологии теории бондграфов [24]) элементов

$$\begin{aligned} v_1 &= R_3 i_1, \\ v_2 &= R_1 i_2, \\ v_5 &= R_2 i_5. \end{aligned}$$

Кроме этого, в цепи имеется элемент инертности ( $I$ -элемент), реализуемый при помощи индуктора, и описываемый дифференциальным уравнением



$$L \frac{di_3}{dt} = v_3.$$

Имеется также элемент податливости (С-элемент, соответствующий линейной пружине), реализуемый при помощи конденсатора, и задаваемый также дифференциальным уравнением вида

$$C \frac{dv_4}{dt} = i_4.$$

Получившуюся систему уравнений следует замкнуть при помощи уравнения источника усилия (SE-элемента), соответствующего источнику напряжения

$$v = E(t),$$

где заданная функция  $E(t)$  в нашем случае является синусоидальной.

В общей сложности получилось двенадцать ДАУ. Для их реализации перейдем к базовому слою описания класса `Model1` — слою языка `Modelica`. После этого в главном окне компилятора получим смешанное текстово-графическое изображение, показанное на Рис. 6.3.1. В таком виде разработчик не может случайно повредить сгенерированные компилятором декларации объектов класса (описаны в секции деклараций перед ключевым словом **equation**) и уравнения модели (описаны в секции поведения после ключевого слова **equation**). Скрытый значками текст на `Modelica` можно развернуть с различной степенью детализации при помощи контекстного меню, предварительно поместив курсор мыши в область основного окна.

```
model Model1
+
+
equation
+
end Model1;
```

Рис. 6.3.1. Сгенерированные компилятором компоненты свернуты.

Реализуем в рамках того же (для простоты) класса `Model1` описанную выше аналитическую модель электрической цепи. Для этого перед ключевым словом **equation** добавляем в модель декларации всех переменных, участвующих в уравнениях аналитической модели. Сами же уравнения, составленные в соответствии с правилами языка `Modelica`, вставляются в текст модели в секцию поведения после ключевого слова **equation**. Результат будет иметь вид, представленный на Рис. 6.3.2.

```

model Modell1
  SI.Voltage v;
  SI.Voltage v1;
  SI.Voltage v2;
  SI.Voltage v3;
  SI.Voltage v4;
  SI.Voltage v5;
  SI.Current i;
  SI.Current i1;
  SI.Current i2;
  SI.Current i3;
  SI.Current i4;
  SI.Current i5;
equation
  v2 = v3;
  v = v1 + v2 + v4 + v5;
  i = i1;
  i1 = i2 + i3;
  i2 + i3 = i4;
  i4 = i5;
  v1 = R3.R*i1;
  v2 = R1.R*i2;
  v5 = R2.R*i5;
  L.L*der(i3) = v3;
  C.C*der(v4) = i4;
  v = E.v;
end Modell1;

```

Рис. 6.3.2. Текст модифицированной модели.

Заметим здесь, что для нормальной трансляции модели следует не забыть добавить в охватывающий модель контейнер директиву импорта имени SI так, как это уже было описано выше. Параметры объектов модели такие, как сопротивления, индуктивность, емкость, а также переменная генерируемого в источнике питания напряжения имеют обычные двухступенчатые имена, принятые для структурированных объектов: R3.R, R1.R, R2.R, L.L, C.C, E.v.

Выполняя так, как это было описано выше, трансляцию модели в режиме прогона, настройку вычислений и симуляцию, выведем в окно графика для определенности, в виде функций времени модели, величину тока, протекающего по катушке индуктора  $L$  из объекта визуальной модели  $L.i$  (синий график) и переменной  $i3$  (красный график) модели аналитической. Получится результат, представленный на Рис. 6.3.3. Как видим, красная кривая, представляющая результат работы аналитической модели полностью «накрывает» синюю, пред-

ставляющую визуальную модель. Это свидетельствует об идентичности обеих моделей, что можно проверить на всех остальных переменных, и (косвенно) обеспечивает верификацию каждой из них.

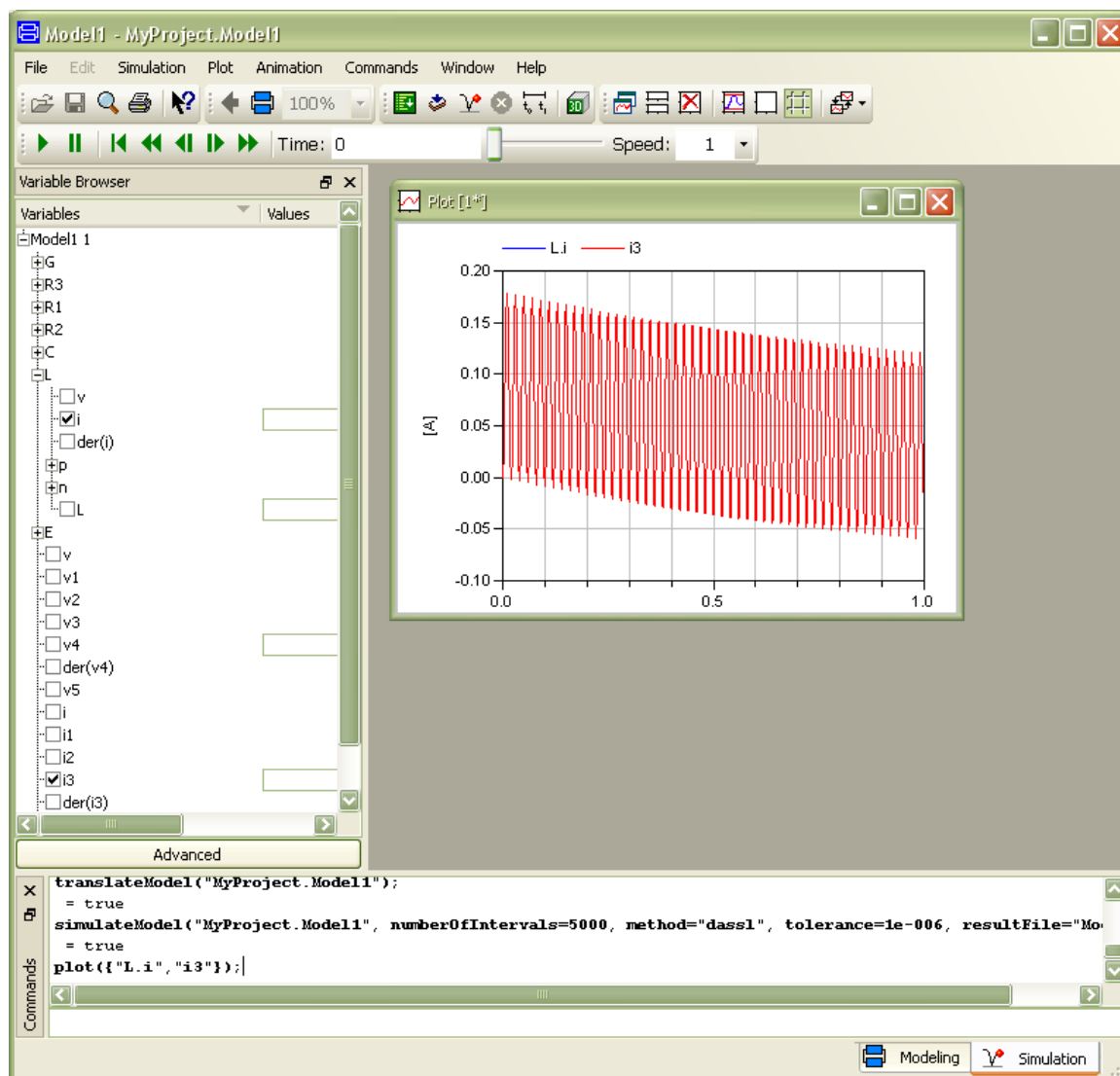


Рис. 6.3.3. Результат верификации модели цепи.

## 6.4. Развитие проекта

Теперь, после верификации разработанной и реализованной модели простейшей электрической цепи, попытаемся усложнить модель — модифицируем поведение одного из резисторов:  $R_3$ . Для этого в модели Resistor стандартной библиотеки Modelica.Electrical.Analog.Basic, помимо уже реализованной линейной зависимости сопротивления от температуры проводника резистора, введем нелинейную зависимость базового (при базовой температуре) сопротивления от тока в виде функции

$$R(i) = \begin{cases} b|i|^3 & \text{при } |i| \leq i_{\max} \\ bi_{\max}^3 & \text{при } |i| > i_{\max} \end{cases},$$

где  $i_{\max} > 0$  является предельным значением силы тока,  $b$  — коэффициент пропорциональности.

Используем в качестве исходного материала для создания новой модели нелинейного резистора стандартный класс `Resistor`. Для этого вначале скопируем класс в свою личную библиотеку. Копирование легко осуществить при помощи команды “Edit → Duplicate Class” контекстного меню копируемого класса (окно диалога показано на Рис. 6.4.1). В этом диалоге в поле “Insert in package:” нужно указать имя библиотеки-приемника копии класса. В нашем случае это `MyProject` (выпадает из списка при помощи соответствующего органа управления справа от поля).

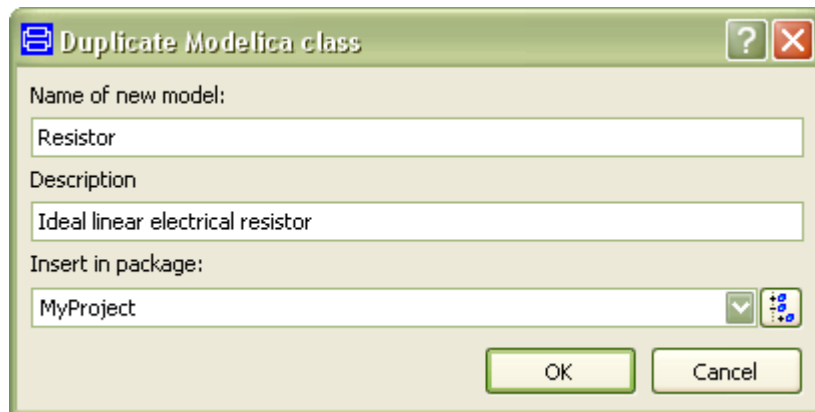


Рис. 6.4.1. Диалог копирования класса.

После копирования кода модели `Resistor` в личную библиотеку его можно модифицировать по своему усмотрению. Выполним необходимые изменения на уровне текста Modelica. Получим код модели `MyResistor`, представленный на Рис. 6.4.2.

```
model MyResistor "Ideal linear electrical resistor"
  SI.Resistance R "Resistance R_ref at temperature T_ref";
  parameter SI.Current imax = 15 "Current of saturation";
  parameter Real b = 0.001;
  parameter SI.Temperature T_ref=300.15 "Reference temperature";
  parameter SI.LinearTemperatureCoefficient alpha=0;
  extends Modelica.Electrical.Analog.Interfaces.OnePort;
  extends Modelica.Electrical.Analog.Interfaces.ConditionalHeatPort(T = T_ref);
  Modelica.SIunits.Resistance R_actual
    "Resistance = R_ref*(1 + alpha*(intenalHeatPort.T - T_ref))";
a;
equation
  assert((1 + alpha*(T_heatPort - T_ref)) >= Modelica.Constants.eps,
    "Temperature outside scope of model!");
  R_actual = R*(1 + alpha*(T_heatPort - T_ref));
  v = R_actual*i;
  LossPower = v*i;
  R = if abs(i) <= imax then b*abs(i)^3 else b*imax^3;
end MyResistor;
```

Рис. 6.4.2. Код модифицированного класса `Resistor`.

Затем, для реализации модифицированной модели цепи, создается копия Model2 класса Model1. В созданной копии на уровне визуального представления уничтожается объект R3 (и автоматически все его соединения). Затем, также как и выше, методом перетаскивания, но уже из личной библиотеки, создается объект R3 класса MyResistor со всеми необходимыми соединениями. Результирующая визуальная модель показана на Рис. 6.4.3.

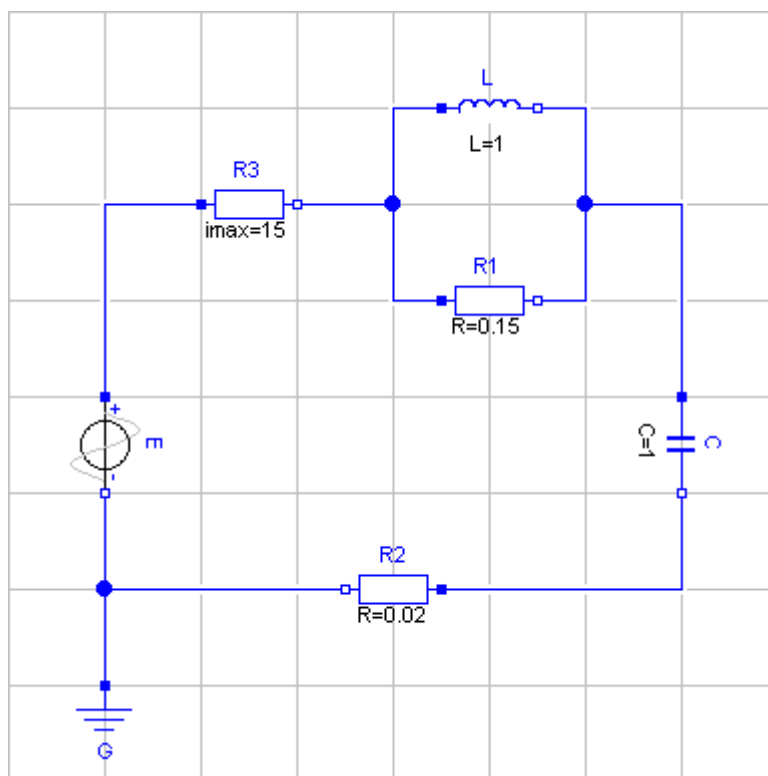


Рис. 6.4.3. Визуальное представление модифицированной модели.

Один из результатов симуляции показан на Рис. 6.4.4, где представлена, полученная при помощи численного эксперимента функция  $R_3(i_3)$ . График на Рис. 6.4.4 получен при помощи еще одной возможности Dymola — экспорта графической информации, обеспечиваемой командой меню “File → Export... → Image...” и позволяющей готовить отчетную документацию о проводимых численных экспериментах. Легко видеть, что график соответствует заданной выше функции  $R(i)$ . Для установления зависимости переменной  $R_3.R$  не от времени, а от переменной  $R_3.i$ , нужно (см. Рис. 6.4.5), оставляя окно требуемого графика в фокусе, установить курсор мыши на переменную аргумента ( $R_3.i$ ) в браузере переменных и выполнить команду “Independent variable →  $R_3.i$ ” контекстного меню. После этого по оси абсцисс данного графика будет откладываться переменная  $R_3.i$ . Затем следует обычным образом пометить в том же браузере переменную  $R_3.R$ , и она будет откладываться по оси ординат в том же окне.

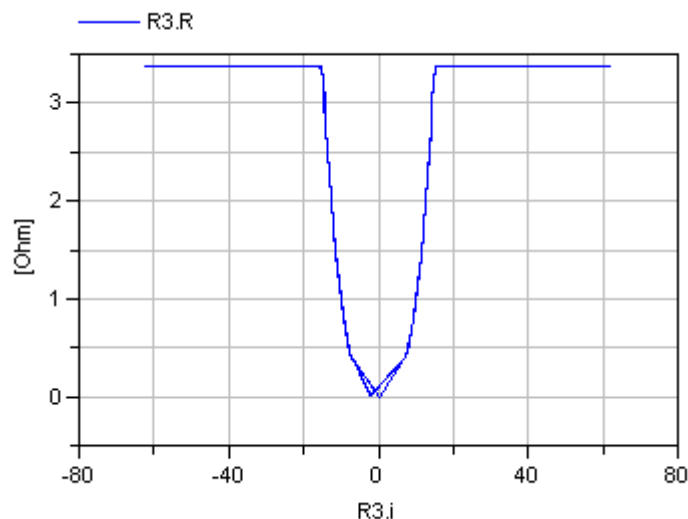


Рис. 6.4.4. График зависимости сопротивления от тока.

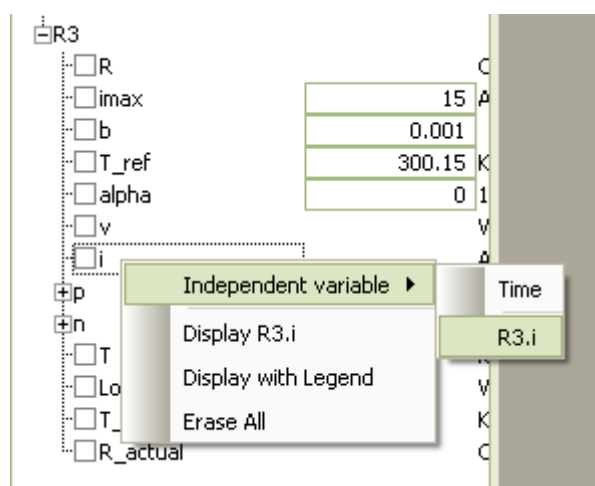


Рис. 6.4.5. Выбор новой независимой переменной.

В данном примере вне внимания остался вопрос формирования начальных условий для переменных состояния. Если начальные условия не указывать, то Dymola всегда предполагает, что они нулевые. В контексте рассмотренной задачи начальные значения фазовых переменных не имели существенного значения. Эта возможность будет более подробно в следующем разделе.

## 6.5. Пример маятника

Ранее, при описании трехмерной анимации, был рассмотрен пример физического маятника. С динамической точки зрения при реализации на языке Modelica задача была предварительно редуцирована к лагранжевой форме и реализована в виде системы обыкновенных дифференциальных уравнений (ОДУ) для голономной механической системы с одной степенью свободы. При помощи языка

Modelica и компилятора Dymola возможен и другой подход к описанию компьютерной модели механической системы. Процесс аналитической редукции можно оставить символьному препроцессору компилятора. Динамику в этом случае можно описывать при помощи системы ОДУ Ньютона (для движения центра масс) и Эйлера (для вращательного движения вокруг центра масс), дополненных нужным набором уравнений связи. Очень часто компилятор способен «переварить» (провести редукцию индекса) получающуюся систему ДАУ, особенно если соблюдать определенные правила при разработке моделей динамики систем твердых тел [21].

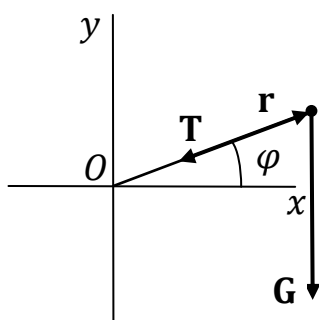


Рис. 6.5.1. Математический маятник.

Для простоты рассмотрим пример математического маятника (Рис. 6.5.1), представляющего собой грузик (материальную точку  $M$ ) массы  $m$ , подвешенный на нерастяжимой нити длины  $L$ , в поле сил тяжести. Опишем движение материальной точки грузика в вертикальной плоскости относительно абсолютной системы координат  $Oxy$ , где точка  $O$  является точкой подвеса, ось  $Ox$  проходит горизонтально, а ось  $Oy$  — вертикально.

На точку  $M$  действуют силы: тяжести  $\mathbf{G} = (0, -mg)^T$  и натяжения нити  $\mathbf{T} = (T_x, T_y)^T$ . Поэтому дифференциальные уравнения Ньютона в векторной форме имеют вид

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v}, \\ m\dot{\mathbf{v}} &= \mathbf{G} + \mathbf{T},\end{aligned}\tag{6.5.1}$$

Помимо дифференциальных в рассматриваемой задаче имеются также алгебраические уравнения связи. Начнем с уравнения кинематической связи

$$x^2 + y^2 - L^2 = 0,$$

которое возникает из-за удержания материальной точки на окружности радиуса  $L$ . Это уравнение можно записать также в виде

$$(\mathbf{r}, \mathbf{r}) - L^2 = 0.$$

Однако, для облегчения работы компилятора при редукции системы ДАУ это уравнение предпочтительно записать в дифференциальной форме

$$x\dot{x} + y\dot{y} = 0,$$

или

$$(\mathbf{r}, \mathbf{v}) = 0. \quad (6.5.2)$$

Кроме кинематической связи имеется также уравнение идеальности связи, выражающее ортогональность силы натяжения нити и текущей скорости точки

$$(\mathbf{T}, \mathbf{v}) = 0. \quad (6.5.3)$$

Вместо скалярного уравнения (6.5.3) можно было бы использовать векторное, эквивалентное двум скалярным уравнениям и характеризующее коллинеарность радиуса-вектора точки и силы натяжения нити,

$$\mathbf{T} = \mu \mathbf{r},$$

где  $\mu$  — дополнительная скалярная переменная.

```
model MathPendulum
  parameter SI.Mass m = 1;
  parameter SI.Length L = 1;
  parameter SI.Acceleration g = 9.81;
  parameter SI.Force[2] G = {0, -m*g};
  parameter SI.Angle phi = -1 "Initial polar angle";
  parameter SI.Velocity v0 = 0.01 "Initial linear velocity";
  SI.Position[2] r(start = {L*cos(phi), L*sin(phi)});
  SI.Velocity[2] v(start = {-L*sin(phi), L*cos(phi)});
  SI.Force[2] T "Tension";
equation
  der(r) = v;
  m*der(v) = G + T;
  r*v = 0;
  T*v = 0;
end MathPendulum;
```

Рис. 6.5.2. Компьютерная модель математического маятника.

Здесь специальное внимание уделено выбору начальных данных задачи. Как и ранее, эта возможность обеспечивается при помощи конструкции `start` для переменных положения `r` и скорости `v`. Формулы для начальных данных выбираются так, чтобы не нарушалась наложенная на механическую систему связь. Более того, при использовании прямоугольных координат нельзя использовать нулевые начальные данные для координат и скоростей, поскольку, как



легко видеть из кода модели на Рис. 6.5.2, в этом случае вырождаются два последних уравнения секции поведения. Поэтому солвер (решатель неявных систем уравнений) компилятора не может запустить процесс интеграции. Однако, если параметр  $v_0$  отличен от нуля, симуляция успешно запускается, и в процессе работы модели она успешно «проходит» моменты упомянутого вырождения.

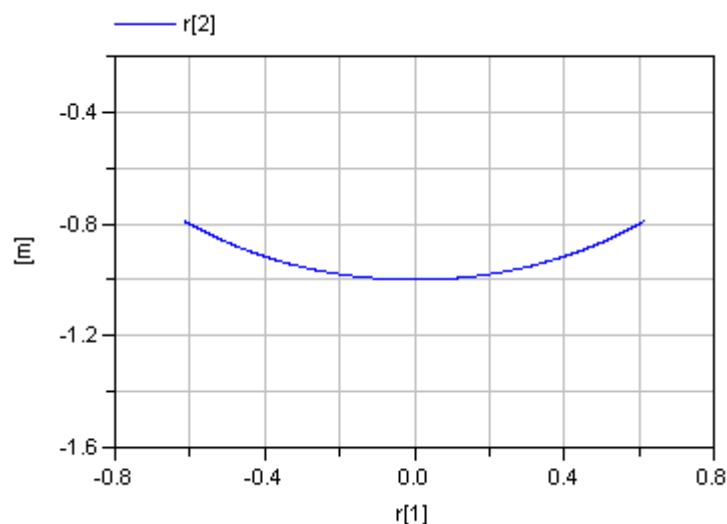


Рис. 6.5.3. Траектория материальной точки.

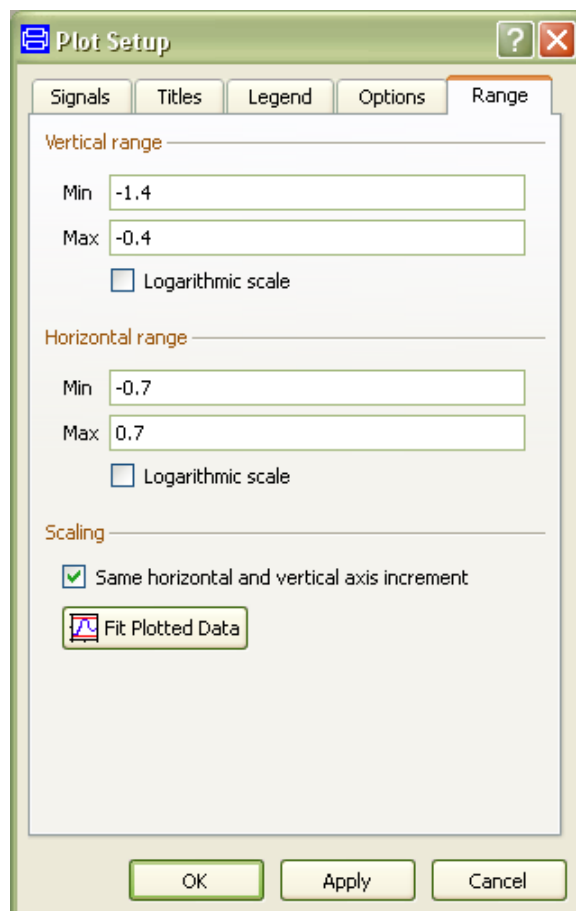


Рис. 2.5.4. Окно настройки параметров графика.

Рассмотренная здесь модель математического маятника в прямоугольных координатах успешно проходит все фазы подготовки и прогона. На Рис. 6.5.3 показана траектория точки в вертикальной плоскости  $Oxy$ . Легко видеть, что это сегмент окружности радиуса  $L = 1$ . Согласование масштабов переменных, откладываемых по осям абсцисс и ординат, обеспечивается при помощи команды настройки графика “Plot → Setup...”, открывающей окно диалога настройки (Рис. 2.5.4). В этом окне на закладке “Range” нужно пометить блок проверки “Same horizontal and vertical axis increment” и нажать кнопку “Apply”, после чего требуемое одинаковое масштабирование будет установлено.

Помимо закладки “Range” имеются другие настройки, позволяющие выбирать цвет и стиль изображения кривых, определять оформление пояснений и задавать дополнительные действия при прорисовке изображения. По умолчанию Dymola автоматически масштабирует выводимые в окно графики в соответствии с текущими размерами этого окна.

# Библиография

1. Анурьев В. И. Справочник конструктора-машиностроителя: В 3-х т.: Т. 1. — М.: Машиностроение, 2001. 920 с.
2. Анурьев В. И. Справочник конструктора-машиностроителя: В 3-х т.: Т. 2. — М.: Машиностроение, 2002. 912 с.
3. Булгаков Б. В. Колебания. — М.: ГИТТЛ, 1954. 892 с.
4. Гузенков П. Г. Детали машин. — М.: Высш. шк., 1986. 359 с.
5. Диментберг Ф. М. Винтовое исчисление и его приложения в механике. — М.: Наука, 1965. 200 с.
6. Дунаев П. Ф., Леликов О. П. Конструирование узлов и деталей машин: Учеб. пособие для техн. спец. вузов. — М.: Высш. шк., 1998. 447 с.
7. Иванов М. Н. Детали машин. — М.: Высш. шк., 1998. 383 с.
8. Киркач Н. Ф., Баласанян Р. А. Расчет и проектирование деталей машин: Учеб. пособие для техн. вузов. — Х.: Основа, 1991. 276 с.
9. Колесов Ю. Б., Сениченков Ю. Б. Моделирование систем. Практикум по компьютерному моделированию. — СПб.: БХВ-Петербург, 2007. 352 с.
10. Колпаков А. П., Карнаухов И. Е. Проектирование и расчет механических передач. — М.: Колос, 2000, 328 с.
11. Конструкции и расчет зубчатых редукторов. Справочное пособие. Кудрявцев В. Н., Державец Ю. А., Глухарев Е. Г. — Л.: «Машиностроение», 1971, 328 с.
12. Мархель И. И. Детали машин: Учебник. — М.: ФОРУМ: ИНФРА-М, 2005. 336 с.
13. Новожилов И. В. Фракционный анализ. — М.: Изд-во механико-математического факультета МГУ, 1995. 224 с.
14. Технология конструкционных материалов. Учебник для вузов. — М.: «Машиностроение», 1977. 664 с.
15. Филиппов А. Ф. Дифференциальные уравнения с разрывной правой частью. — М.: Наука 1985. 224 с.
16. Энциклопедический справочник «Машиностроение». Раздел первый. Инженерные расчеты в машиностроении, т. 2. М.: Государственное научно-техническое издательство машиностроительной литературы, 1948, 891 с.
17. Dymola. Dynamic Modeling Laboratory. User Manual. Volume 1. Dymola 7.3 — Lund: Dynasim AB, Research Park Ideon, 2009. 485 pp.
18. Dymola. Dynamic Modeling Laboratory. User Manual. Volume 2. Dymola 7.3 — Lund: Dynasim AB, Research Park Ideon, 2009. 356 pp.
19. Fritzson P. Principles of object-oriented modeling and simulation with Modelica 2.1. — Piscataway, NJ: IEEE Press, 2004. 898 pp.

20. <http://www.modelica.org>
21. Kosenko I. I. Physically oriented approach to construct multibody system dynamics models using Modelica language // Proceedings of Multibody 2007, Multibody Dynamics 2007. An ECCOMAS Thematic Conference, Politecnico di Milano, Milano, Italy, June 25–28, 2007, 20 pp.
22. Modelica — A unified object-oriented language for physical systems modeling. Language Specification. Version 3.0. — Modelica Association, 2007. 193 pp.
23. Modelica — A unified object-oriented language for physical systems modeling. Tutorial. — Modelica Association, 2000. 52 pp.
24. Mukherjee A., Karmakar R. Modelling and simulation of engineering systems through bondgraphs. — Alpha Science International Ltd., 2000. 471 pp.
25. Trébaol G. Designing parametric spur gears with CATIA V5. [http://gtrebaol.free.fr/doc/catia/spur\\_gear.html](http://gtrebaol.free.fr/doc/catia/spur_gear.html)

Научное издание

*Иван Иванович Косенко*

*Лариса Викторовна Кузнецова*

*Анатолий Викторович Николаев*

*Леонид Юрьевич Кузнецов*

МЕТОДЫ ГЕОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ И ВИРТУАЛЬНОГО  
ПРОТОТИПИРОВАНИЯ. ПЕРСПЕКТИВНЫЕ ТЕХНОЛОГИИ

*Монография*

Печатается в авторской редакции