

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Лекционный материал по компьютерному
геометрическому моделированию

Содержание

Введение	4
Лекция 1. Жизненный цикл продукта. Роль геометрического моделирования.....	6
Лекция 2. Объёмное моделирование твёрдого тела. Способы моделирования.....	13
Лекция 3. Функции моделирования. Моделирование кривых линий и поверхностей.....	26
Лекция 4. Классификация поверхностей. Развёртки.....	46
Лекция 5. Ядра геометрического моделирования.....	68
Лекция 6. Параметрическое моделирование.....	79
Лекция 7. Прямое моделирование.....	88
Лекция 8. Методы проецирования. Техническое рисование.....	99
Лекция 9. Основы графического программирования.....	115
Лекция 10. Возможности CAD модуля системы ADEM.....	133
Список литературы.....	158

Введение

Для обеспечения конкурентоспособности создаваемых изделий путем снижения сроков проектирования и изготовления, уменьшения себестоимости и значительного повышения качества конструкторско-технологическая подготовка производства в настоящее время предполагает использование 3D моделей изделий, их узлов и деталей. Такой подход обеспечивает также возможность оценки работоспособности и технического обслуживания разрабатываемых конструкций в условиях эксплуатации, создания в автоматизированном режиме необходимой технической документации для разработки путей и способов их изготовления и контроля с применением новых технологий, оборудования и инструмента.

Переход на безбумажные, с использованием компьютера технологии, когда информация создается, хранится и используется в едином информационном пространстве в электронном виде, требует выполнения новых условий при подготовке специалистов, понимания ими принципов функционирования прикладных программ, их классификации и возможностей использования в конкретных целях, устойчивых навыков работы в диалоговом режиме. При этом безбумажная графическая подготовка является основой сквозной подготовки современного специалиста.

Для обеспечения подготовки специалистов для инновационного машиностроения в учебном процессе института двигателей и энергетических установок (ИДЭУ) Самарского университета принята концепция виртуального предприятия, которая предполагает приобретение знаний и навыков студентами путем имитационного моделирования, в частности, сквозного конструкторско-технологического проектирования в среде единого информационного пространства (ЕИП) института.

Геометрические модели используются для решения многих задач: визуализации, построения расчетных сеток, генерации управляющих программ ЧПУ и т.д. В первую очередь, они предназначены для хранения информации о форме объектов и их взаимном расположении и предоставления ее для обработки в удобном для компьютерной программы виде. В этом - ключевое отличие электронной геометрической модели от чертежа, который представляет собой условное символично-графическое изображение, предназначенное для чтения человеком.

Цели и задачи курса

Цели и задачи курса продиктованы основными **требованиями** современного специалиста теплоэнергетика к уровню геометро-модельной подготовки:

- создание геометрических моделей объекта производства, его деталей, технологических схем оборудования, средств технологического оснащения, формообразующего, обрабатывающего и мерительного инструмента на основе 3D моделирования, в том числе параметрического для типовых изделий;

- автоматизированный выпуск комплектов конструкторской и технологической документации по 3D моделям различных изделий на основе технологии баз данных в среде единого информационного пространства (ЕИП).

Лекция 1

Жизненный цикл продукта. Роль геометрического моделирования.

Современные предприятия не смогут выжить во всемирной конкуренции, если не будут выпускать новые продукты лучшего качества, более низкой стоимости и за меньшее время. Поэтому они стремятся использовать огромные возможности памяти компьютеров, их высокое быстродействие и возможности удобного графического интерфейса для того, чтобы автоматизировать и связать друг с другом задачи проектирования и производства, которые раньше были весьма утомительными и совершенно не связанными друг с другом. Таким образом сокращается время и стоимость разработки и выпуска продукта. Для этой цели используются технологии автоматизированного проектирования (computer-aided design - CAD), автоматизированного производства (computer-aided manufacturing - CAM) и автоматизированной разработки и конструирования (computer-aided engineering - CAE). Чтобы понять значение систем CAD/CAM/CAE (по-русски все эти системы вместе называются системами автоматизированного проектирования - САПР), мы должны изучить различные задачи и операции, которые приходится решать и выполнять в процессе разработки и производства продукта. Все эти задачи, взятые вместе, называются *жизненным циклом продукта (product cycle)*. Пример жизненного цикла продукта, приведен на рисунке 1.1.

Прямоугольники, нарисованные сплошными линиями, представляют два главных процесса, составляющих жизненный цикл продукта: процесс разработки и процесс производства. Процесс разработки начинается с запросов производства, которые обслуживаются отделом маркетинга, и заканчивается полным описанием продукта.

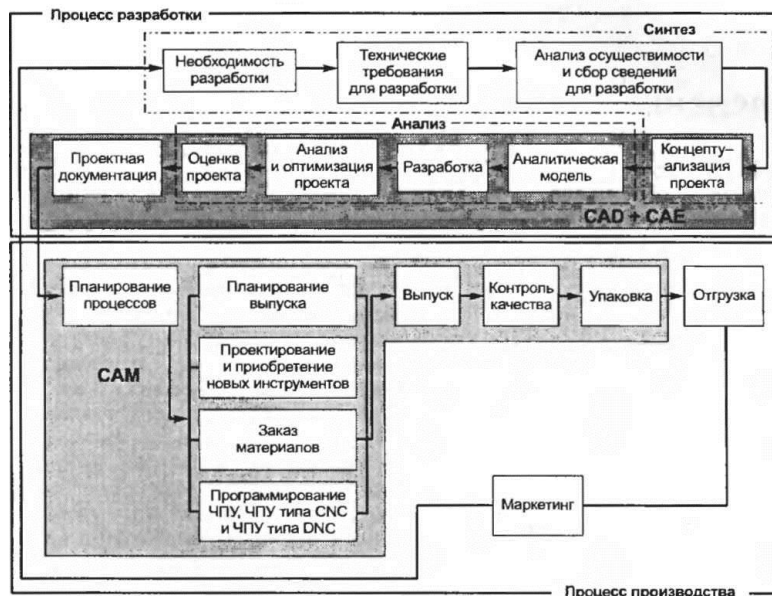


Рисунок 1.1 - Жизненный цикл продукта

Процесс производства начинается с технических требований и заканчивается поставкой готовых изделий заказчику.

Операции, относящиеся к процессу разработки, можно разделить на аналитические и синтетические. Как следует из рисунка 1.1, первичные операции разработки, такие как определение необходимости разработки, формулирование технических требований, анализ

осуществимости и сбор важной информации, а также концептуализация разработки, относятся к подпроцессу синтеза. Результатом подпроцесса синтеза является концептуальный проект предполагаемого продукта в форме эскиза или топологического чертежа, отражающего связи различных компонентов продукта. В этой части цикла делаются основные финансовые вложения, необходимые для реализации идеи продукта, а также определяется его функциональность. Большая часть информации, порождаемой и обрабатываемой в рамках подпроцесса синтеза, является качественной, а следовательно, неудобной для компьютерной обработки.

Готовый концептуальный проект анализируется и оптимизируется - это уже подпроцесс анализа. Прежде всего, вырабатывается аналитическая модель, поскольку анализируется именно модель, а не сам проект. Несмотря на быстрый рост количества и качества компьютеров, используемых в конструировании, в обозримом будущем отказаться от использования абстракции аналитической модели мы не сможем. Аналитическая модель получается, если из проекта удалить маловажные детали, редуцировать размерности и учесть имеющуюся симметрию. Редукция размерностей, например, подразумевает замену тонкого листа из какого-либо материала на эквивалентную плоскость с атрибутом толщины или длинного и тонкого участка на линию с определенными параметрами, характеризующими поперечное сечение. Симметричность геометрии тела и нагрузки, приложенной к нему, позволяет рассматривать в модели лишь часть этого тела.

После завершения проектирования и выбора оптимальных параметров начинается этап оценки проекта. Для этой цели могут изготавливаться прототипы. В конструировании прототипов все большую популярность приобретает новая технология, названная *быстрым прототипированием (rapid prototyping)*. Эта технология позволяет конструировать прототип снизу вверх, то есть непосредственно из проекта, поскольку фактически требует только лишь данных о поперечном сечении конструкции. Если оценка проекта на основании прототипа показывает, что проект не удовлетворяет требованиям, описанный выше процесс разработки повторяется снова.

Если же результат оценки проекта оказывается удовлетворительным, начинается подготовка проектной документации. К ней относятся чертежи, отчеты и списки материалов. Чертежи обычно копируются, а копии передаются на производство.

Как видно (см. рисунок 1.1), процесс производства начинается с планирования, которое выполняется на основании полученных на этапе проектирования чертежей, а заканчивается готовым продуктом. Технологическая подготовка производства - это этап, устанавливающий список технологических процессов по изготовлению продукта и задающий их параметры. Одновременно выбирается оборудование, на котором будут производиться технологические операции, такие как получение детали нужной формы из заготовки. В результате подготовки производства составляются план выпуска, списки материалов и управляющие программы (УП) для оборудования с ЧПУ. На этом же этапе обрабатываются прочие специфические требования, в частности рассматриваются конструкции средств технологического оснащения (зажимов, креплений и пр.). Подготовка занимает в процессе производства примерно такое же место, как подпроцесс синтеза в процессе проектирования, требуя значительного человеческого опыта и принятия качественных решений. Такая характеристика подразумевает сложность компьютеризации данного этапа. После завершения технологической подготовки

начинается выпуск готового продукта и его проверка на соответствие требованиям. Детали, успешно проходящие контроль качества, собираются вместе, проходят тестирование функциональности, упаковываются, маркируются и отгружаются заказчиком.

Описанный выше жизненный цикл продукта является типовым. Посмотрим теперь, каким образом на этапах этого цикла могут быть применены технологии CAD, CAM и CAE. Как уже говорилось, компьютеры не могут широко использоваться в подпроцессе синтеза, поскольку они не обладают способностью хорошо обрабатывать качественную информацию. Однако даже на этом этапе разработчик может, например, при помощи коммерческих баз данных успешно собирать важную для анализа осуществимости информацию, а также пользоваться данными из каталогов.

Полезными могут оказаться средства параметрического и геометрического моделирования, а также макропрограммы в *системах автоматизированной разработки чертежей (computer-aided drafting)*. Все это типичные примеры систем CAD.

Система геометрического моделирования (geometric modeling system) - это трехмерный эквивалент системы автоматизированной разработки чертежей, то есть программный пакет, работающий с трехмерными, а не с плоскими объектами.

В аналитической фазе проектирования ценность компьютеров проявляется по-настоящему. Программных пакетов для анализа напряжений, контроля столкновений и кинематического анализа существует столько, что приводить какие-либо названия смысла не имеет. Эти программные пакеты относятся к средствам автоматизированного конструирования (CAE). Главная проблема, связанная с их использованием, заключается в необходимости формирования аналитической модели. Проблемы не существовало бы вовсе, если бы аналитическая модель автоматически выводилась из концептуального проекта. Однако, как уже отмечалось, аналитическая модель не идентична концептуальному проекту - она выводится из него путем исключения несущественных деталей и редукции размерностей. Необходимый уровень абстракции зависит от типа анализа и желаемой точности решения. Следовательно, автоматизировать процесс абстрагирования достаточно сложно, поэтому аналитическую модель часто создают отдельно. Обычно абстрактная модель проекта создается в системе разработки рабочих чертежей или в системе геометрического моделирования, а иногда с помощью встроенных средств аналитического пакета. Аналитические пакеты обычно требуют, чтобы исследуемая структура была представлена в виде объединения связанных сеток, разделяющих объект на отдельные участки, удобные для компьютерной обработки. Если аналитический пакет может генерировать сетку автоматически, человеку остается задать только границы абстрактного объекта. В противном случае сетка также создается пользователем либо в интерактивном режиме, либо автоматически, но в другой программе. Процесс создания сетки называется *моделированием методом конечных элементов (finite-element modeling)*. Моделирование этим методом включает в себя также задание граничных условий и внешних нагрузок.

Подпроцесс анализа может выполняться в цикле оптимизации проекта по каким-либо параметрам. Разработано множество алгоритмов поиска оптимальных решений, а на их основе построены коммерчески доступные программы. Процедура оптимизации может считаться компонентом системы автоматизированного проектирования, но более естественно рассматривать эту процедуру отдельно.

Фаза оценки проекта также выигрывает от использования компьютера. Если для оценки проекта нужен прототип, мы можем быстро сконструировать его по заданному проекту при помощи программных пакетов, генерирующих код для машины быстрого прототипирования. Такие пакеты считаются программами для автоматизированной подготовки производства (САМ). Разумеется, форма прототипа должна быть определена заранее в наборе входных данных. Данные, определяющие форму, получаются в результате геометрического моделирования.

Быстрое прототипирование — удобный способ конструирования прототипа, однако еще удобнее пользоваться виртуальным прототипом, который часто называется «цифровой копией» (digital mock-up) и позволяет получить столь же полезные сведения.

Когда аналитические средства для работы с цифровыми копиями станут достаточно мощными, чтобы давать столь же точные результаты, что и эквивалентные эксперименты на реальных прототипах, цифровые копии начнут вытеснение обычных прототипов. Эта тенденция будет усиливаться по мере совершенствования технологий виртуальной реальности¹, позволяющих нам ощущать цифровую копию так же, как реальный прототип. Построение цифровой копии называется *виртуальным прототипированием*. Виртуальный прототип может быть создан и в специализированной программе геометрического моделирования.

Последняя фаза процесса разработки - подготовка проектной документации. На этом этапе чрезвычайно полезным оказывается использование систем подготовки рабочих чертежей. Способность подобных систем работать с файлами позволяет систематизировать хранение и обеспечить удобство поиска документов.

Компьютерные технологии используются и на стадии производства. Процесс производства включает в себя планирование выпуска, проектирование и приобретение новых инструментов, заказ материалов, программирование станков с ЧПУ, контроль качества и упаковку. Компьютерные системы, используемые в этих операциях, могут быть классифицированы как системы автоматизированного производства. Например, программа автоматизированной технологической подготовки (computer-aided process planning - CAPP) используется на этапе подготовки производства и относится к системам автоматизированного производства (САМ). Как отмечалось выше, подготовка производства с трудом поддается автоматизации, поэтому полностью автоматических систем технологической подготовки в настоящий момент не существует. Однако существует множество хороших программных пакетов, генерирующих код для станков с числовым программным управлением. Станки этого класса позволяют получить деталь нужной формы по данным, хранящимся в компьютере. Они аналогичны машинам для быстрого прототипирования.

К системам автоматизированного производства относят также программные пакеты, управляющие движением роботов при сборке компонентов и перемещении их между операциями, а также пакеты, позволяющие программировать координатно-измерительную машину (coordinate measuring machine - CMM), используемую для проверки геометрии продукта.

¹ Виртуальная реальность — это технология создания изображений, выглядящих как настоящие объекты. Эта технология позволяет пользователю ощущать цифровые объекты и манипулировать ими так же, как настоящими.

Итак, вы получили представление о том, каким образом компьютерные технологии используются в этапах, составляющих жизненный цикл продукта, и какие задачи решаются при помощи систем автоматизированного проектирования.

Согласно выше сказанному, *автоматизированное проектирование (computer-aided design - CAD)* представляет собой технологию, состоящую в использовании компьютерных систем для облегчения создания, изменения, анализа и оптимизации проектов. Таким образом, любая программа, работающая *с компьютерной графикой*, так же как и любое приложение, используемое в инженерных расчетах, относится к системам автоматизированного проектирования. Другими словами, множество средств CAD простирается от геометрических программ для работы с формами до специализированных приложений для анализа и оптимизации. Между этими крайностями умещаются программы для анализа допусков, расчета масс-инерционных свойств, моделирования методом конечных элементов и визуализации результатов анализа.

Самая основная функция CAD - определение *геометрии* конструкции, поскольку геометрия определяет все последующие этапы жизненного цикла продукта. Для этой цели обычно используются *системы геометрического моделирования* и разработки рабочих чертежей. Вот почему эти системы обычно и считаются системами автоматизированного проектирования. Более того, геометрия, определенная в этих системах, может использоваться в качестве основы для дальнейших операций в системах CAE и CAM. Это одно из *наиболее значительных* преимуществ CAD, позволяющее экономить время и сокращать количество ошибок, связанных с необходимостью определять геометрию конструкции с нуля каждый раз, когда она требуется в расчетах. Можно, следовательно, утверждать, что *системы геометрического моделирования* и системы автоматизированной разработки рабочих чертежей являются *наиболее важными* компонентами автоматизированного проектирования.

Особенности геометро-модельной подготовки студентов института двигателей и энергетических установок на кафедре инженерной графики Самарского университета.

Для оценки содержания учебной деятельности кафедры инженерной графики в структуре ИДЭУ на основе рассмотрения концептуальной модели учебной деятельности сформулированы требования к геометро-модельной подготовке для обеспечения сквозной подготовки специалистов в условиях работы в ЕИП института и пути их реализации.

Анализ концептуальной модели показал, что геометро-модельная подготовка должна, в первую очередь, решить следующие задачи:

- развить пространственное воображение у студентов;
- изучить возможности создания и использования 2D и 3D моделей, в том числе, параметрических геометрических моделей стандартных и типовых деталей;
- изучить и реализовать стандарты ЕСКД;
- изучить основы документооборота в ЕИП.

Решение первой задачи предполагает освоение студентами:

-курса: "Основы геометрического моделирования в машиностроении", включающего темы по проецированию на плоскости и правила построения аксонометрических проекций, традиционно рассматриваемые в курсе начертательной геометрии, а также формирующего умения и навыки по созданию пространственных и плоских компьютерных объектов, по

способам решения метрических и позиционных задач, по решению задач с использованием преобразований по циклу 2D-3D-2D;

-курса инженерной графики, включающего решение традиционных задач проекционного черчения.

Вторая задача - изучение основ создания объёмных моделей реальных (физических) объектов различными способами моделирования (конструктивным, B-Rep, параметрическим, прямым, гибридным и пр.), возможностей создания и использования библиотек параметрических баз 2D и 3D геометрических моделей базовых элементов формы, стандартных и типовых деталей реализуется студентами при решении разнообразных задач за время обучения на кафедре.

Освоение стандартов ЕСКД (третья задача), необходимых для выпуска технической документации, в том числе в автоматизированном режиме осуществляется студентами путем решения традиционных задач курса инженерной графики.

Для приобретения начальных сведений о работе с документами в едином информационном пространстве необходимо внедрение разработанной лабораторной работы с использованием системы ADEM v.8.1 и учетом возможностей электронного архива ADEM Vault. Цель лабораторной работы: изучить назначение и принципы функционирования системы управления и хранения конструкторской и технологической информации. Задачи работы: научиться использовать ADEM Vault в качестве хранилища данных, освоить приемы работы в среде электронного архива.

Для решения перечисленных выше задач в условиях недостатка учебного времени, работы с контингентом студентов, не имеющим должной до вузовской подготовки, пересмотрены и улучшены традиционные подходы и методики преподавания.

Во-первых, поднята роль геометро - модельной подготовки в глазах студентов как основы успешной современной конструкторско - технологической подготовки специалиста в вузе, без которой немыслимы проектирование и производство конкурентоспособной продукции.

Во-вторых, созданы и используются методические пособия и интерактивные обучающие системы для обеспечения возможности самостоятельного освоения студентами учебного материала, в первую очередь ГОСТов ЕСКД во внеаудиторное время.

В-третьих, каждое практическое занятие, связанное с созданием или редактированием технической документации в соответствии с ГОСТами ЕСКД в рамках кафедры инженерной графики или при курсовом проектировании на кафедрах конструкторского и технологического циклов предваряется прохождением контроля по теме занятия с использованием компьютера.

В-четвертых, практические занятия студенты проводят с обязательным использованием методических разработок в форме индивидуальной работы в присутствии преподавателя и оценкой результатов работы студента по итогам занятия.

В-пятых, задания студентам второго курса составляются с учетом необходимого пополнения графических баз деталей и сборочных единиц, необходимых для курсового проектирования на последующих и выпускающих кафедрах.

В-шестых, обеспечена сквозная, единообразная геометро-модельная подготовка в пределах всего процесса обучения студентов, по крайней мере, на первых трёх курсах, в частности, при изучении основ метрологии и взаимозаменяемости, обработки

конструкционных материалов, теории механизмов и машин, основ конструирования машин с использованием компьютерных средств контроля знаний предметной области (ГОСТов ЕСКД).

В заключении приведём пример (см. презентацию к лекции) сквозного конструкторско - технологического проектирования, виртуального изготовления и контроля детали "вал-шестерня" на основе использования объёмной (3D) параметрической модели её комплексного представителя.

Вопросы для самоконтроля

1. Опишите различие между проектной и аналитической моделями.
2. Почему аналитическая модель отличается от проектной?
3. Какие аналитические операции выполняются в рамках процесса разработки?
4. Как используются средства CAD в процессе разработки?
5. Какова основная функция CAD?
6. Какой вариант использования средств CAD в процессе разработки считается наиболее важным?
7. Перечислите наиболее важные типы средств CAD.
8. Как используются средства CAE в процессе проектирования?
9. Как используются средства CAM в процессе производства?
10. Какова цель курса "Основы геометрического моделирования в машиностроении"?
11. Перечислите профессиональные компетенции специалиста-машиностроителя.
12. Перечислите, что должен знать специалист по итогам изучения курса?
13. Перечислите, что должен уметь специалист по итогам изучения курса?
14. Перечислите, чем должен владеть специалист по итогам изучения курса?
15. Каковы задачи геометро-модельной подготовки на кафедре инженерной графики?
16. Объясните место параметрического моделирования в подготовке специалистов на кафедре инженерной графики.
17. Перечислите особенности геометро-модельной подготовки на кафедре инженерной графики.

Лекция 2

Объёмное моделирование твёрдого тела. Способы моделирования.

Инженеры и конструкторы имеют дело с математической (прежде всего - геометрической) моделью разрабатываемого изделия. Исторически первым и главным языком их общения (то есть языком описания инженерных моделей) был язык чертежей. Чертеж (и другие подобные графические схемы) широко использовался (и используется до сих пор) не только для описания механического изделия и его частей, но также и для описания электрических схем, архитектурных конструкций, карт местности и т. п. Четкие стандарты (как национальные, так и международные) гарантируют однозначное понимание языка чертежей всеми «читателями» - от инженера - конструктора до токаря, слесаря и фрезеровщика. Однако создание чертежей вручную - чрезвычайно дорогостоящая процедура, доступная только подготовленным специалистам и требующая использования специальной чертежной доски с линейкой - кульмана, а также разных вспомогательных средств (например, лекал для рисования кривых).

Неудивительно, что первые системы автоматизации в этой области были предназначены именно для упрощения и ускорения создания чертежей (подобно другой эпохальной концепции автоматизации с помощью компьютера - текстовым процессорам, предназначенным для упрощения создания текстовых документов и легкого внесения изменений в них). Системы класса computer aided drafting существуют и поныне, самый известных их представитель – AutoCAD. Типичная функциональность таких систем включает в себя средства, необходимые для создания и редактирования чертежей, а процесс работы концептуально не отличается от работы в графическом редакторе (основными понятиями графических моделей таких систем являются графические примитивы с атрибутами, отображаемые уровни, а также различные способы конструирования). В нашем курсе мы не будем рассматривать системы этого класса, так как в настоящее время они стремительно уступают свои позиции системам трехмерного моделирования (даже Autodesk делает основную ставку не на AutoCAD, а на Inventor Series). Дело в том, что с изобретением трехмерной компьютерной графики (возможности реалистического изображения трехмерной сцены на двумерном дисплее компьютера и ее вращения с помощью манипулятора «мышь» в воображаемом трехмерном пространстве) у инженеров появилась возможность работать напрямую с трехмерной геометрической моделью проектируемого изделия, а не с его двумерными чертежами.

Модель - такое представление данных, которое наиболее адекватно отражает свойства реального объекта, существенные для процесса проектирования.

Геометрические модели описывают объекты, обладающие геометрическими свойствами.

Геометрическое моделирование - моделирование объектов различной природы с помощью геометрических типов данных.

Геометрическое моделирование оказалось настоящим прорывом в конструировании и производстве изделий. Оно не только значительно упрощает процесс проектирования (теперь инженер-конструктор не обязан обладать развитым пространственным мышлением или использовать подручные материалы типа пластилина - он видит проектируемое изделие

непосредственно на экране), но снимает многие коммуникативные проблемы.

Геометрические модели в САПР используются для решения многих задач. В первую очередь они предназначены для хранения информации о форме объектов и их взаимном расположении и предоставления ее для обработки в удобном для компьютерной программы виде. В этом ключевое отличие электронной геометрической модели от чертежа, который представляет собой условное символично-графическое изображение, предназначенное для чтения человеком.

Представьте себе ребенка, который лепит что-то из пластилина. Малыш мнет пластилин, а иногда добавляет или отрывает кусочки, и постепенно приближается к конечному результату. Такую деятельность вполне можно назвать процессом проектирования, поскольку налицо детализация формы по мере развития идеи. Ребенок погружается в проектирование, не имея ни малейшего представления о технических чертежах, не умея пользоваться даже ручкой и бумагой. Если ребенок захочет передать результат другим, например для прототипирования и массового производства, ему достаточно отдать реальный результат своих трудов, из которого можно будет извлечь всю необходимую информацию.

Системы геометрического моделирования были созданы для того, чтобы преодолеть проблемы, связанные с использованием физических моделей в процессе проектирования. Эти системы создают среду, подобную той, в которой создаются и изменяются физические модели. Другими словами, в системе геометрического моделирования разработчик изменяет форму модели, добавляет и удаляет ее части, детализируя форму визуальной модели таким же образом, как ребенок формирует фигурку из пластилина. Визуальная модель может выглядеть точно так же, как физическая, но она нематериальна. Однако трехмерная визуальная модель хранится в компьютере вместе со своим математическим описанием, благодаря чему устраняется главный недостаток физической модели - необходимость выполнения измерений для последующего прототипирования или серийного производства.

За последние пятьдесят лет было разработано множество способов геометрического моделирования, которые мы разберем детально ниже.

Типы (способы) объёмного моделирования

Хронологически различают следующие подходы к геометрическому моделированию:

- каркасное моделирование;
- поверхностное моделирование;
- твердотельное моделирование;
- немногообразное (гибридное) моделирование.

Каркасное моделирование - это исторически первая технология представления объемной геометрии. Она естественным образом развилась из систем 2D-черчения. Это самый простой способ представления трехмерных моделей - так называемые проволочные каркасы, или просто каркасы, которые дают неоспоримые преимущества по сравнению с моделированием на плоскости. Они помогают более ясно представлять модель и надежно контролировать взаимное расположение составляющих ее элементов. Кроме того, каркасы можно использовать и для создания проекционных видов. Достаточно простые структуры данных и

алгоритмы работы с каркасами позволили реализовать их на маломощном оборудовании конца 70-х годов XX века.

В *системах каркасного моделирования (wireframe modeling systems)* форма представляется в виде набора характеризующих ее линий и конечных точек. Линии и точки используются для представления трехмерных объектов на экране, а изменение формы осуществляется путем изменения положения и размеров отрезков и точек. Другими словами, визуальная модель представляет собой каркасный чертеж формы, а соответствующее математическое описание представляет собой набор уравнений кривых, координат точек и сведений о связности кривых и точек. Сведения о связности описывают принадлежность точек к конкретным кривым, а также пересечение кривых друг с другом. Системы каркасного моделирования были популярны в ту пору, когда геометрическое моделирование только начало зарождаться. Их популярность объяснялась тем, что в системах каркасного моделирования создание форм выполнялось через последовательность простых действий, так что пользователям было достаточно легко создавать формы самостоятельно.

Недостаток каркасного представления моделей состоит в том, что программы не могут отобразить всех особенностей поверхностей, определяемых каркасами, и это делает невозможным построение, например, точных сечений. Визуальное представление достаточно аскетичное и в ряде случаев не дает возможности однозначно интерпретировать увиденное (рисунок 2.1).

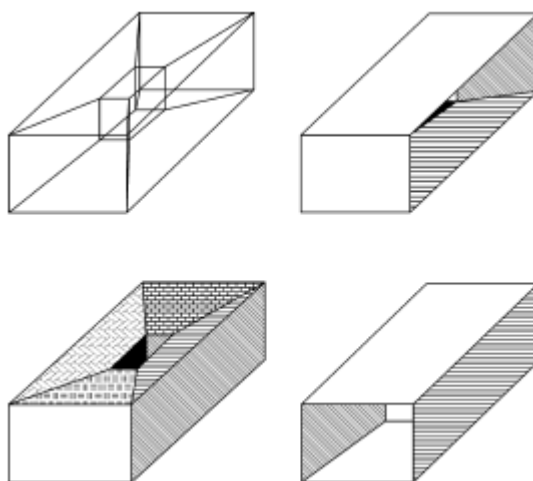


Рисунок 2.1 - Неоднозначные каркасные модели

Тем не менее даже такая, имеющая множество ограничений технология позволила существенно расширить функциональные возможности САПР по сравнению с 2D -системами. Более того, соответствующее математическое описание не содержит сведений о внутренних и внешних поверхностях моделируемого объекта. Без этих сведений невозможно рассчитать массу объекта, определить траектории перемещения инструмента при обработке объекта или создать сетку для конечно-элементного анализа, несмотря на то что объект кажется трехмерным. Поскольку эти операции являются неотъемлемой частью процесса проектирования, системы каркасного моделирования были постепенно вытеснены системами поверхностного и твердотельного моделирования.

В настоящее время построение каркасов также используется в геометрическом моделировании САПР, но лишь как вспомогательная система промежуточных построений (рисунок 2.2).

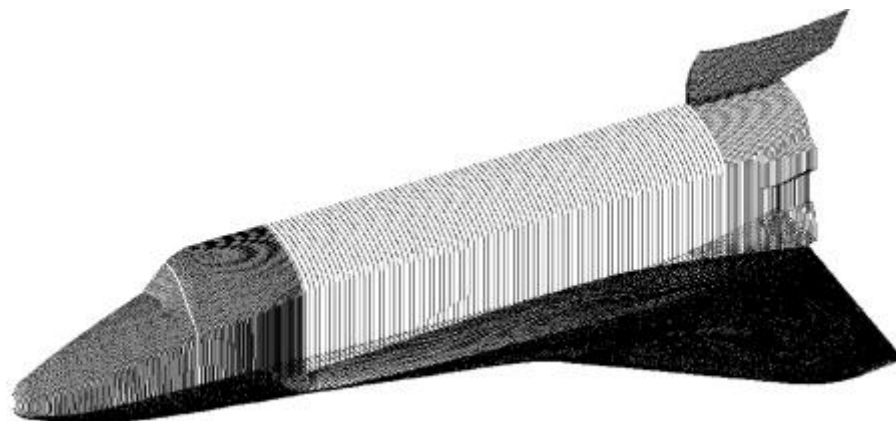


Рисунок 2.2 - Каркасная модель теоретических обводов космического корабля Буран

Поверхностное моделирование

В отличие от каркасного представления, моделирование при помощи поверхностей имеет существенно меньше ограничений, так как позволяет определить своеобразную «оболочку» трехмерного объекта (рисунок 2.3).



Рисунок 2.3 - Поверхностная модель теоретических обводов космического корабля Буран

Геометрические модели на основе поверхностного представления обеспечивают качественную визуализацию, более простой переход к построению расчетных сеток для численного моделирования, обеспечивают ряд полезных функций, таких как построение пространственных сопряжений, сечений, определения линии пересечения оболочек, генерацию чертежных проекций.

Поверхностные модели различаются по способу аппроксимации поверхности. Более простой в части структуры данных и используемых для работы с ними алгоритмов является *полигональная* аппроксимация, когда поверхность представляется набором взаимосвязанных плоских граней, на практике чаще всего треугольных. Такая аппроксимация легко строится, для нее разработаны эффективные алгоритмы реалистичной визуализации, она не требует значительных вычислительных ресурсов, хотя может быть и затратной по памяти. Главным

ограничением подобной аппроксимации является то, что она имеет фиксированную точность, то есть отклонение положения модельной поверхности от «идеальной» моделируемой. Для достижения высокой точности требуется создание сеток с малым шагом, что ведет к росту требований к вычислительным возможностям системы. Поэтому использование полигональной аппроксимации в САПР на текущий момент ограничено подсистемами визуализации и простейшего 3D-эскизирования.

Этих недостатков лишена технология NURBS (Non-Uniform Rational B-Spline, неоднородный B-сплайн), сегодня наиболее часто используемая в практике САПР.

Такое описание поверхности обеспечивает определение координат любой ее точки, радиуса кривизны в ней, направления нормали к поверхности с высокой, в общем случае (без учета вычислительных затрат) с любой наперед заданной точностью.

Определенным недостатком такого подхода является сложность алгоритмов работы с NURBS, однако это обстоятельство исторически преодолено исследователями и разработчиками.

В определенной степени NURBS-представление является развитием полигонального, но в отличие от него позволяет описывать не только плоские, но и криволинейные грани и ребра (кромки). Совокупность таких граней с общими ребрами по традиции называют сеткой.

Технология NURBS обеспечивает реализацию ряда функциональных возможностей, недоступных или существенно ограниченных при использовании каркасного либо полигонального представления: вычисление радиуса кривизны поверхностей, их гладкое сопряжение, построение траекторий на поверхности, что важно для подготовки ЧПУ-программ, получение точных изображений, спроецированных на плоскость, например, для получения чертежных видов и т. д. и т. п.

Поверхностное моделирование играет важную роль при проектировании изделий из листового металла (sheet metal parts), таких как капоты и крылья автомобилей, где форма поверхности важна как для дизайна, так и для аэродинамики изделия (рисунок 2.4).

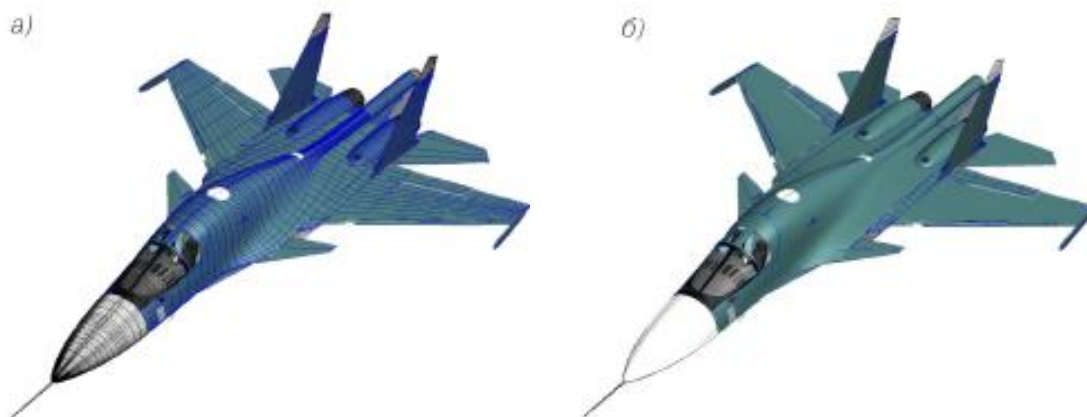


Рисунок 2.4 - Аппроксимация теоретических обводов самолета: а) полигональная и б) NURBS-аппроксимация

Традиционно используются несколько типовых **контекстов создания поверхностей**:

-плоская поверхность - получается заполнением плоского контура (2D-эскиз или набор замкнутых кромок, лежащих в одной плоскости);

-поверхность вытяжки - образуется в результате плоскопараллельного вытягивания замкнутого или разомкнутого 2D/3D-эскиза в направлении, перпендикулярном плоскости эскиза, или под произвольным углом;

-поверхность вращения - получается вращением произвольного профиля (2D -эскиз) относительно оси;

-поверхность по траектории - создается движением 2D/3D эскиза вдоль криволинейной образующей (2D/3D-эскиз, 3D-кривая) и произвольного числа направляющих кривых (2D/3D - эскиз, 3D - кривая), деформирующих исходный контур;

-поверхность по сечениям - аналог поверхности по траектории; отличается тем, что строится не по одному, а по нескольким поперечным сечениям с направляющими кривыми;

-граничная поверхность - аналог поверхности по сечениям; отличается тем, что строится по нескольким произвольно сориентированным в пространстве 3D-кромкам других поверхностей с сохранением касательности к ним и с соблюдением непрерывности по второй производной (гладкая стыковка); при построении могут использоваться направляющие кривые;

-поверхность свободной формы - строится разбиением сетки с управляющими точками на поверхности грани 3D-модели; изменение формы поверхности достигается перетаскиванием контрольных точек;

-эквидистантная поверхность - получается смещением на определенное расстояние от существующих граней или поверхностей;

-поверхность разъема - используется при проектировании литейных форм, а также в качестве вспомогательной геометрии для разделения матрицы и пуансона;

-срединная поверхность - создается на середине (или заданном проценте) толщины тонкостенной детали;

-линейчатая поверхность - строится под углом к выбранной кромке и предназначена для построения граней с уклоном.

В том случае, если установлена программная связь между вспомогательным каркасом и результирующей поверхностью, моделирование становится *ассоциативным*, при этом изменение каркасных элементов ведет к автоматическому изменению геометрии поверхностей, построенных с использованием этого контекста.

В *системах поверхностного моделирования (surface modeling systems)* математическое описание визуальной модели включает в себя не только сведения о характеристических линиях и их конечных точках, как в каркасном моделировании, но и данные о поверхностях. При работе с отображаемой на экране моделью изменяются уравнения поверхностей, уравнения кривых и координаты конечных точек. Если поверхности не окрашены и не затушеваны, визуальная модель в системе поверхностного моделирования может выглядеть точно так же, как в системе каркасного моделирования.

Математическое описание может включать сведения о связности поверхностей, то есть о том, как поверхности соединяются друг с другом и по каким кривым. В некоторых приложениях эти сведения оказываются очень полезными. Например, программа для формирования траектории перемещения фрезы с ЧПУ может воспользоваться этой информацией для проверки, не задевает ли фреза поверхности, примыкающие к обрабатываемой.

Существует *три стандартных метода* создания поверхностей в системах поверхностного моделирования.

1. Интерполяция входных точек.
2. Интерполяция криволинейных сеток.
3. Трансляция или вращение заданной кривой.

Способы ввода для каждого метода могут зависеть от конкретной системы поверхностного моделирования. Однако базовый метод ввода для каждой системы легко определить по представлению кривых и плоскостей.

Системы поверхностного моделирования используются для создания моделей со сложными поверхностями, потому что визуальная модель позволяет оценить эстетичность проекта, а математическое описание позволяет построить программу для обработки поверхностей детали на станке с ЧПУ.

Несмотря на достаточно широкие возможности, которые предоставляет поверхностное моделирование, оно имеет ряд существенных ограничений с точки зрения использования в САПР, в частности невозможность вычисления объемов, масс и моментов инерции объектов, ограниченность применения к ним булевых операций (вычитания, объединения, пересечения).

Твердотельное (объемное) моделирование - логическое развитие каркасного и поверхностного. Основной объект моделирования - трехмерное объемное тело, которое может описываться разными способами: декомпозиционным, конструктивным или граничным. Мы разберем их подробнее ниже. Главным преимуществом твердотельного моделирования перед каркасным и поверхностным является свойство физической корректности - все твердотельные модели имеют аналоги в реальном мире (чего не скажешь о каркасных и поверхностных моделях).

При использовании твердотельного моделирования, ставшего на сегодня стандартом де-факто в 3D CAD/CAM/CAE-системах снимаются ограничения поверхностного моделирования.

Системы твердотельного моделирования (solid modeling systems) предназначены для работы с объектами, состоящими из замкнутого объема, или *монолита (solid)*. В системах твердотельного моделирования, в отличие от систем каркасного и поверхностного моделирования, не допускается создание наборов поверхностей или характеристических линий, если они не образуют замкнутого объема. Математическое описание объекта, созданного в системе твердотельного моделирования, содержит сведения, по которым система может определить, где находится какая-либо точка: внутри объема, снаружи него или на его границе. По этим сведениям можно получить любую информацию об объеме тела, а значит, могут быть написаны приложения, работающие с объектом на уровне объема, а не на уровне поверхности.

Если бы система твердотельного моделирования требовала ввода всех данных для полного математического описания, она была бы слишком сложной для пользователей, и они отказались бы от нее. Процесс детализации формы не был бы похож на интуитивный процесс физического моделирования, и в результате получилось бы совсем не то, на что рассчитывали создатели систем геометрического моделирования. Поэтому разработчики систем твердотельного моделирования стараются предоставить простые и естественные функции, чтобы пользователи могли работать с объемными формами точно так же, как они работают с

физическими моделями, не вдаваясь в подробности математического описания. Функции моделирования, подобные созданию примитивов, булевым операциям, вытягиванию, построению фигуры вращения, повороту и скруглению, требуют от пользователя совсем немного. Обо всех деталях математического описания системы заботятся сами.

Существуют различные алгоритмические методы представления твердотельных моделей:

- декомпозиционные модели;
- конструктивные модели;
- граничные модели

Декомпозиционные модели являются простейшим подходом к твердотельному моделированию, представляя трехмерное тело композицией некоторых простых элементов. Различают следующие декомпозиционные модели:

- воксельное (voxel) представление;
- октантное дерево;
- ячеечное представление.

Воксельное представление - полный трехмерный аналог растрового одноцветного изображения. Тело представляется трехмерным булевым массивом, каждый элемент которого является пространственным кубиком одинакового размера со своими уникальными координатами. Такой кубик называется вокселем (voxel - от VOlume piXEL). Воксели (рисунок 2.5) равномерно покрывают всю область (тор), в которой содержится моделируемое тело. Отметим удобство воксельного представления для реализации на его основе булевых операций твердотельного моделирования. Для этого необходимо построить согласованные воксельные представления двух тел и применить соответствующую операцию к булевым значениям ячеек массива.

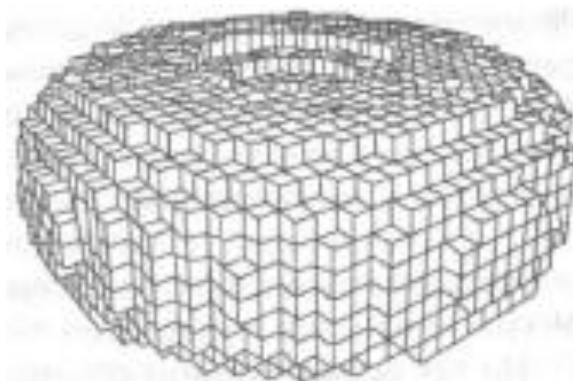


Рисунок 2.5 - Воксельное представление тора

Сложность такого алгоритма будет прямо зависеть от числа вокселей. На воксельном представлении несложно вычислять объемные параметры тела - достаточно лишь вычислить их аналитически для каждого вокселя и просуммировать.

Октантное дерево является развитием воксельного представления. Каждый узел октантного дерева соответствует некоторому кубу в трехмерном пространстве, который является либо:

- полностью (с заданной точностью) принадлежащим описываемому телу;
- полностью не принадлежащим описываемому телу;
- частично пересекающимся с описываемым телом.

Первые два типа узлов - терминальные (листья в октантном дереве), а каждый узел третьего типа обязательно имеет 8 дочерних узлов, соответствующих геометрическому разбиению его куба на 8 частей (октантов) (рисунок 2.6).

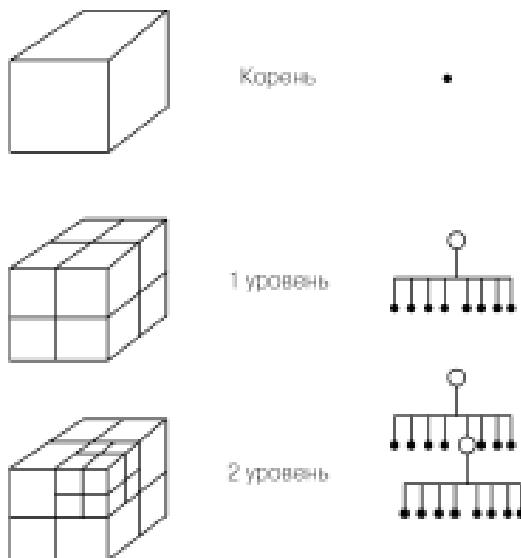


Рисунок 2.6 - Октантное дерево

Ячеечное представление соответствует разбиению моделируемого тела на произвольные непересекающиеся выпуклые многогранники, полностью (в соответствии с заданной погрешностью) заполняющие его объем.

Конструктивные модели основаны на применении операций твердотельного моделирования. Для создания конструктивной модели, модели CSG (constructive solid geometry) реализуют конструктивный подход в терминах булевых операций над параметрическими твердотельными примитивами (прямоугольный параллелепипед, цилиндр и пр.).

В рамках CSG представления для описания составных твердых тел определены следующие операции (рисунок 2.7) над исходными (а) элементарными телами:

- вычитание (б);
- объединение (в);
- пересечение (г).

Таким образом, любое составное тело может быть описано в виде традиционного уравнения из булевых функций, в котором аргументами являются либо элементарные тела, либо другие составные тела. Это представление называют деревом построений (рисунок 2.8).

Такое представление, кроме удобства модификации геометрии результирующего тела, позволяет существенно снизить требования к вычислительным ресурсам за счет применения оптимизирующих процедур к дереву построений.

Представление твердых тел в виде дерева построений удобно также и с точки зрения организации пользовательского интерфейса, обеспечивающего наглядный и быстрый доступ к любому элементу, входящему в описание геометрии тела, его модификацию и получение отчетной информации.

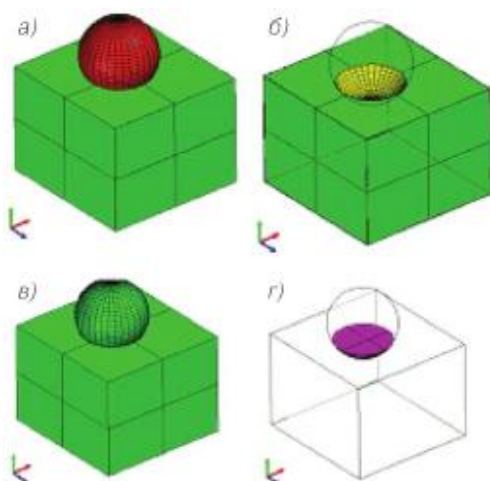


Рисунок 2.7 - Булевы операции над простыми твердыми телами

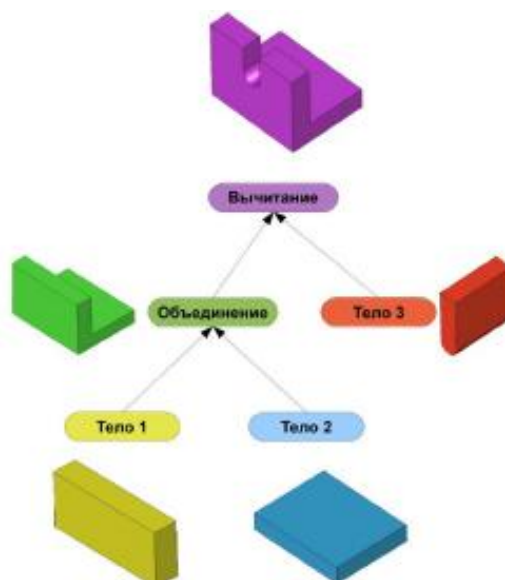


Рисунок 2.8 - Дерево построения составных твердых тел

Граничные модели

Твердое тело (рисунок 2.9) образуется в пространстве одной или несколькими поверхностями, образующими замкнутый объем. Оно включает в себя *грани*, *ребра* (или кромки) и *вершины*. **Грань** – часть граничной поверхности, образующей тело, граница которой состоит из криволинейных сегментов, при пересечении которых происходит существенное изменение вектора нормали к поверхности, т.е. нарушается непрерывность изменения кривизны поверхности (рисунок 2.10). Криволинейные сегменты, ограничивающие грань, называются **ребрами**. Точки, в которых встречаются соседние ребра, называются **вершинами**.

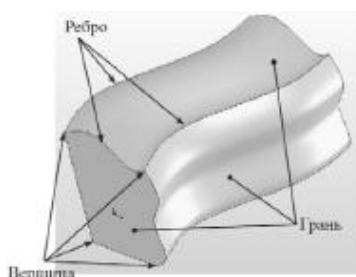


Рисунок 2.9 - Твердое тело

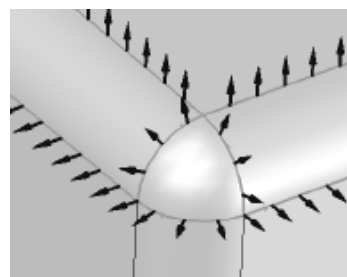


Рисунок 2.10 - Грани, ребра и нормали к поверхности

Граничные модели хранят информацию о границах тела (гранях, ребрах и вершинах). Для простоты манипулирования эта информация подразделяется на геометрические и топологические данные.

Геометрические данные для каждой граничной сущности свои:

- для вершины - ее координаты;
- для ребра - параметрическое уравнение кривой (прямой);
- для грани - параметрическое уравнение поверхности либо тип и набор параметров в случае канонической поверхности (плоскости, сферы, цилиндра, конуса, тора).

Топологические данные - это информация о смежности вершин и ребер, ребер и граней, а также о внутренних и внешних границах грани. Для удобного манипулирования

топологической информацией было предложено несколько структур данных, называемых **BRep** (Boundary Representation):

- многогранные (фасетные) модели;
- вершинные модели;
- полурёберные модели;
- крыльевые рёберные модели.

Разберем одну из самых популярных структур – полурёберную, основанную на том простом факте, что каждое ребро границы твердого тела принадлежит ровно двум граням (рисунок 2.11).

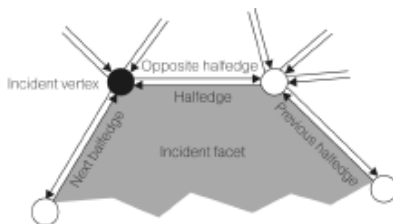


Рисунок 2.11 - Полурёберная модель

Структура данного представления такова:

- список тел (solid), каждое тело состоит из списка его граней (faces), ребер (edges) и вершин (vertices);
- грань состоит из колец (loop), представляющих собой внешнюю границу грани, а также ее опциональные внутренние границы;
- кольцо состоит из списка полурёбер (halfedges);
- полуребро указывает на начальную вершину и следующее полуребро, а также на свое ребро;
- ребро хранит указатели на два своих полуребра.

Одним из достоинств граничной модели является удобный способ расчета ее объемных параметров.

В практике САПР наиболее широкое применение имеет технология, базирующаяся на граничном представлении элементарных односвязных тел (BRep, Boundary Representation) в совокупности с конструктивной геометрией (CSG, Constructive solid geometry), описывающей операции над телами.

Граничное представление определяет сплошное тело неявно путем описания ограничивающей его поверхности. Суть BRep-представления заключается в том, что твердое тело описывает замкнутая пространственная область, ограниченная набором элементарных тонких поверхностей (граней) с общими образующими контурами (ребрами) на границе поверхностей и признаком внешней или внутренней стороны поверхности (рисунок 2.12), а также обеспечивающим следующий ряд операций, определенных над телами:

- проверка правильности задания, для односвязных тел осуществляется по формуле Эйлера, в наиболее общем виде записываемой как:

$$V - E + F = 2,$$

- где **V**-количество вершин, **E**- количество ребер, **F** - количество граней;
- вычисление габаритного объема;
 - вычисление нормали в точке;
 - вычисление кривизны поверхности;
 - нахождение точки пересечения с контуром или другой поверхностью;
 - определение положения точки относительно поверхности.

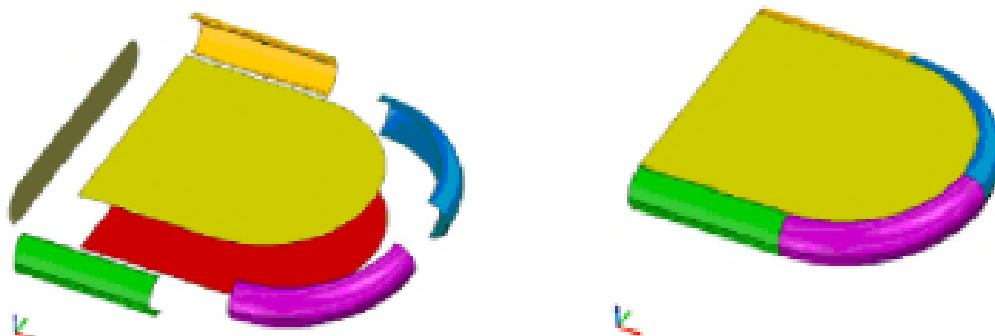


Рисунок 2.12 - BRep-представление простых твердых тел

Для описания сложных тел, моделирующих объекты реального мира, получаемые обработкой материала или неразъемной сборкой, используется иерархическая структура, описывающая тела как последовательность применения булевых операций над набором элементарных твердых тел - так называемое CSG-дерево (Constructive Solid Geometry tree) (рисунок 2.13, см. также рисунок 2.8).

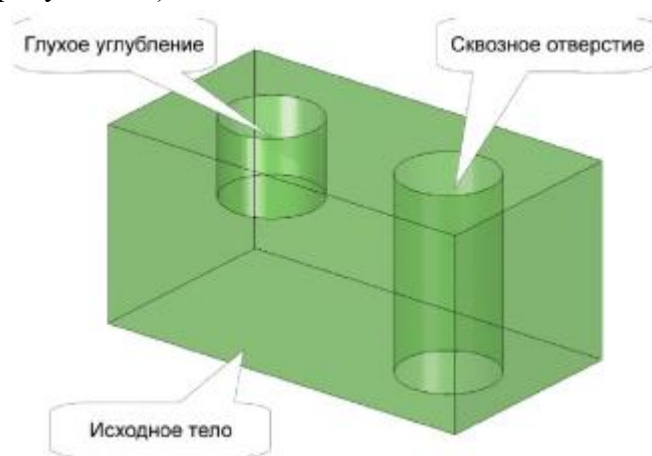


Рисунок 2.13 - BRep-представление сложных твердых тел

Так же как и для элементарного тела, правильность построения определяется формулой Эйлера, записываемой для многосвязных тел в частном виде:

$$V - E + F - H = 2(C - G),$$

где **V** - количество вершин, **E** - количество ребер, **F** - количество граней, **H** - количество несквозных отверстий, **C** - количество компонент, **G** - количество сквозных отверстий.

Немногообразное (гибридное) моделирование снимает ограничения, присущие классическому твердотельному моделированию: с его помощью можно описывать

геометрические модели, которые локально могут быть не только многообразиями размерности три (объемными телами), но и размерности два (поверхностями), один (кривыми), нуль (точками), а также участками сопряжения многообразий разной размерности.

Вопросы для самоконтроля

1. Дайте определение модели.
2. Дайте определение геометрической модели.
3. В чём суть геометрического моделирования?
4. Для чего используются геометрические модели в САПР?
5. Назовите способы геометрического моделирования
6. В чём состоит способ каркасного моделирования?
7. Каковы недостатки каркасного моделирования?
8. В чём суть поверхностного моделирования?
9. Назовите способы аппроксимации поверхности.
10. В чем преимущества и недостатки каркасной и полигональной аппроксимации трехмерной геометрии?
11. В чем заключается преимущество технологии NURBS?
12. Перечислите типовые способы создания поверхностей (11).
13. Объясните суть создания плоской поверхности, поверхности вытяжки, поверхности вращения, поверхности по траектории.
14. Объясните суть создания поверхности по сечениям, граничной поверхности, поверхности свободной формы.
15. Объясните суть создания эквидистантной поверхности, поверхности разъёма, срединной поверхности, линейчатой поверхности.
16. В чём суть ассоциативного моделирования поверхностей?
17. Какую информацию содержит математическое описание поверхности?
18. Назовите три стандартных метода создания поверхностей в системах поверхностного моделирования. В чём заключается каждый из них?
19. В чём суть твёрдотельного моделирования?
20. В чём отличие твёрдотельного моделирования от каркасного и поверхностного?
21. Назовите методы (3) представления твёрдотельных моделей.
22. Объясните суть представления твёрдого тела в виде декомпозиционных моделей: воксельного (voxel) представления, октантного дерева, ячеечного представления. В чём проявляются недостатки такого метода?
23. Объясните суть представления твёрдого тела в виде конструктивной модели (CSG). Назовите операции над исходными элементарными и составными телами.
24. Объясните суть представления твёрдого тела в виде граничной модели (B-Rep).
25. Приведите формулы Эйлера для представления простых и сложных тел.
26. Раскройте суть метода истории построения геометрии.
27. В чём суть немногообразного (гибридного) моделирования?
28. Какое представление геометрии наиболее оптимально для САПР?

Лекция 3

Функции твёрдотельного моделирования. Кривые линии.

Функции моделирования, поддерживаемые большинством систем твердотельного моделирования, могут быть разделены на пять основных групп.

В первую группу входят функции, используемые для создания простых форм на основе объемных заготовок, имеющихся в программе, - так называемые *функции создания примитивов (primitive creation functions)*. К этой же группе относятся функции добавления и вычитания объема - *булевы операторы (Boolean operations)*. Функции моделирования из первой группы позволяют проектировщику быстро создать форму, близкую к окончательной форме детали, подобно тому, как ребенок сминает пластилин и создает из него приблизительную физическую модель.

Ко второй группе относятся функции создания объемных тел путем перемещения поверхности. Функция *заметания (sweeping)* позволяет создавать объемное тело трансляцией или вращением области, заданной на плоскости. Построение тела вращения из плоской кривой называется также *качанием* или *вращательным заметанием (swinging)*. Задавая замкнутую плоскую область, пользователь может указывать геометрические ограничения или вводить данные о размерах, а не рисовать форму вручную. Здесь под геометрическими ограничениями понимаются соотношения между элементами рисунка (перпендикулярность отрезков, касание дуги окружности отрезком и т. д.). В этом случае система построит точную форму, удовлетворяющую ограничениям, самостоятельно. Изменение геометрических ограничений или размеров даст другую плоскую область и другое объемное тело. Такой подход называется *параметрическим моделированием*, поскольку изменение параметров позволяет получить разные объекты. Параметрами могут быть постоянные, входящие в геометрические ограничения, а также размеры. Функция *скиннинга (skinning)* создает объемное тело, натягивая поверхность на заданные поперечные сечения. Функции второй группы позволяют проектировщику начать моделирование с формы, весьма близкой к конечному результату, поскольку одних поперечных сечений вполне достаточно для точного описания конечного объемного тела.

В третью группу входят функции моделирования, предназначенные главным образом для изменения существующей формы. Типичными примерами являются функции *скругления* или *плавного сопряжения (rounding, blending)* и *поднятия (lifting)*.

К четвертой группе относятся функции, позволяющие непосредственно манипулировать составляющими объемных тел, то есть вершинами, ребрами и гранями. Работа с этими функциями (аналогичными функциям систем поверхностного моделирования) называется *моделированием границ (boundary modeling)*.

В пятую группу входят функции, используя которые проектировщик может моделировать твердое тело при помощи свободных форм. Например, он может давать системе команды типа «сделать отверстие такого-то размера в таком-то месте» или «сделать фаску такого-то размера в таком-то месте». Работа с такими функциями называется *объектно-ориентированным моделированием (feature-based modeling)*. В последнее время функциям пятой группы уделяется особое внимание, поскольку модель, построенная с их помощью, содержит информацию о процессе создания, без которой невозможно автоматическое

формирование плана технологического процесса для детали. Заметьте, что модель, созданная другими средствами, содержит только элементарные геометрические сведения о вершинах, ребрах и гранях.

I. Функции создания примитивов позволяют выбирать и создавать простейшие объекты, заранее определенные авторами системы моделирования. Размер примитива задается пользователем. Примитивы, поддерживаемые большинством систем твердотельного моделирования, показаны на рисунке 3.1. Размеры, указанные на этом рисунке, могут устанавливаться пользователем. Примитивы сохраняются в базе данных процедурой, осуществляющей их создание, а параметры примитивов передаются этой процедуре в качестве аргументов.

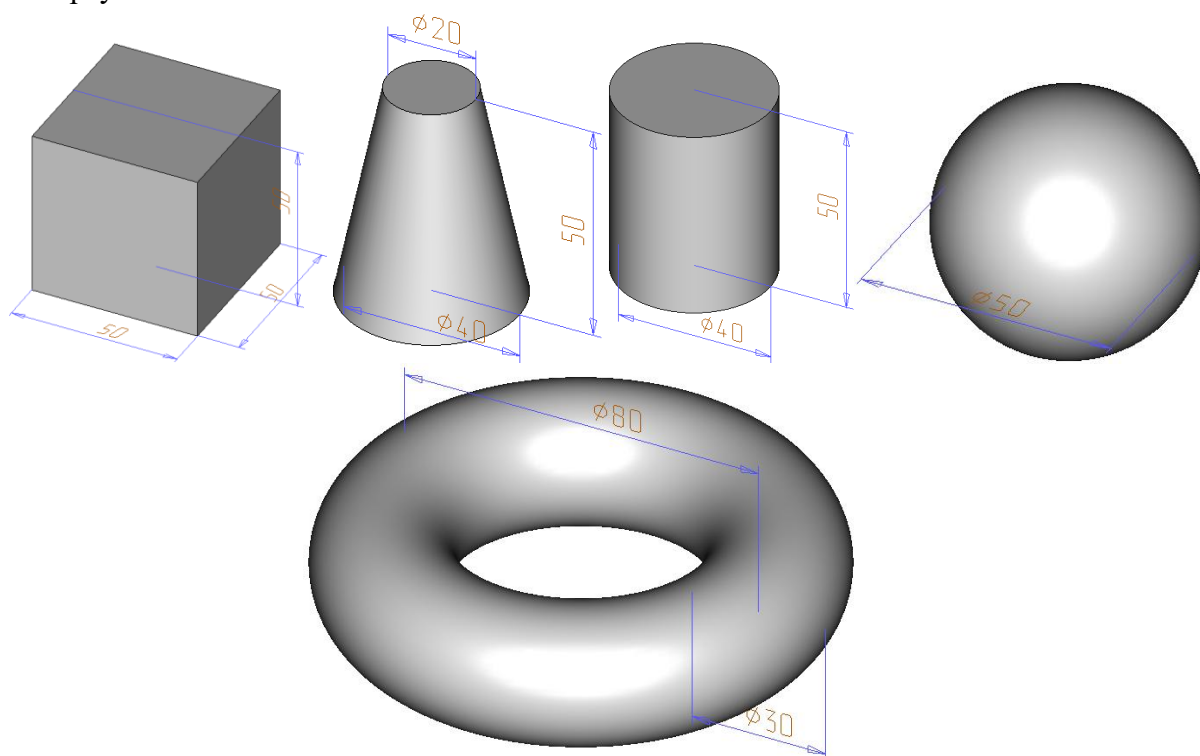


Рисунок 3.1 - Типичные примитивы

Булевы операции. Если бы в списке примитивов можно было найти любое объемное тело, это было бы замечательно. Однако из-за разнообразия возможных применений систем геометрического моделирования сохранить заранее все мыслимые формы невозможно. Гораздо проще приблизиться к решению, предоставив пользователю средства для комбинирования примитивов. В качестве метода комбинирования в твердотельном моделировании применяются булевы операции теории множеств. Другими словами, каждое примитивное объемное тело считается множеством точек, к множествам применяются булевы операции, а в результате получается объемное тело, состоящее из точек, полученных после преобразований.

Большинством систем твердотельного моделирования поддерживаются следующие булевы операции: объединение, пересечение и разность (рисунки 3.2, 3.3 и 3.4 соответственно). До применения булевых операций необходимо определить относительное

положение и ориентацию примитивов. Булевы операции могут применяться не только к примитивам, хотя на рисунках в качестве примеров изображены именно примитивы.

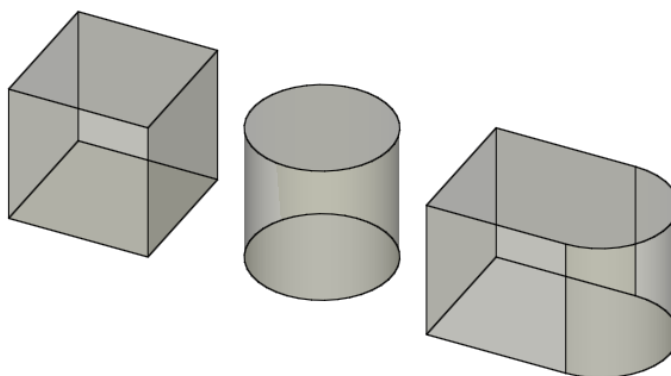


Рисунок 3.2 - Объединение примитивов

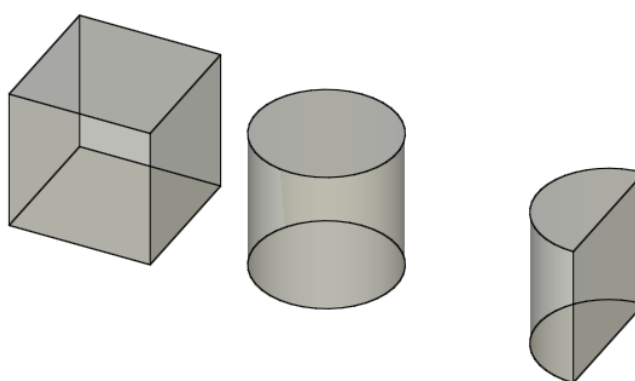


Рисунок 3.3 - Пересечение примитивов

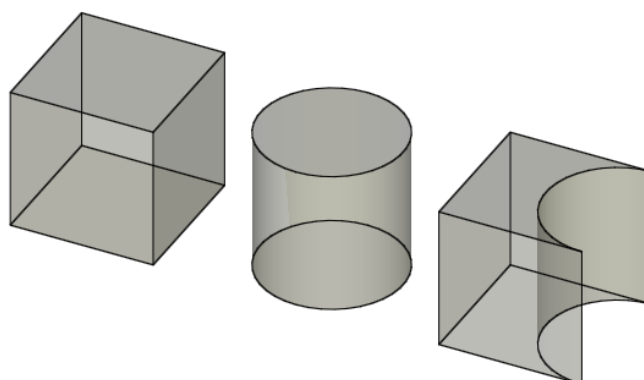


Рисунок 3.4 - Разность примитивов

Еще одна функция моделирования реализуется подобно булевым операциям, это функция разрезания объемного тела плоскостью, после применения которой случается тело из двух частей. Того же результата можно достичь, применив операцию вычитания к объемному телу, которое должно быть разрезано, и кубу, одной из граней которого является секущая плоскость. По этой причине функция разрезания также может быть отнесена к булевым операциям.

При использовании булевых операций следует быть внимательным, чтобы не получить в результате тело, не являющееся объемным (рисунок 3.5). Некоторые системы выдают предупреждение о возможности получения некорректного результата, другие могут просто

завершить работу с сообщением об ошибке. Системы гибридного (немногообразного) моделирования способны обрабатывать и такие специфические ситуации, поскольку они работают не только с объемными телами, но и с поверхностями и каркасами.

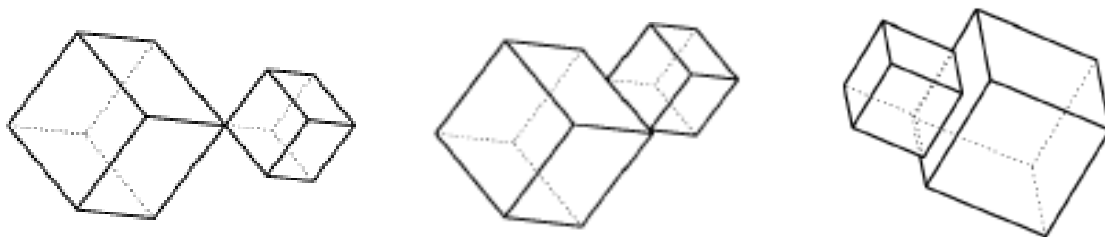


Рисунок 3.5 - Некорректная булева операция

II.Заметание. Функция **заметания** (*sweeping*) формирует объемное тело трансляцией или вращением замкнутой плоской фигуры. В первом случае процесс формирования называется **заметанием при трансляции** (*translational sweeping*), во втором случае — **построением фигуры вращения** (*swinging, rotational sweeping*). Если плоская фигура будет незамкнутой, в результате заметания получится не объемное тело, а поверхность. Такой вариант заметания поддерживается системами поверхностного моделирования.

Заметание при трансляции и вращении представлено на рисунках 3.6 и 3.7 соответственно. Хотя рисунок 3.7 демонстрирует вращение на 360°, большинство систем твердотельного моделирования позволяют поворачивать фигуру на произвольный угол.

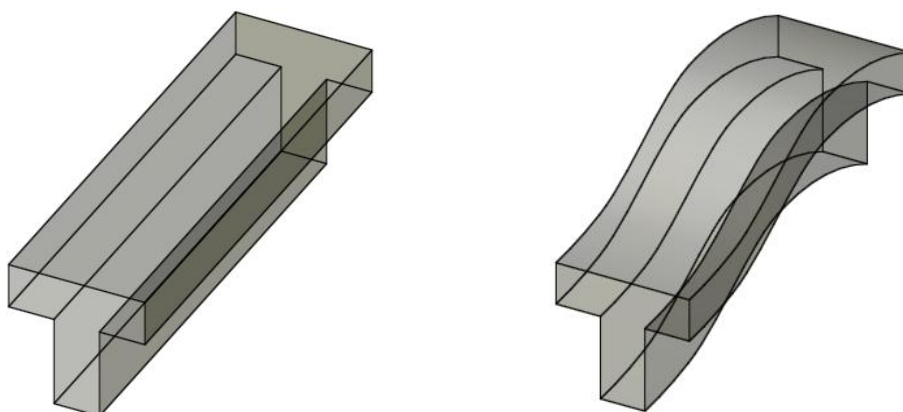


Рисунок 3.6 - Заметание при трансляции

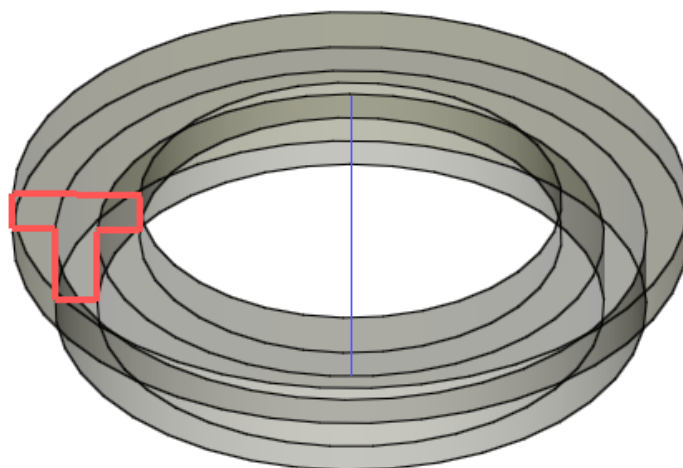


Рисунок 3.7 - Заметание при вращении

Скиннинг. Функция *скиннинга* (*skinning*) формирует замкнутый объем, натягивая поверхность на заданные плоские поперечные сечения тела (рисунок 3.8). Можно представить себе, что на каркас фигуры, образованный границами поперечных сечений, натягивается ткань или винил. Если к натянутой поверхности не добавить конечные грани (два крайних сечения), в результате получится поверхность, а не замкнутый объем. В таком варианте функция скиннинга представлена в системах поверхностного моделирования.

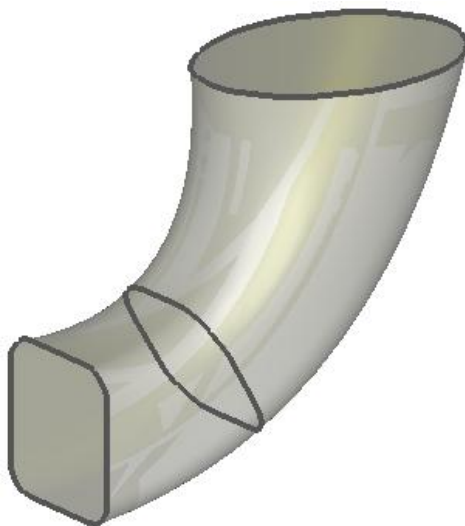


Рисунок 3.8 - Создание объемного тела методом скиннинга

III.Скругление или плавное сопряжение. Скругление (*rounding*), или плавное сопряжение (*blending*), используется для модифицирования существующей модели, состоящего в замене острого ребра или вершины гладкой криволинейной поверхностью, векторы нормали к которой непрерывно продолжают векторы нормали поверхностей, сходящихся у исходного ребра или вершины. Замену острого прямого ребра цилиндрической поверхностью демонстрирует рисунок 3.9а. Векторы нормали к цилиндрической поверхности продолжают векторами соседних плоских граней. Замена острой вершины сферической поверхностью показана на рисунке 3.10. Здесь также обеспечивается непрерывность векторов нормали. Частный случай скругления с добавлением, а не удалением материала показан на рисунке 3.9б. Такая процедура называется выкружкой (*filleting*).

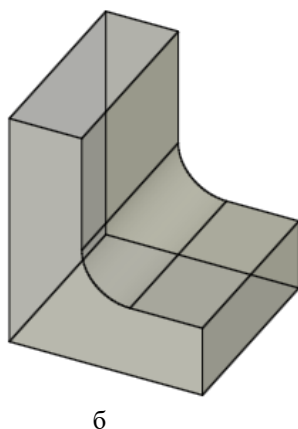
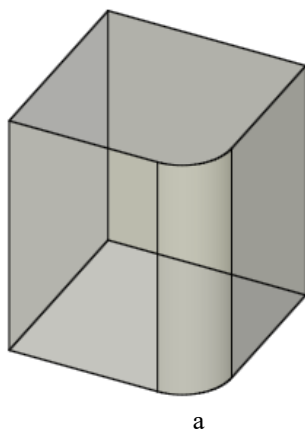


Рисунок 3.9 - Скругление ребер

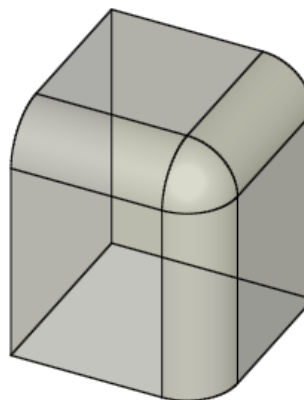


Рисунок 3.10 - Скругление вершин

Поднятие. Поднятием (*lifting*) называется перемещение всей грани объемного тела или ее части в заданном направлении с одновременным удлинением тела в этом направлении (рисунок 3.11а). Если нужно поднять только часть грани (рисунок 3.11б), эту грань необходимо заранее разделить.

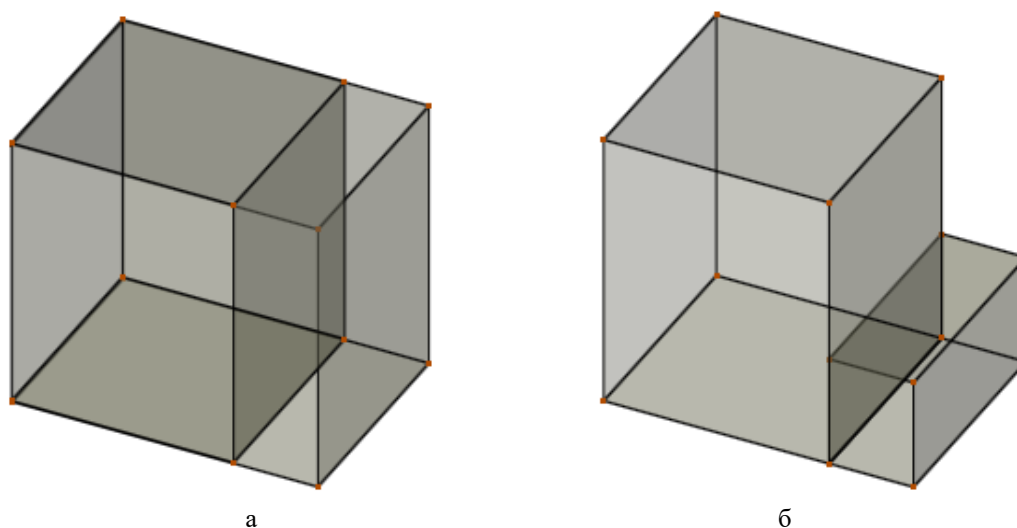


Рисунок 3.11 - Поднятие грани и ее части

Для этого достаточно добавить ребро, которое будет разбивать грань на нужные части. Но внутри системы при этом будут произведены некоторые дополнительные действия, результатом которых станет деление грани. Обычно при этом обновляются сведения о связности поверхности. Внутри системы подобные действия выполняются при помощи операторов Эйлера. Чтобы добавленная часть тела не пересекалась с исходной (рисунок 3.12) при работе с функцией поднятия необходимо правильно указывать направление и величину поднятия. Пересечение не вызовет проблем, если функция поднятия реализована так, что при пересечении выполняется объединение добавленной части с исходным телом. Однако изначально функция поднятия разрабатывалась для небольших местных изменений, поэтому ситуация, показанная на рисунке 3.12, дает некорректное объемное тело.

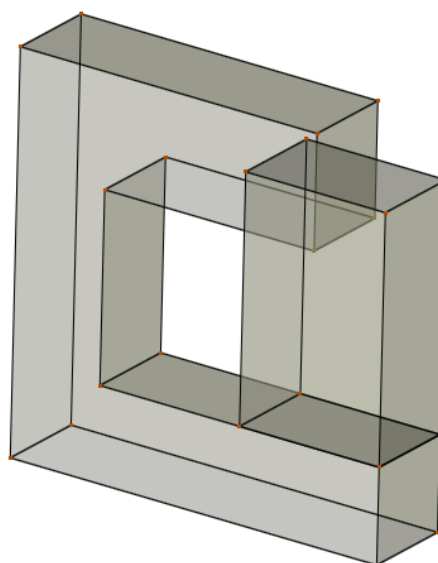


Рисунок 3.12 - Самопересечение в результате поднятия

IV. Моделирование границ. Функции моделирования границ используются для добавления, удаления и изменения элементов объемного тела - его вершин, ребер и граней. Следовательно, процедура, использующая функции моделирования границ, будет выглядеть точно так же, как в системах поверхностного моделирования. Другими словами, вначале создаются точки, затем создаются ребра, соединяющие эти точки, и наконец, граничные ребра определяют поверхность. Однако в системах твердотельного моделирования, в отличие от систем поверхностного моделирования, нужно определить все поверхности таким образом, чтобы образовался замкнутый объем. Создание клина при помощи функций моделирования границ иллюстрирует рисунок 3.13. Процедура включает создание точек, ребер, граней и поверхностей.

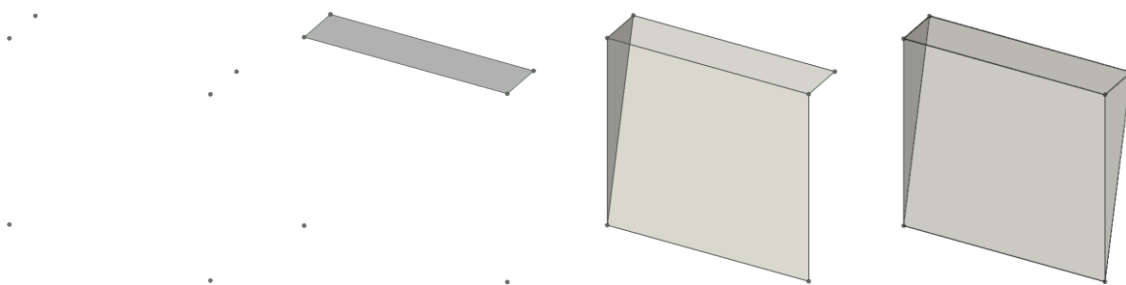


Рисунок 3.13 - Создание тела при помощи функций моделирования границ

Создавать объемное тело исключительно при помощи функций моделирования границ очень утомительно. Эти функции используются главным образом для создания плоских фигур, которые затем служат сечениями объемным телам, образуемым заметанием или скиннингом. Однако функции моделирования границ удобно применять для изменения формы уже существующего тела. Вершину можно передвинуть в новое положение, изменив соответствующим образом соседние ребра и грани (рисунок 3.14). Прямое ребро можно заменить криволинейным, в результате чего изменятся связанные грани и вершины (рисунок 3.15). Плоскую поверхность можно заменить на криволинейную с модификацией ребер и вершин (рисунок 3.16). Иногда плоскую поверхность можно заменять криволинейной, импортированной из системы поверхностного моделирования. Перечисленные функции моделирования называются *функциями тонкого редактирования (tweaking functions)*. Они используются для моделирования тел, ограниченных криволинейными поверхностями, поскольку такие поверхности легко получать из плоских граней многогранника.

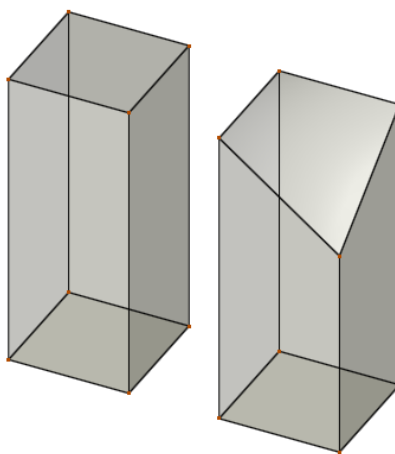


Рисунок 3.14 - Модификация объекта перемещением вершины

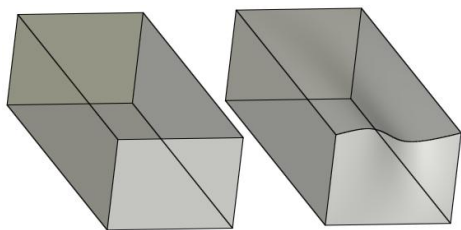


Рисунок 3.15 - Модификация объекта заменой ребра

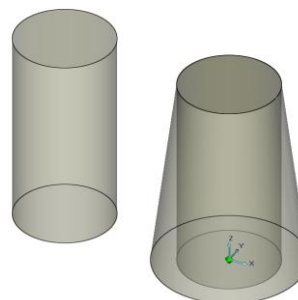


Рисунок 3.16 - Модификация объекта заменой поверхности

V. Объектно-ориентированное моделирование.

Объектно-ориентированное моделирование (feature-based modeling) позволяет конструктору создавать объемные тела, используя привычные *элементы форм [features]*. Созданное тело несет в себе информацию об этих элементах в дополнение к информации об обычных геометрических элементах (вершинах, ребрах, гранях и др.). Например, конструктор может давать команды типа «сделать отверстие такого-то размера в таком-то месте» или «сделать фаску такого-то размера в таком-то месте», и получившаяся фигура будет содержать сведения о наличии в конкретном месте отверстия (или фаски) конкретного размера. Набор доступных в конкретной программе элементов формы зависит от спектра применения этой программы.

Большинством систем объектно-ориентированного моделирования поддерживаются такие элементы, которые используются при изготовлении деталей: фаски, отверстия, скругления, пазы, выемки и т. д. Такие элементы называются *производственными*, поскольку каждый из них может быть получен в результате конкретного процесса производства. Например, отверстие создается сверлением, а выемка — фрезерованием. Следовательно, на основании сведений о наличии, размере и расположении производственных элементов можно попытаться автоматически сформировать план технологического процесса. Автоматическое планирование технологического процесса, если оно будет разработано на практическом уровне, перебросит мост между CAD и CAM, которые сейчас существуют отдельно друг от друга. Таким образом, в настоящий момент лучше моделировать объекты, подобные изображенному на рисунке 3.17, с использованием команд объектно-ориентированного моделирования «Паз» и «Отверстие», а не просто булевых операций. Модель, созданная при помощи таких команд, облегчит планирование технологического процесса, если не сделает его полностью автоматическим. Использование производственных элементов в моделировании иллюстрирует рисунок 3.18.

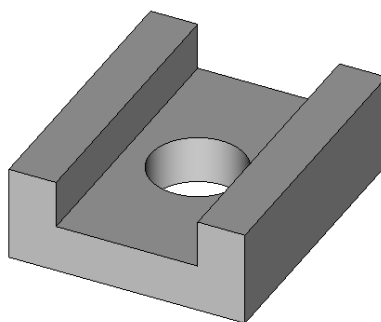


Рисунок 3.17 - Модель, созданная командами "Паз" и "Отверстие"

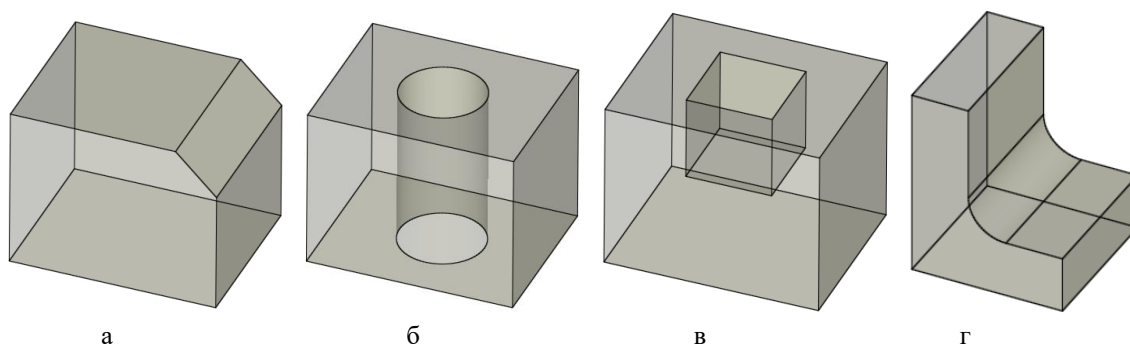


Рисунок 3.18 - Производственные элементы: а - фаска; б - отверстие; в - колодец; г - скругление

Один из недостатков объектно-ориентированного моделирования заключается в том, что система не может предоставить все элементы, нужные для всех возможных приложений. Для каждой задачи может потребоваться свой набор элементов. Чтобы исключить этот недостаток, большинство систем объектно-ориентированного моделирования поддерживают какой-либо язык, на котором пользователь при необходимости может определять свои собственные элементы. После определения элемента необходимо задать параметры, указывающие его размер. Элементы, как и примитивы, могут быть разного размера, а задаются размеры параметрами в момент создания элемента. Создание элементов разного размера путем присваивания различных значений соответствующим параметрам является разновидностью параметрического моделирования.

Моделирование кривых линий и поверхностей

Все не прямые и не ломаные линии называются кривыми. Кривые линии разделяются на два вида:

- плоские кривые, т.е. такие, все точки которых располагаются в одной плоскости;
- пространственные кривые (линии двойной кривизны), т. е. такие, точки которых не принадлежат одной плоскости.

Если закон перемещения точки может быть выражен аналитически в виде уравнения, то образующая при этом линия называется закономерной, в противном случае - не закономерной, или графической. Закономерные кривые линии делятся на алгебраические, определяемые алгебраическими уравнениями (эллипс, парабола, гипербола и др.), и трансцендентные, определяемые трансцендентными уравнениями (синусоида, циклоида, спираль Архимеда и др.). Важной характеристикой алгебраической кривой является её порядок (трансцендентные кривые порядка не имеют). С алгебраической точки зрения порядок кривой линии равен степени её уравнения, с геометрической - наибольшему числу точек пересечения кривой с прямой линией для плоских кривых и с произвольной плоскостью для пространственных. В число точек пересечения включаются как действительные точки, так и совпавшие и мнимые. Например, прямую линию, имеющую уравнение первой степени $ax+by+c=0$ (с произвольной прямой пересекается в одной точке), можно рассматривать как линию первого порядка, эллипс - кривая второго порядка, имеет уравнение $x^2/a^2+y^2/b^2=1$ второй степени, пересекается с прямой максимум в двух точках. Кривыми второго порядка являются также окружность, парабола, гипербола. Примерами кривых третьего порядка могут

служить строфоида, Декартов лист, циссоида; четвертого - лемниската Бернулли, кардиоида, улитка Паскаля.

Прямая, пересекающая кривую линию в одной, двух и более точках, называется секущей (прямая m) (рисунок 3.19). Касательной прямой t в данной точке A линии l называется предел, к которому стремится секущая (AB), когда точка B , оставаясь на линии l , стремится к точке A . Касательная к прямой линии согласно этому определению есть сама прямая. Нормалью к кривой l называется прямая n , перпендикулярная к l и проходящая через точку касания A .

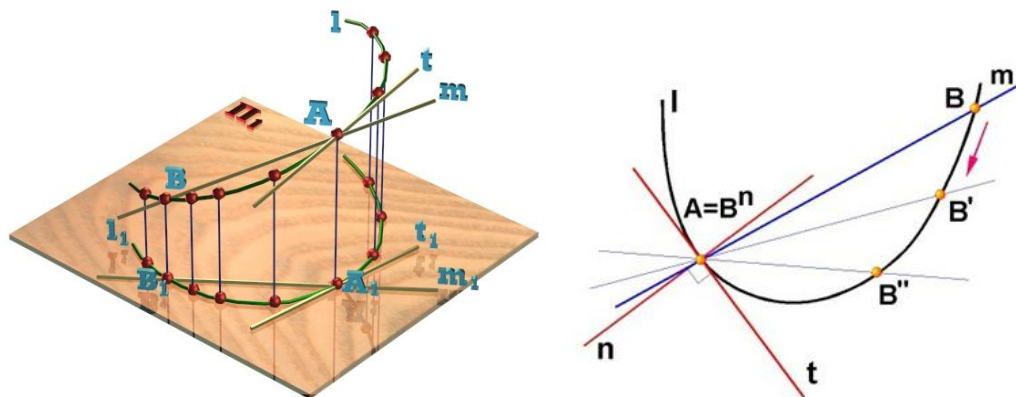


Рисунок 3.19 - Кривая линия

Кривая второго порядка имеет уравнение второй степени в декартовой системе координат. С прямой линией пересекается в двух точках (действительных, совпавших или мнимых).

Эллипс - геометрическое место точек, сумма расстояний которых до двух заданных точек (фокусов) - величина постоянная, равная $2|a|$ (длине большой оси эллипса). Эллипс не имеет несобственных точек.

Парабола - геометрическое место точек, равноудаленных от данной точки F (фокуса) и данной прямой d (директрисы). Парабола имеет одну несобственную точку.

Гипербола - геометрическое место точек, разность расстояний которых до двух заданных точек (фокусов) - величина постоянная, равная $2|a|$ (расстоянию между вершинами гиперболы). Гипербола имеет две несобственные точки, по одной на каждой асимптоте.

Кривые второго порядка - эллипс, окружность, парабола и гипербола - могут быть получены при пересечении конуса плоскостью и поэтому называются **коническими сечениями**.

Из закономерных пространственных кривых наибольшее практическое применение находят винтовые линии, в частности, цилиндрическая винтовая линия (рисунок 3.20).

Цилиндрическая винтовая линия представляет собой пространственную кривую, описываемую точкой, совершающей равномерно-поступательное движение по образующей цилиндра вращения, которая в свою очередь вращается вокруг оси цилиндра с постоянной угловой скоростью. Величина P , на которую поднимается точка за один оборот образующей, называется шагом винтовой линии. Горизонтальная проекция винтовой линии является окружностью, а фронтальная - синусоидой. На развертке цилиндрической поверхности винтовая линия изобразится в виде прямой. Угол подъема винтовой линии равен углу наклона касательной t в любой точке винтовой линии к плоскости, перпендикулярной её оси.

Цилиндрическая винтовая линия, подобно прямой и окружности, обладает свойством сдвигаемости. Свойство сдвигаемости состоит в том, что каждый отрезок линии может сдвигаться вдоль неё, не подвергаясь деформации. Это свойство винтовой линии лежит в основе работы винтовых пар (винт-гайка). Винтовая линия является геодезической на цилиндрической поверхности. *Геодезической* называется линия, принадлежащая поверхности и кратчайшая из всех линий, которые можно провести между двумя точками поверхности. Кроме цилиндрической винтовой линии, геодезическими линиями также являются прямая на плоскости, окружность большого круга на сфере и др. Геодезическая линия изображается на развертке поверхности в виде прямой линии.

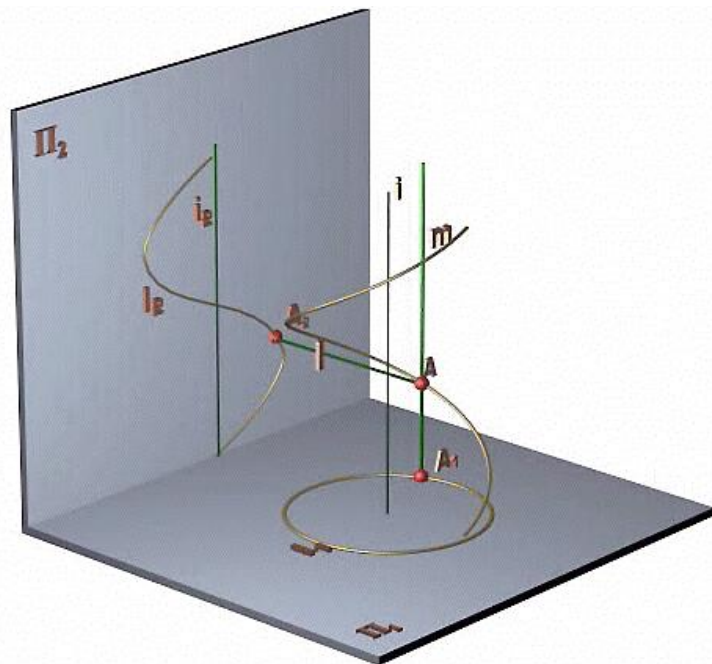


Рисунок 3.20 - Винтовая линия

В большинстве случаев с помощью тел и поверхностей, называемых аналитическими, нельзя описать реальные геометрические объекты. На протяжении долгого времени выход из этого положения заключался в представлении объекта с помощью произвольно проведенных линий, взаимное сопряжение которых осуществлялось специалистами очень высокой квалификации: модельщиками, литейщиками, изготовителями штампов. Решение этой задачи в разных областях техники характеризовалось общим свойством - отсутствием какого бы то ни было предварительного математического определения, даже частичного, в результате чего решение задачи начиналось с выполнения последовательных шагов аппроксимации и заканчивалось вручную. Каждый шаг подгонки основывался на результатах экспериментов или просто на указаниях дизайнера. Из-за недостатка информации такие зависимости часто назывались экспериментальными.

В инженерной геометрии существует большое количество отработанных графических способов построения плавных кривых - обводов. Они традиционно применялись при проектировании таких объектов, как корпус корабля, кузов автомобиля, фюзеляж и крыло самолета и многих других (рисунок 3.21).

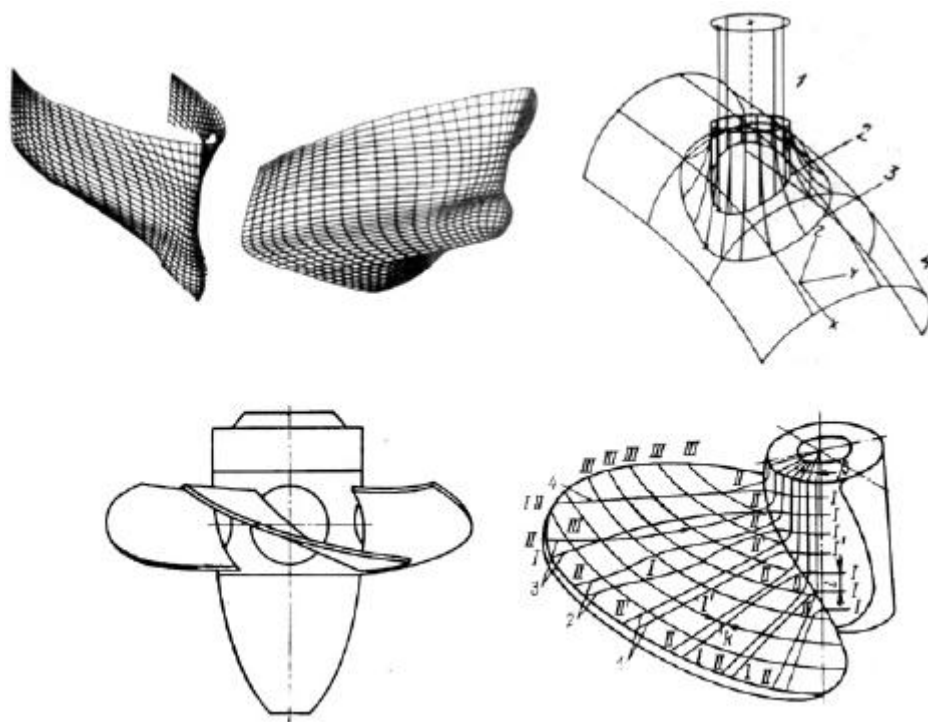


Рисунок 3.21 - Примеры кривых поверхностей в технике

В результате форма тела задается при помощи набора очень точно изготовленных сечений - плазов. Кроме перечисленных примеров сложную форму поверхности имеют лопасти гидравлических и паровых турбин, гребные винты, лопатки авиационных двигателей, крыльчатки и др., т. е. объекты, функциональные требования к которым связаны с омытием какой-либо средой. Кроме функциональных требований к изделиям предъявляются и эстетические. В этом случае также решаются задачи проектирования кривых линий и поверхностей произвольной формы.

Появление ЭВМ позволило перейти от «ручных» графических способов моделирования к более эффективному способу задания кривых и поверхностей. Процесс проектирования происходит следующим образом: задают координаты сравнительно небольшого числа опорных точек (например расчетных или полученных в результате эксперимента), лежащих на искомой линии или поверхности, и через эти точки проводят плавные кривые линии. Такое конструирование носит итерационный характер.

Модель (аналитическое описание), полученную на некотором шаге итерации, модифицируют и улучшают до тех пор, пока не будет достигнута желаемая форма кривой или поверхности. Проблема состоит в математически точном воспроизведении формы изделия, исходя из координат точек, расположенных на его поверхности.

При моделировании кривых линий и поверхностей, их машинного представления возникают задачи аппроксимации, интерполяции и сглаживания исходных данных (рисунок 3.22).

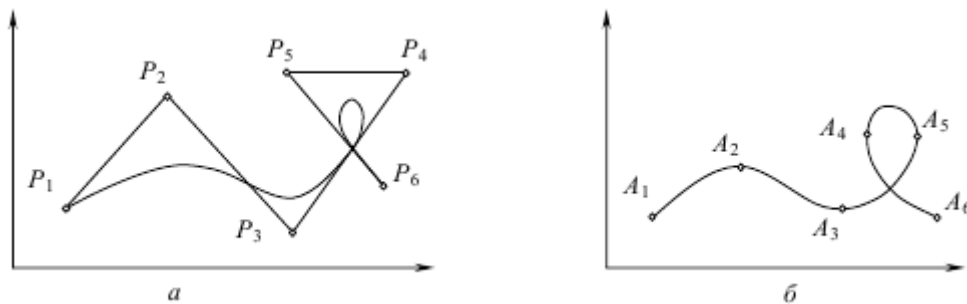


Рисунок 3.22 - Аппроксимация (а) и интерполяция (б) кривых

Задачи аппроксимации (приближенного представления) возникают при замене кривой или поверхности, описываемых сложными функциями, другими функциями, описываемыми более простыми уравнениями, без потери необходимой точности.

Задачи интерполяции (приближенного восстановления) связаны с поиском гладких кривых или поверхностей, проходящих через множество заданных точек.

Задачи сглаживания возникают, когда необходимо, чтобы искомая кривая или поверхность описывались функцией, обеспечивающей, например, необходимую степень дифференцирования.

Таким образом, при моделировании кривых линий и поверхностей решается следующая задача: по заданному массиву точек на плоскости или в пространстве строится кривая, проходящая через все эти точки (задача интерполяции) либо проходящая вблизи от этих точек (задача сглаживания).

Обратимся для определенности к задаче интерполяции и начнем рассмотрение с обсуждения правил выбора класса кривых.

Полином Лагранжа

Пусть на плоскости задан набор точек (x_i, y_i) , $i = 0, 1, \dots, n$, таких, что $x_0 < x_1 < \dots < x_{n-1} < x_n$ (рисунок 3.23).

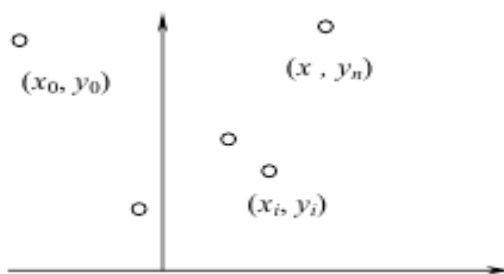


Рисунок 3.23 Набор точек

То обстоятельство, что точки заданного набора пронумерованы в порядке возрастания их абсцисс, позволяет искать кривую в классе графиков функций. Можно описать основные проблемы сглаживания этого дискретного набора, ограничившись случаем многочлена.

Из курса математического анализа известно, что существует интерполяционный многочлен Лагранжа

$$L_n(x) = \sum_{i=0}^n y_i \frac{\omega_n(x)}{(x - x_i) \omega_n'(x_i)},$$

где $\omega_n(x) = \prod_{j=0}^n (x - x_j)$ график которого проходит через все заданные точки $(x_i, y_i), i = 0, 1, \dots, n$.

Это обстоятельство и простота описания (заметим, что многочлен однозначно определяется набором своих коэффициентов; в данном случае их число совпадает с количеством точек в заданном наборе) являются несомненными преимуществами построенного интерполяционного многочлена.

Однако данный подход имеет и некоторые недостатки.

1. Степень многочлена Лагранжа на единицу меньше числа заданных точек. Поэтому чем больше точек задано, тем выше степень такого многочлена. И, хотя график интерполяционного многочлена Лагранжа всегда будет проходить через все точки массива, его уклонение (от ожидаемого) может оказаться довольно значительным (рисунок 3.24).

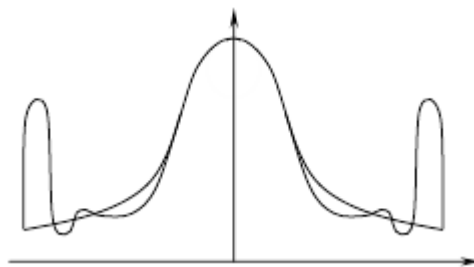


Рисунок 3.24 - Интерполяция полиномом Лагранжа

Кроме этого повышение степени многочлена приводит к нежелательным осцилляциям.

2. Изменение одной точки (ситуация, довольно часто встречающаяся на практике) требует полного пересчета коэффициентов интерполяционного многочлена и к тому же может существенно повлиять на вид задаваемой кривой.

Приближающую кривую можно построить и совсем просто: если последовательно соединить точки заданного набора прямолинейными отрезками, то в результате получится ломаная (рисунок 3.25).

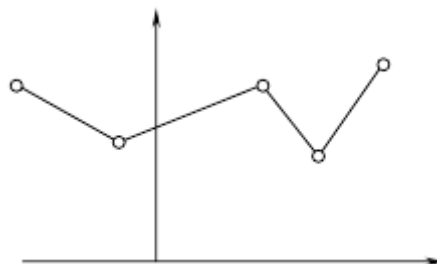


Рисунок 3.25 - Кусочно-линейная интерполяция

При такой кусочно-линейной интерполяции требуется найти всего $2n$ чисел (каждый прямолинейный отрезок определяется ровно двумя коэффициентами), но построенная таким образом аппроксимирующая кусочно-линейная функция не обладает нужной гладкостью: уже первая производная этой функции терпит разрывы в узлах интерполяции.

Рассмотрев эти две крайние ситуации, найдем класс функций, которые в основном сохранили бы перечисленные выше преимущества обоих подходов и одновременно были бы в известной степени свободны от их недостатков.

Для этого поступим так: будем использовать многочлены (как и в первом случае) и строить их последовательно, звено за звеном (как во втором случае). В результате получится, так называемый, полиномиальный многозвенник. При подобном подходе важно правильно выбрать степени привлекаемых многочленов, а для плавного изменения результирующей кривой необходимо еще тщательно подобрать коэффициенты многочленов (из условий гладкого сопряжения соседних звеньев). То, что получится в результате описанной схемы построения, называют сплайн-функциями или сплайнами.

Форма математического сплайна повторяет контур физического сплайна - гибкой металлической линейки, которыми издревле пользовались чертёжники и слесари-лекальщики (рисунок 3.26).

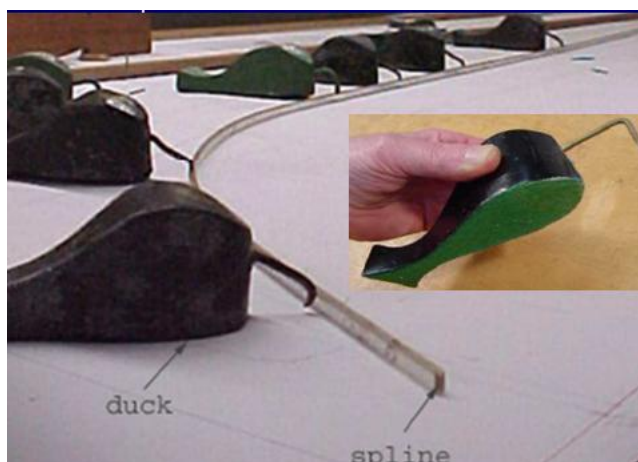


Рисунок 3.26 - Физический сплайн

Для изменения формы линейки используются точечные закрепления и концевые заделки. Меняя количество и расположение этих произвольно расположенных базовых точек, кривую можно выгнуть произвольным образом. Кривая всегда остаётся гладкой (не имеет разрывов и переломов) (рисунок 3.27).

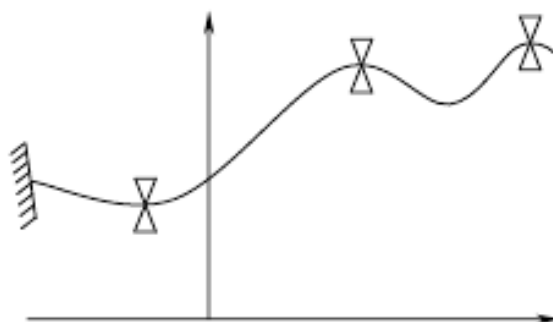


Рисунок 3.27 Сплайн

Задачи аппроксимации и интерполяции (интерполяция является частным случаем аппроксимации) при конструировании кривых линий и поверхностей могут быть решены с использованием различных вычислительных методов. При этом сегменты пространственных кривых аппроксимируются отрезками, дугами окружности, параболами и кривыми более высоких порядков.

Методы конструирования кривых линий и поверхностей: метод Эрмита, метод Безье, метод аппроксимации с помощью кубических В-сплайнов, метод Кунса и др. используют

одинаковый подход: разбиение пространственной кривой на отдельные сегменты - элементарные куски. Описание этих сегментов полигонами третьей степени, определенные условия соединения элементарных кусков в составные кривые. Такие кривые в литературе обобщенно называют *сплайновыми кривыми*.

Сравнение форм Эрмита, Безье и В-сплайнов

Каждое из этих представлений оказывается полезным в разных ситуациях. Форма Эрмита пригодна для аппроксимации уже имеющихся поверхностей, когда необходимо добиться как соответствия точек, так и соответствия касательных векторов, в то время как представление в виде В-сплайнов удобно для аппроксимации точек.

Формы Безье и В-сплайнов пригодны для работы в интерактивном режиме, так как их геометрические векторы состоят из одних только точек. Обе эти формы обладают свойством выпуклой оболочки, которое оказывается полезным при изображении кривых. Отметим, что кривую, первоначально заданную в одной форме, можно преобразовать в другую форму.

На рисунке 3.28 и рисунке 3.29 показано различие в поведении кривых, построенных разными методами на основе одного и того же точечного базиса.

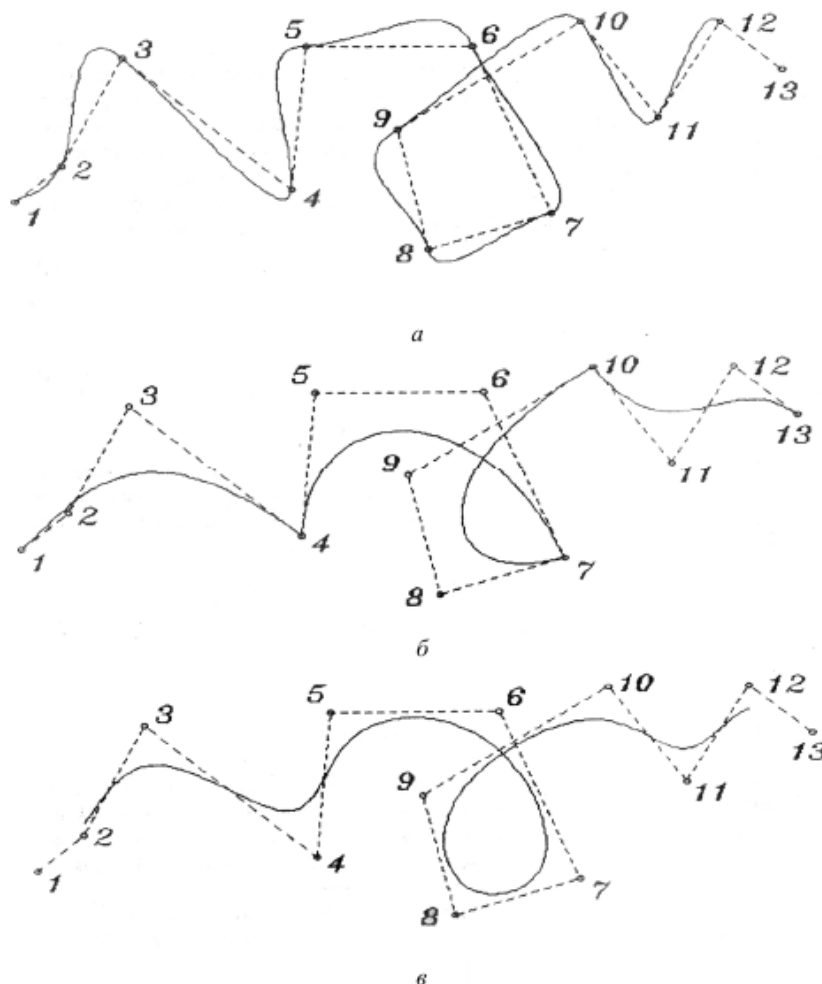


Рисунок 3.28 - Три формы построения кривых: а – форма Эрмита;
б – форма Безье; в – форма В-сплайна

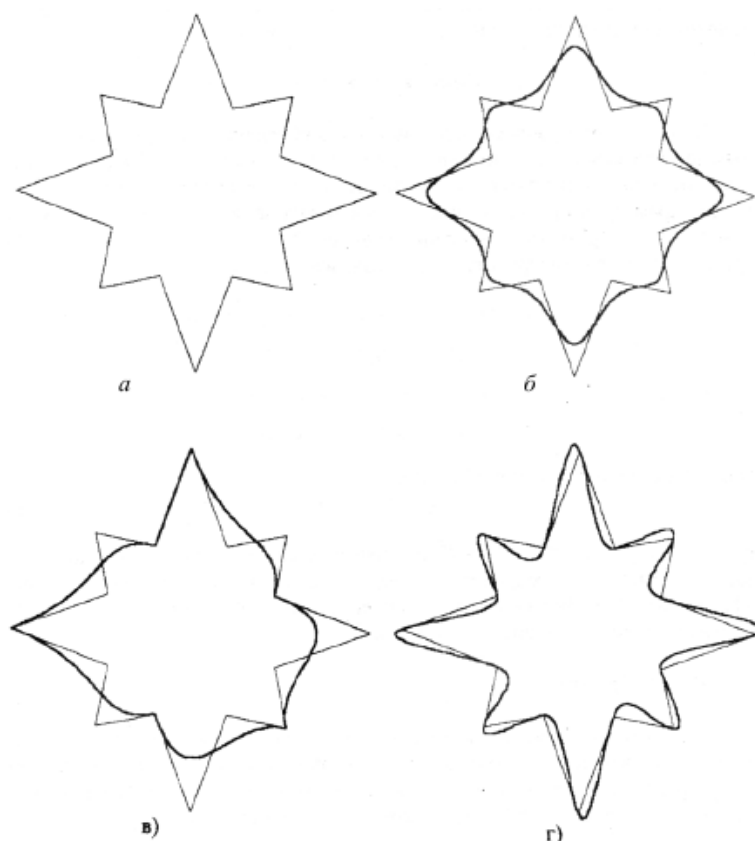


Рисунок 3.29 - Сглаживание фигуры: а – исходная фигура; б – форма В-сплайна; в – форма Безье; г – форма Эрмита

Одними из самых популярных сплайнов, применяемых в компьютерном моделировании, являются "кривые Безье" и NURBS-кривые (Non-Uniform Rational B-Spline - неоднородный рациональный фундаментальный сплайн). Эти сплайны обладают высокой степенью гладкости, легко вычисляются и управляются.

Кривая (сплайн) Безье задаётся вершинами многоугольника и в общем случае не проходит через базовые точки (рисунок 3.30).

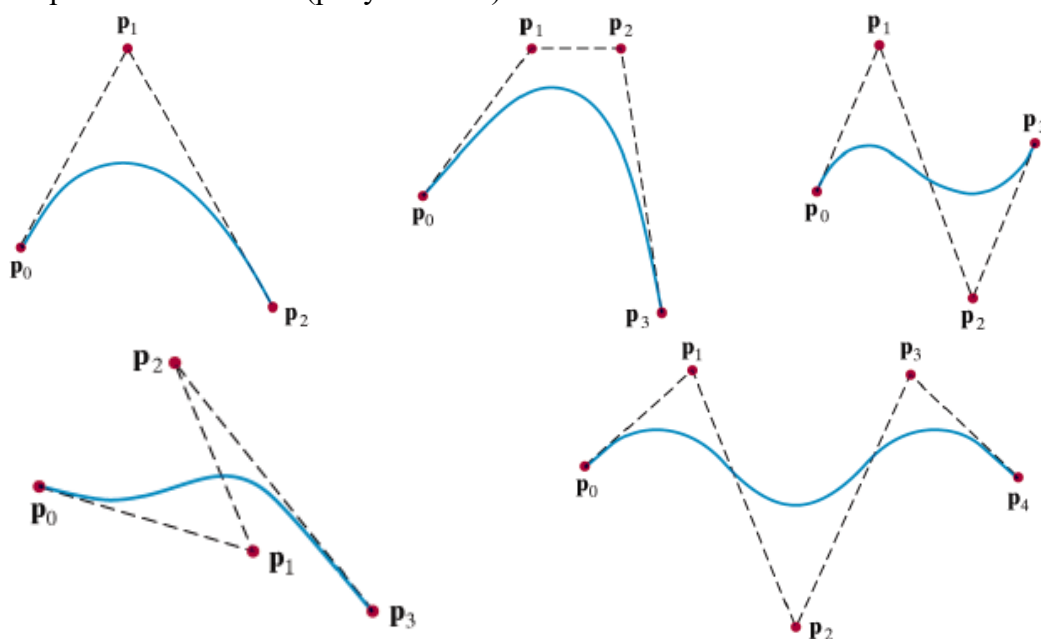


Рисунок 3.30 - Сплайны Безье

Для получения кривой, проходящей через заданные точки применяются составные кривые Безье, которые также можно использовать для задач интерполяции.

Неоднородный рациональный В-сплайн или NURBS подобен обычному кубическому В-сплайну.

NURBS-кривые не проходят через базовые точки, а усредняют - аппроксимируют их, причём узловые точки могут быть расставлены неравномерно. Такие линии называются неоднородными, а каждый такой узел, имеющий свой вес (коэффициент, определяющий влияние данной точки на кривизну линии), - рациональным. Перемещая узловые точки, можно управлять формой кривой (рисунок 3.31).

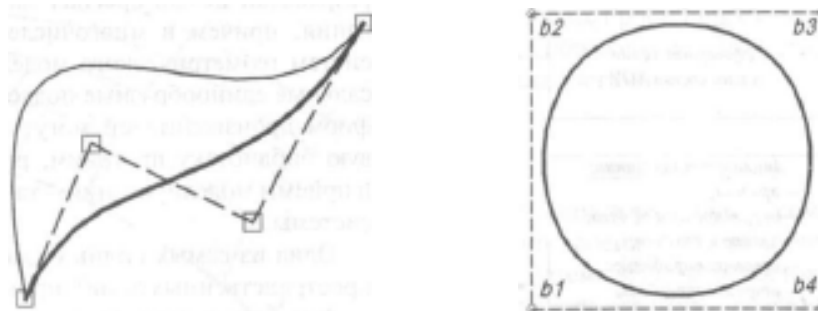


Рисунок 3.31 - Интерактивное управление формой гладких NURBS-кривых

NURBS позволяет точно воспроизвести все конические сечения - окружность (см. рисунок 3.31 справа), эллипс, параболу и гиперболу. В NURBS - представлении можно работать не только с коническими сечениями, но с кривыми Безье, рациональными кривыми Безье и В-сплайнами. Преобразование всех этих кривых к NURBS может значительно сократить объем программирования.

Поверхности Безье используются при интерактивном проектировании по тем же причинам, что и кривые Безье: управляющие точки позволяют легко изменять форму куска поверхности. Поверхности Безье так же, как и кривые Безье, обладают свойством выпуклой оболочки. Геометрическая матрица состоит из 16 управляющих точек (рисунок 3.32).

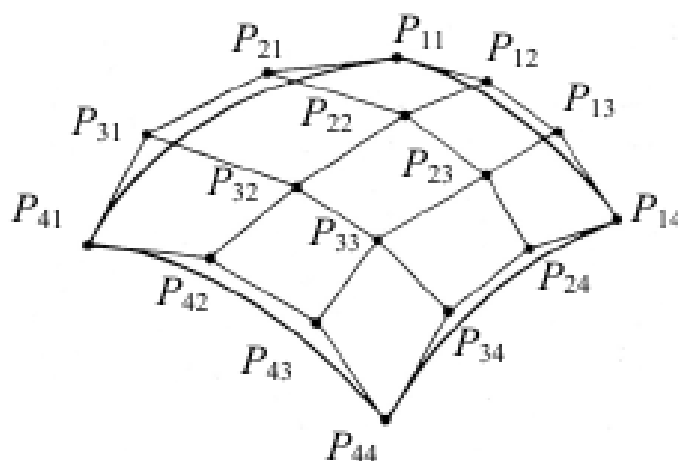


Рисунок 3.32 - Шестнадцать управляющих точек бикубического куска Безье

Для достижения непрерывности в поперечном направлении относительно ребер кусков необходимо равенство четырех управляющих точек, принадлежащих общим ребрам соседних кусков. Для непрерывности касательного вектора требуется, чтобы две четверки

управляющих точек по обеим сторонам общего ребра были коллинеарны другим ребрам. Коллинеарными будут следующие совокупности управляющих точек: (P_{13}, P_{14}, P_{15}) , (P_{23}, P_{24}, P_{25}) , (P_{33}, P_{34}, P_{35}) , (P_{43}, P_{44}, P_{45}) . Отношения длин коллинеарных отрезков должны быть постоянными (рисунок 3.33).

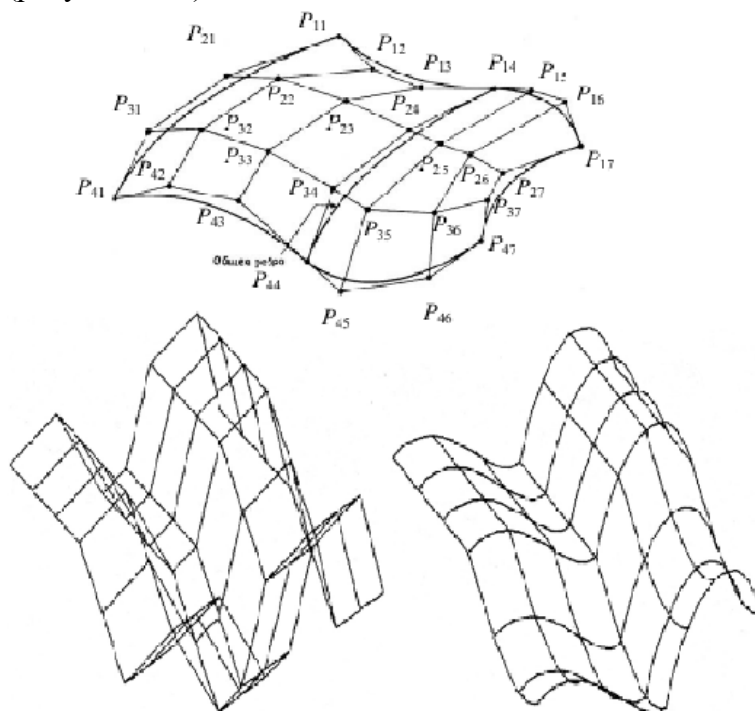


Рисунок 3.33 - Составные поверхности Безье

На рисунках 3.34 и 3.35 приведены примеры В-сплайновых поверхностей.

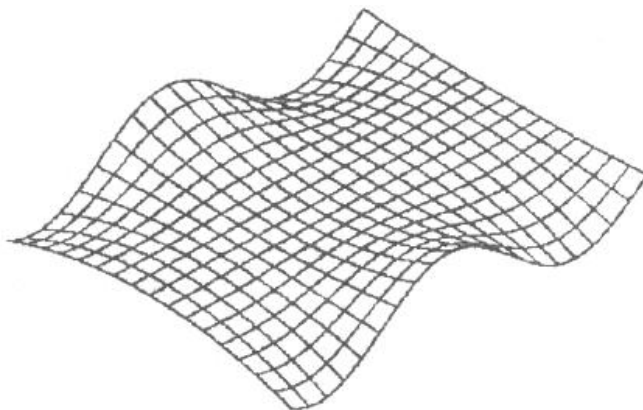


Рисунок 3.34 - Пример В-сплайновой поверхности

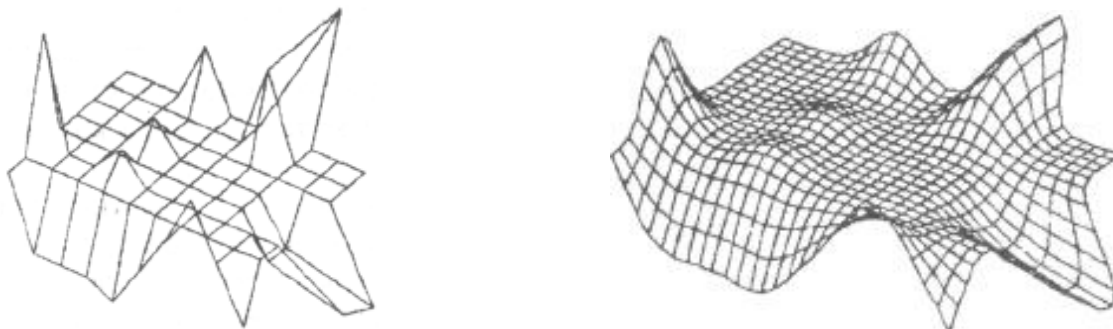


Рисунок 3.35 - Каркас и В-сплайновая поверхность

На рисунке 3.36 показана NURBS-поверхность третьего порядка по каждому из двух направлений. Характеристический многогранник изображен тонкими линиями.

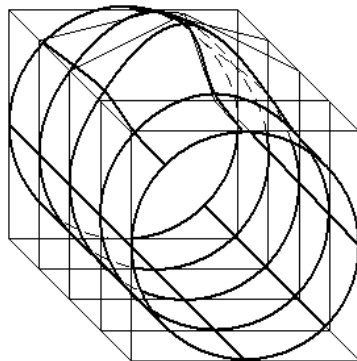


Рисунок 3.36 - NURBS-поверхность

Вопросы для самоконтроля

1. Перечислите основные группы функций (5) моделирования твёрдого тела.
2. В чём суть функций создания примитивов и булевых операторов?
3. В чём суть функций заметания, качания (вращения) и скининга?
4. В чём суть функций скругления (плавного сопряжения) и поднятия?
5. В чём суть функций моделирования границ?
6. В чём суть функций объектно-ориентированного моделирования?
7. Какие линии называют кривыми?
8. Перечислите кривые, получаемые сечением плоскостью поверхности конуса. Условия их образования.
9. Как формируется цилиндрическая винтовая линия?
10. В чём суть плазового - шаблонного метода моделирования поверхностей?
11. В чём состоят методы интерполяции и аппроксимации кривых? Чем они отличаются?
12. Приведите пример интерполяции (полином Лагранжа). В чём его недостатки?
13. Объясните суть аппроксимации (интерполяции) массива точек сплайнами.
14. Перечислите методы конструирования кривых линий и поверхностей. Какой подход при этом используют?
15. Сравните формы построения кривых (Эрмита, Безье, В-сплайна). Приведите графические примеры.
16. В чём преимущества и суть построения сплайнов и поверхностей Безье (в том числе составных)?
17. В чём отличие NURBS от В-сплайнов? Перечислите преимущества NURBS.

Лекция 4

Классификация поверхностей. Развёртки

Поверхность представляет собой множество последовательных положений линии, перемещающейся в пространстве. Эту линию называют образующей поверхности. Она может быть прямой или кривой. Закон перемещения образующей задается линиями - направляющими.

Многогранники

Поверхность, образованная частями попарно пересекающихся плоскостей, называется многогранной. Элементами многогранных поверхностей являются грани, ребра и вершины. Отсеки плоскостей, образующие многогранную поверхность, называются гранями, линии пересечения смежных граней - ребрами, точки пересечения не менее чем трех граней - вершинами (рисунок 4.1).

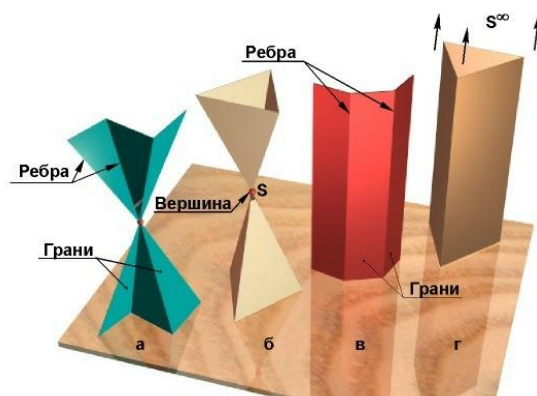


Рисунок 4.1 - Элементы многогранных поверхностей

Если каждое ребро многогранной поверхности принадлежит одновременно двум её граням, то её называют замкнутой (рисунок 4.1б, г), в противном случае - незамкнутой (рисунок 4.1а, в). Многогранная поверхность называется пирамидальной, если все её ребра пересекаются в одной точке - вершине (см. рисунок 4.1а). Пирамидальная поверхность имеет две неограниченные полы. Многогранная поверхность называется призматической, если все её ребра параллельны между собой (см. рисунок 4.1г).

Геометрическое тело, со всех сторон ограниченное плоскими многоугольниками, называется многогранником (рисунок 4.2). Простейшими многогранниками являются пирамиды и призмы.

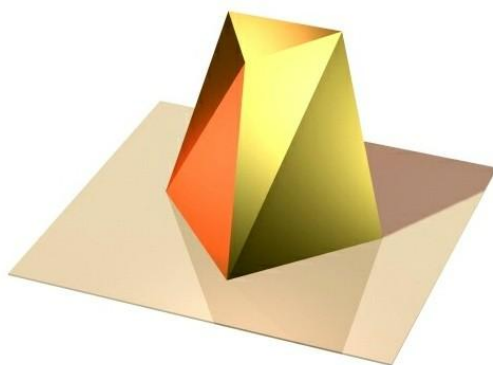


Рисунок 4.2 - Многогранник

Среди других видов многогранников следует выделить - призматойды и правильные многогранники (тела Платона). Призматойдом называется многогранник, у которого верхнее и нижнее основания - многоугольники, расположенные в параллельных плоскостях, а боковые грани представляют собой треугольники или трапеции.

Существует пять правильных многогранников:

-**тетраэдр** (четырёхгранник) - ограничен четырьмя равносторонними и равными треугольниками;

-**гексаэдр** (четырёхгранник, или куб) - ограничен шестью равными квадратами;

-**октаэдр** (восьмигранник) - ограничен восемью равносторонними и равными треугольниками;

-**додекаэдр** (двенадцатигранник) - ограничен двенадцатью равносторонними и равными пятиугольниками;

-**икосаэдр** (двадцатигранник) - ограничен двадцатью равносторонними и равными треугольниками.

Вокруг всех правильных многогранников можно описать сферу. Совокупность всех ребер и вершин многогранника называется его сеткой.

Кривые поверхности (рисунок 4.3) широко применяются в различных областях науки и техники при создании очертаний различных технических форм или как объекты инженерных исследований.

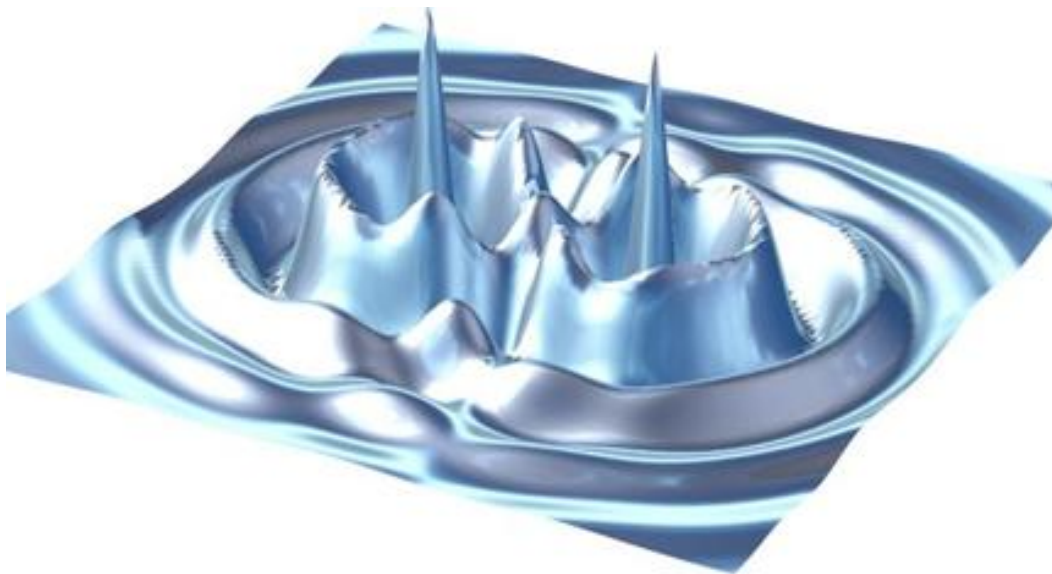


Рисунок 4.3 - Пример сложной поверхности

Существуют три способа задания кривых поверхностей:

-**аналитический** - при помощи уравнений;

-**при помощи каркаса**;

-**кинематический** - перемещением линий в пространстве.

Составлением уравнений поверхностей занимается аналитическая геометрия; она рассматривает кривую поверхность как множество точек, координаты которых удовлетворяют некоторому уравнению или системе алгебраических уравнений.

При каркасном способе задания кривая поверхность задается совокупностью некоторого количества линий, принадлежащих поверхности. В качестве линий, образующих каркас, как правило, берут семейство линий, получающихся при пересечении поверхности рядом параллельных плоскостей. Этот способ применяется при проектировании кузовов автомобилей, в самолето- и судостроении, в топографии и т.п.

При кинематическом способе образования и задания кривых поверхностей каждая кривая поверхность рассматривается как совокупность последовательных положений образующей линии, перемещающейся в пространстве по определенному закону. Образующая линия при своем движении может оставаться неизменной, а может и менять свою форму. Закон перемещения образующей линии, как правило, задается при помощи направляющих линий и алгоритма перемещения образующей по направляющим.

Определителем поверхности называют совокупность условий, необходимых и достаточных для задания поверхности в пространстве. Определитель поверхности выявляется путем анализа способов образования поверхности или её основных свойств. В общем случае поверхность может быть образована несколькими способами и поэтому может иметь несколько определителей. Обычно из всех способов образования поверхности выбирают простейший. Определитель поверхности состоит из двух частей:

-*геометрической части* - совокупности геометрических фигур, с помощью которых можно образовать поверхность;

-*алгоритмической части* - алгоритма формирования поверхности при помощи фигур, входящих в геометрическую часть определителя.

Рассмотрим примеры выявления определителя для некоторых простейших поверхностей (рисунок 4.4):

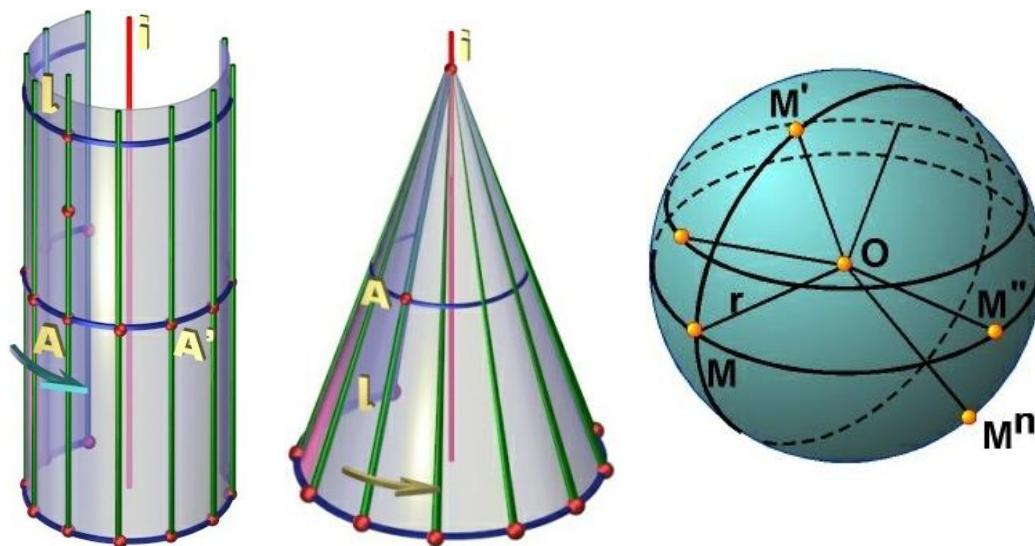


Рисунок 4.4 - Определители цилиндра, конуса и сферы

1. **Цилиндрическая поверхность вращения** может быть образована вращением прямой $l \parallel i$ вокруг оси i . Геометрическая часть определителя поверхности состоит из образующей l и оси i . Алгоритмическая часть определителя состоит из операции вращения образующей линии l вокруг оси i .

2. **Коническая поверхность вращения** может быть образована вращением прямой l , пересекающей ось вращения i под некоторым углом. Алгоритмическая часть определителя состоит из словесного указания о том, что поверхность образуется вращением образующей l вокруг оси i .

3. **Сферой** называется поверхность, образованная множеством точек пространства, находящихся на расстоянии $|r|$ от данной точки O . Геометрическая часть определителя сферы состоит из точки O (центра сферы) и точки M , принадлежащей её поверхности. Алгоритм построения любой точки сферы заключается в проведении через точку O произвольной прямой и откладывания на ней от точки O отрезка $|OM'| = |OM| = |r|$.

Кривые поверхности разделяются на линейчатые и нелинейчатые, закономерные и незакономерные. Поверхность называется линейчатой, если она может быть образована перемещением прямой линии, в противном случае она называется нелинейчатой. Если поверхность может быть задана каким-либо уравнением, она называется закономерной, в противном случае - незакономерной, или графической (задается только чертежом). Закономерные поверхности, в зависимости от вида уравнения, разделяются на алгебраические и трансцендентные. Алгебраическое уравнение n -й степени (в декартовых координатах) задает алгебраическую поверхность n -го порядка (трансцендентные поверхности порядка не имеют). Алгебраическая поверхность n -го порядка пересекается плоскостью по кривой n -го порядка, а с прямой линией - в n точках. Плоскость, имеющую уравнение первой степени (с произвольной плоскостью пересекается по прямой линии, а с прямой - в одной точке), можно рассматривать как поверхность первого порядка.

Примерами кривых поверхностей второго порядка могут служить поверхности, образованные вращением кривых второго порядка вокруг одной из своих осей. Поверхности второго порядка пересекаются с произвольной плоскостью по кривым второго порядка, а с прямой - в двух точках. Примером поверхности четвертого порядка может служить тор.

Наибольшее применение в технике получили кинематические кривые поверхности с образующими постоянной формы:

- линейчатые поверхности: развертывающиеся, неразвертывающиеся, винтовые.
- поверхности вращения.

Поверхность называется линейчатой, если она может быть образована перемещением прямой линии. Поверхность, которая не может быть образована движением прямой линии, называется нелинейчатой. Например, конус вращения - линейчатая поверхность, а сфера - нелинейчатая. Через любую точку линейчатой поверхности можно провести, по крайней мере, одну прямую, целиком принадлежащую поверхности. Множество таких прямых представляет собой непрерывный каркас линейчатой поверхности. Все нелинейчатые поверхности являются неразвертывающимися.

Поверхность называется развертывающейся, если она путем изгибания может быть совмещена с плоскостью без образования складок и разрывов. Очевидно, что все многогранные поверхности являются развертывающимися. Из кривых поверхностей этим свойством обладают только те линейчатые поверхности, которые имеют ребро возврата.

Существует только три вида линейчатых поверхностей, имеющих ребро возврата: торсы, конические и цилиндрические (рисунок 4.5).

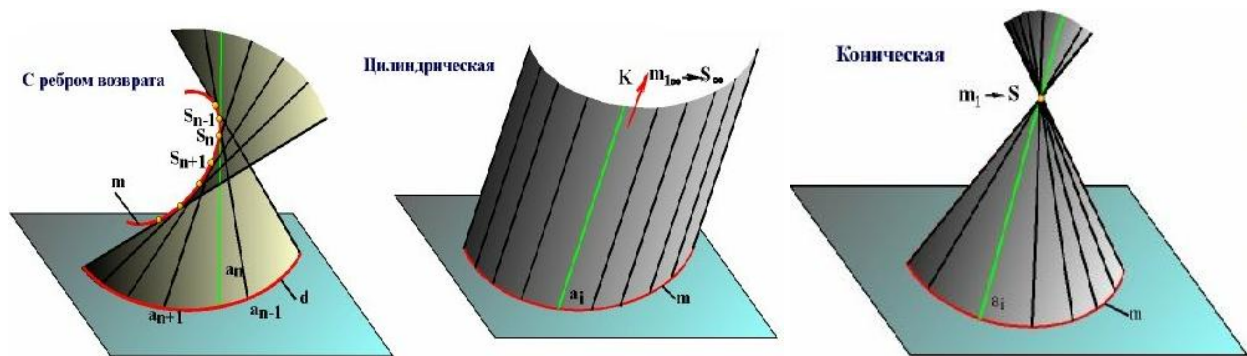


Рисунок 4.5 - Линейчатые поверхности с ребром возврата

Торсы

Возьмем некоторую пространственную ломаную линию 1, 2, 3, 4, 5, 6, ... и продолжим её стороны так, как показано на рисунке 4.6. В результате получим двухполую многогранную развертываемую поверхность. При неограниченном увеличении числа сторон ломаной линии с одновременным стремлением длины каждой из них к нулю и переходе к пределу получим:

- пространственная ломаная линия 1, 2, 3, 4, 5, 6, ... преобразуется в пространственную кривую линию m ;
- ребра многогранной поверхности преобразуются в касательные к пространственной кривой m ;
- многогранная поверхность преобразуется в линейчатую двухполую развертывающуюся кривую поверхность, которая называется торсом.

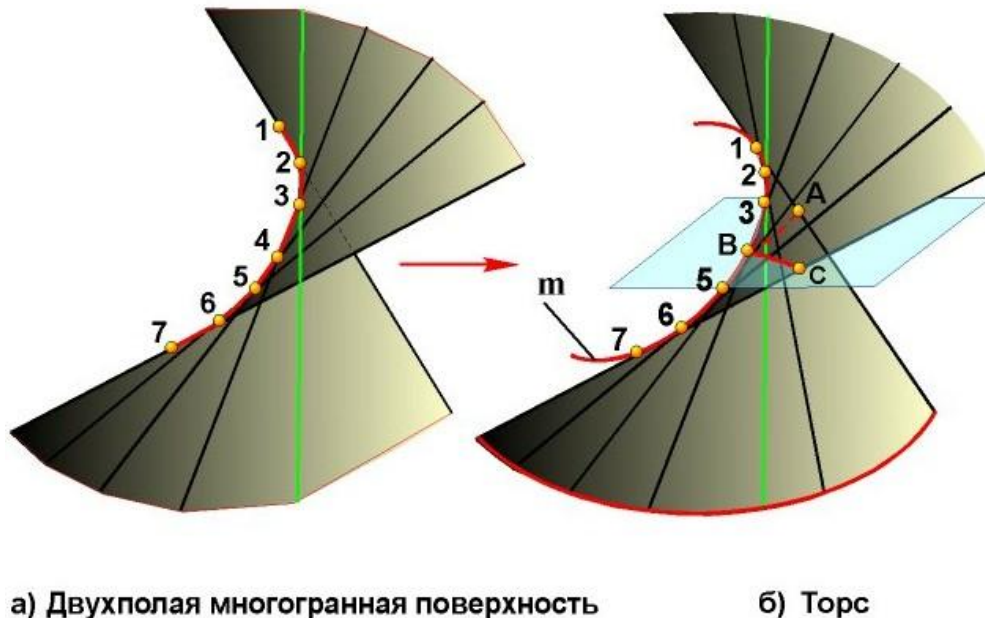


Рисунок 4.6 - Формирование торса

Множество всех касательных прямых к пространственной кривой представляет собой непрерывный каркас поверхности торса. Через каждую точку поверхности проходит одна касательная к кривой m . Таким образом, торс представляет собой поверхность, которая образуется непрерывным движением прямолинейной образующей, касающейся во всех своих положениях некоторой пространственной кривой линии. Направляющая пространственная

кривая m служит границей между двумя полостями поверхности торса и называется ребром возврата. Если взять на кривой m какую-либо точку B и провести через неё плоскость Σ , пересекающую обе полости поверхности, то полученная в пересечении кривая ABC будет иметь так называемую точку возврата B . Следовательно, ребро возврата является множеством точек возврата кривых линий, полученных при пересечении данной поверхности различными плоскостями. Этим и объясняется её название. Если ребром возврата является цилиндрическая винтовая линия, то такая поверхность называется развертывающимся геликоидом.

Свойством развёртываемости торс обладает потому, что он является пределом некоторой развертывающейся многогранной поверхности. Геометрическая часть определителя торса состоит из ребра возврата. Алгоритмическая часть определителя торса состоит из указания о том, что образующая прямая при своём движении остается касательной к ребру возврата. Если ребро возврата выродится в собственную точку пространства, то образующие торса, проходя через неё, образуют коническую поверхность произвольного вида. Если эта точка (вырожденное ребро возврата) будет несобственной точкой пространства, то образующие торса, проходя через неё, окажутся параллельными между собой и образуют цилиндрическую поверхность общего вида. Таким образом, цилиндрическая и коническая поверхности обладают свойством развёртываемости, так как являются частными случаями поверхности торса. Однако, чтобы задать коническую или цилиндрическую поверхности, недостаточно иметь только ребро возврата (собственную или несобственную точку) - положение образующей прямой не определяется одной точкой. Необходимо задать ещё направляющую линию.

Цилиндрическая поверхность образуется движением прямой линии, скользящей по некоторой неподвижной замкнутой или незамкнутой кривой и остающейся параллельной своему исходному положению (рисунок 4.7).

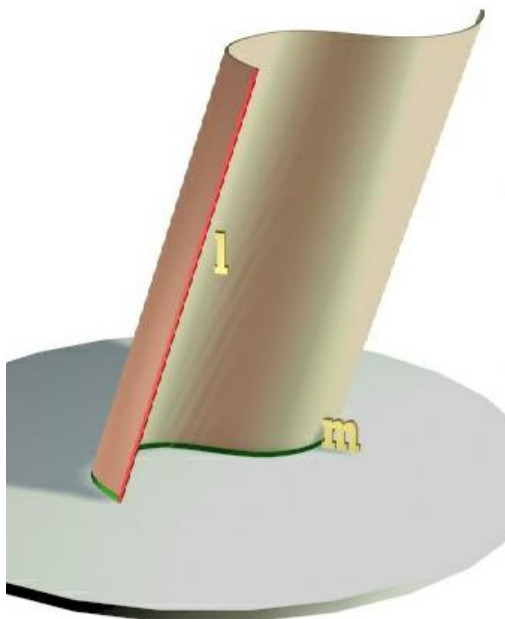


Рисунок 4.7 - Формирование цилиндрической поверхности

Множество прямолинейных образующих представляет собой непрерывный каркас цилиндрической поверхности. Через каждую точку поверхности проходит одна прямолинейная образующая (см. рисунок 4.5). Неподвижная кривая m , по которой скользит образующая l , называется направляющей. Если направляющая линия является кривой второго порядка, то и цилиндрическая поверхность будет второго порядка. Геометрическая часть определителя цилиндрической поверхности состоит из направляющей линии m и исходного положения образующей l . Алгоритмическая часть определителя состоит из указания о том, что любая образующая поверхности может быть построена как прямая, пересекающая кривую m и параллельная прямой l . Цилиндрическая поверхность является бесконечной в направлении своих образующих. Часть замкнутой цилиндрической поверхности, заключенная между двумя плоскими параллельными сечениями, называется цилиндром, а фигуры сечения - его основаниями. Сечение цилиндрической поверхности плоскостью, перпендикулярной её образующим, называется нормальным. В зависимости от формы нормального сечения цилиндры бывают (рисунок 4.8):

- круговые - нормальное сечение - окружность;
- эллиптические - нормальное сечение - эллипс;
- параболические - нормальное сечение - парабола;
- гиперболические - нормальное сечение - гипербола;
- общего вида - нормальное сечение - кривая случайного вида.

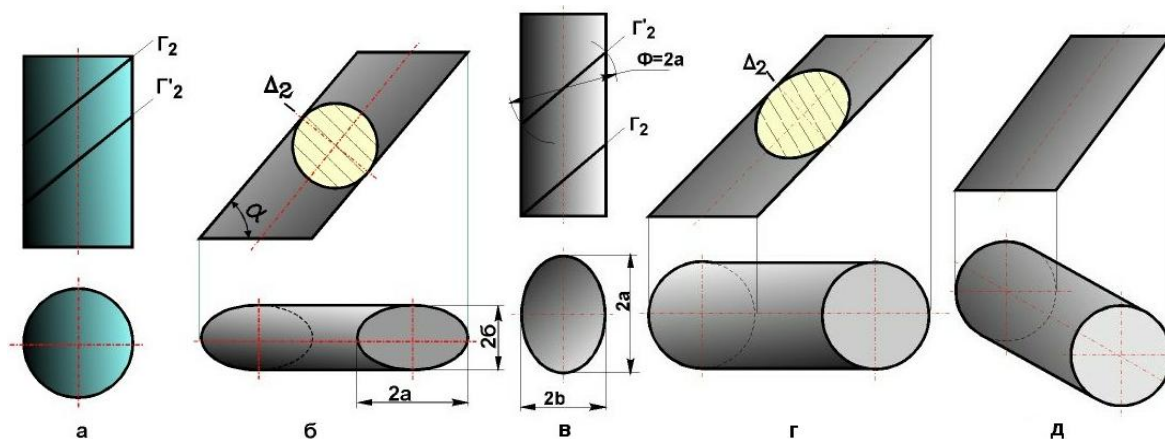


Рисунок 4.8 - Разновидности цилиндра

Если за основание цилиндра принимается его нормальное сечение (окружность), цилиндр называют прямым (рисунок 4.8а). Если за основание цилиндра принимается одно из косых сечений, цилиндр называют наклонным (рисунок 4.8б, г, д). Наклонные сечения прямого кругового цилиндра являются эллипсами (см. рисунок 4.8а). Наклонные сечения прямого эллиптического цилиндра в общем случае - эллипсы (см. рисунок 4.8б). Однако его всегда можно пересечь плоскостью, наклонной к его образующим, таким образом, что в сечении получится окружность.

Коническая поверхность образуется движением прямой линии, скользящей по некоторой неподвижной замкнутой или незамкнутой кривой и проходящей во всех своих положениях через неподвижную точку (рисунок 4.9).

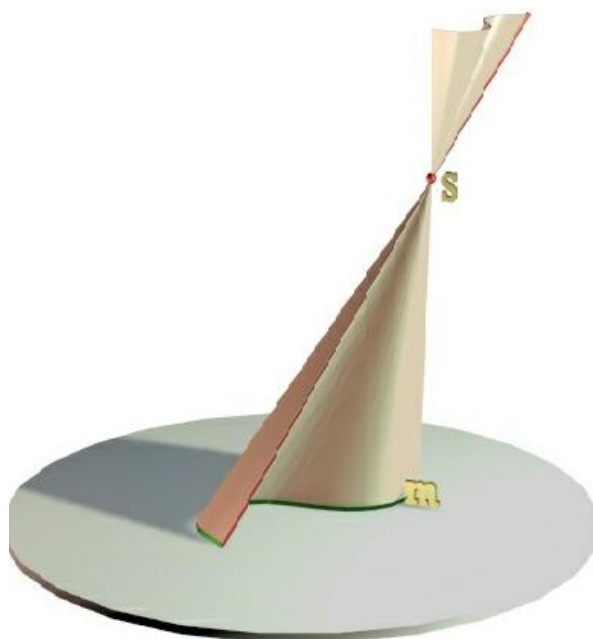


Рисунок 4.9 - Формирование конической поверхности

Неподвижная кривая m , по которой скользит образующая l , называется направляющей (см. рисунок 4.5). Если направляющая линия является кривой второго порядка, то и коническая поверхность будет второго порядка. неподвижная точка S , делящая поверхность на две бесконечные половины, называется вершиной. Множество прямолинейных образующих представляет собой непрерывный каркас конической поверхности. Через каждую точку поверхности проходит одна прямолинейная образующая (исключением является только вершина S , которая называется "особой точкой поверхности"). Геометрическая часть определителя конической поверхности состоит из направляющей кривой m и вершины S . Алгоритмическая часть определителя состоит из указания о том, что любая образующая поверхности может быть построена как прямая, проходящая через вершину S и пересекающая кривую m . Часть замкнутой конической поверхности, ограниченная вершиной и какой-либо плоскостью, пересекающей все её образующие, называется конусом. Фигура сечения конической поверхности этой плоскостью называется основанием конуса. Сечение конической поверхности плоскостью, перпендикулярной её оси, называется нормальным. Осью конической поверхности называется линия пересечения её плоскостей симметрии. Следовательно, не все конические поверхности имеют ось, а только те, которые имеют не меньше двух плоскостей симметрии. Конические поверхности, не имеющие оси (а следовательно, и нормального сечения), называются коническими поверхностями общего вида.

Конические поверхности, имеющие ось, в зависимости от вида нормального сечения бывают (рисунок 4.10):

- круговые - нормальное сечение - окружность;
- эллиптические - нормальное сечение - эллипс;
- другие.

Если за основание конуса принимается фигура его нормального сечения, конус называют прямым, если иное сечение - наклонным. Прямой круговой конус изображен на рисунке 4.10а,

наклонный круговой конус - на рисунке 4.10б. Основанием такого конуса может быть только эллипс, ось его не проходит через центр основания. Прямой эллиптический конус показан на рисунке 4.10в. Эллиптический конус (так же как и эллиптический цилиндр) имеет две системы круговых сечений. Если принять одну из них за основание конуса, получим наклонный эллиптический конус с круговым основанием (рисунок 4.10г). Ось наклонного конуса не проходит через центр основания.

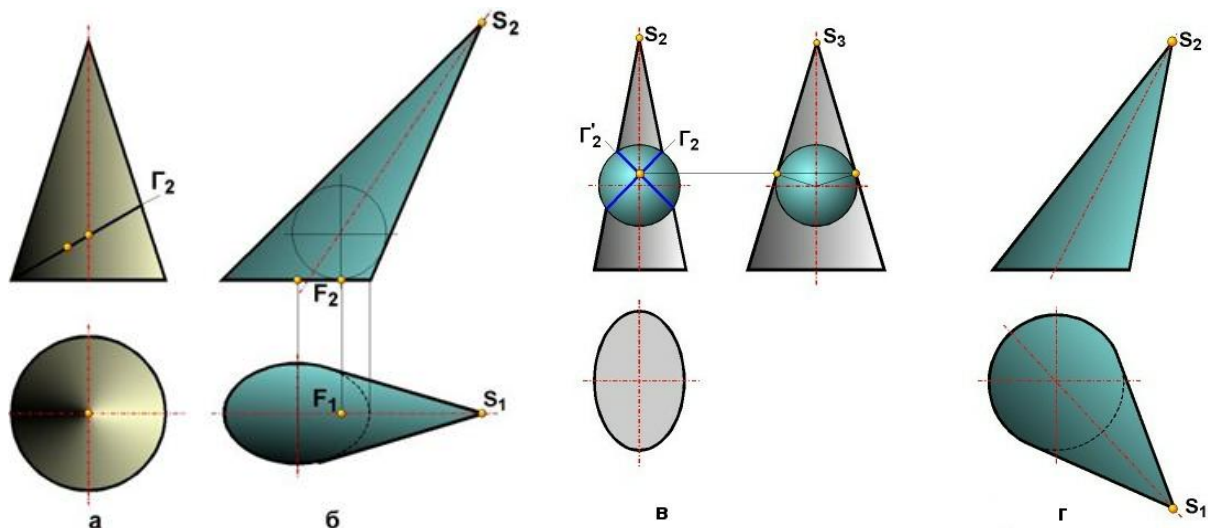


Рисунок 4.10 - Разновидности конуса

Заметим, что у всех развертывающихся линейчатых поверхностей две смежные образующие либо пересекаются (торс, коническая поверхность), либо параллельны (цилиндрическая поверхность).

Неразвёртывающиеся (косые) линейчатые поверхности в общем случае образуются движением прямолинейной образующей по трем направляющим линиям, которые однозначно задают закон её перемещения (рисунок 4.11). Множество возможных положений прямой, которая в каждый момент времени пересекает три скрещивающиеся прямые образует однополостный гиперболоид.

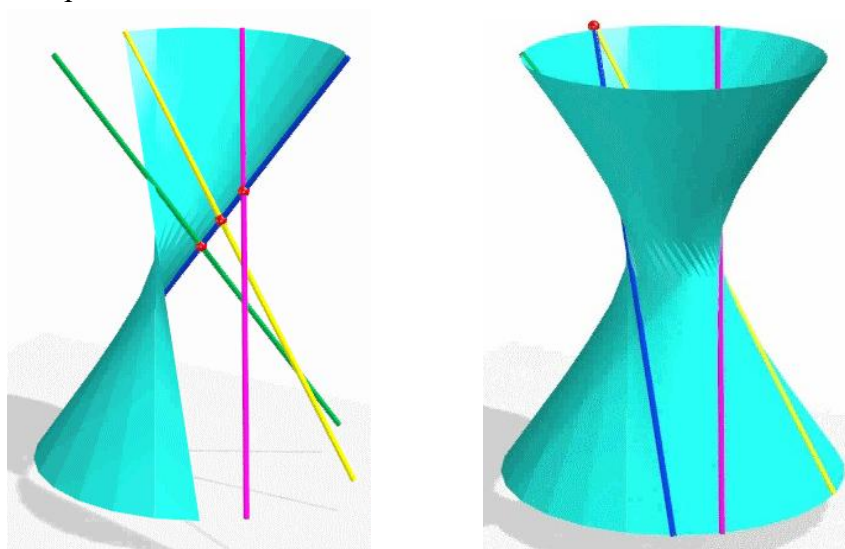


Рисунок 4.11 - Однополостный гиперболоид

Направляющие линии могут быть прямыми и кривыми. В общем случае линейчатая поверхность может быть построена из множества образующих прямых, пересекающих три заданные пространственные кривые (рисунок 4.12).

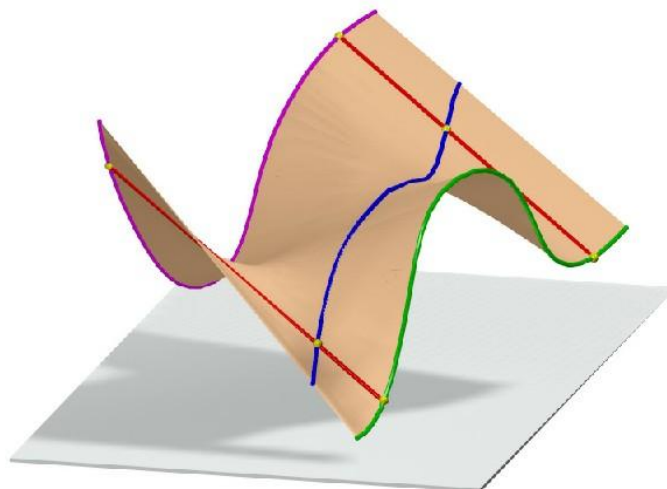


Рисунок 4.12 - Формирование кривой поверхности

Разновидностями косых поверхностей являются линейчатые поверхности с направляющей плоскостью и частные их виды - линейчатые поверхности с плоскостью параллелизма (поверхности Каталана). В первом случае поверхность однозначно задается двумя направляющими линиями и направляющей плоскостью, которая заменяет третью направляющую линию. Образующая прямая скользит по двум направляющим и сохраняет постоянный угол α с некоторой плоскостью Σ , которая называется направляющей. В частном случае, если угол α равен нулю, образующая прямая будет параллельна направляющей плоскости, которая в этом случае называется плоскостью параллелизма.

Поверхности с направляющей плоскостью ($\alpha \neq 0$) называются косыми цилиндрами, если обе направляющие являются кривыми линиями; косыми коноидами - если одна из направляющих - прямая линия; дважды косой плоскостью, если направляющие - скрещивающиеся прямые.

На рисунке 4.13 показан дважды косой цилиндрический, как линейчатая поверхность с тремя направляющими, из которых две - пространственные кривые и одна прямая.

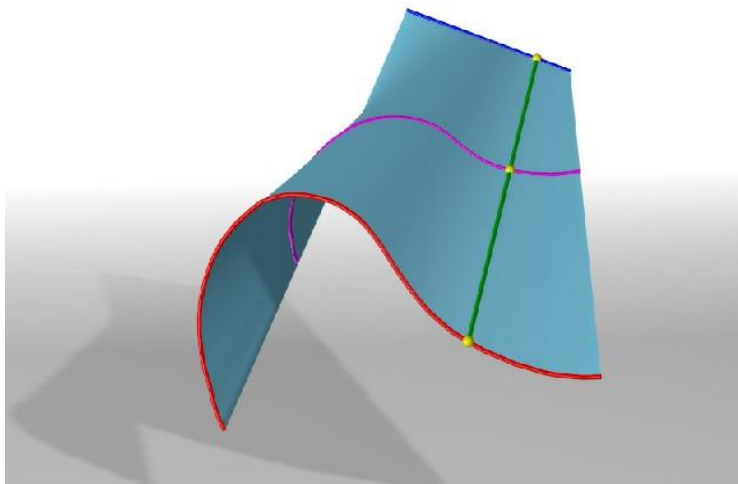


Рисунок 4.13 - Дважды косой цилиндрический

На рисунке 4.14 показан дважды косой коноид, образованный перемещением образующей прямой по трем направляющим, из которых две - прямые. Показано построение одной образующей, как результата пересечения вспомогательной плоскости, проходящей через одну из прямолинейных направляющих, с двумя другими направляющими.

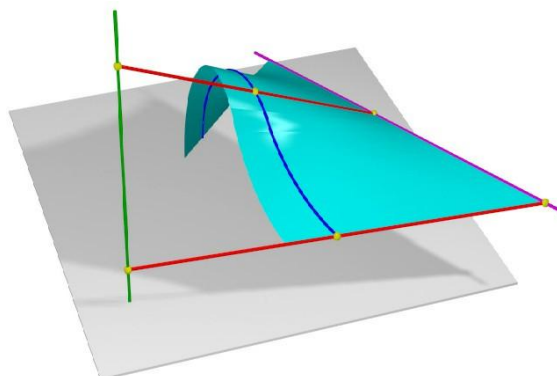


Рисунок 4.14 - Дважды косой коноид

Поверхности с плоскостью параллелизма в аналогичных случаях соответственно называются прямыми цилиндрами, прямыми коноидами и косой плоскостью.

Линейчатые поверхности с плоскостью параллелизма (поверхности Каталана)

Прямой цилиндр. Прямым цилиндром называется поверхность, образованная движением прямой линии, скользящей по двум криволинейным направляющим, не принадлежащим одной плоскости, и остающейся во всех своих положениях параллельной некоторой заданной плоскости. Эта плоскость называется плоскостью параллелизма.

Через каждую точку поверхности проходит одна образующая. Геометрическая часть определителя цилиндра состоит из двух направляющих кривых линий (m и n) и плоскости параллелизма (Σ). Алгоритмическая часть определителя состоит из указания о том, что любая образующая поверхности может быть построена как прямая, пересекающая направляющие кривые и параллельная плоскости параллелизма.

Прямой коноид. Прямым коноидом называется поверхность, образованная движением прямой линии, скользящей по двум направляющим, одна из которых - кривая, а вторая - прямая, и остающейся во всех своих положениях параллельной некоторой плоскости параллелизма. Коноид, направляющими которого являются кривая и прямая, а плоскостью параллелизма - плоскость, изображен на рисунке 4.15.

Геометрическая часть определителя коноида состоит из двух направляющих линий (прямой и кривой) и плоскости параллелизма. Алгоритмическая часть определителя состоит из указания о том, что любая образующая поверхности может быть построена как прямая, пересекающая направляющие линии и параллельная плоскости параллелизма.

Косая плоскость. Косой плоскостью называется поверхность, образованная движением прямой линии, скользящей по двум скрещивающимся прямым и остающейся во всех своих положениях параллельной некоторой плоскости параллелизма.

Косая плоскость имеет два семейства прямолинейных образующих и две плоскости параллелизма. Образующие одного семейства - скрещивающиеся прямые, каждая образующая

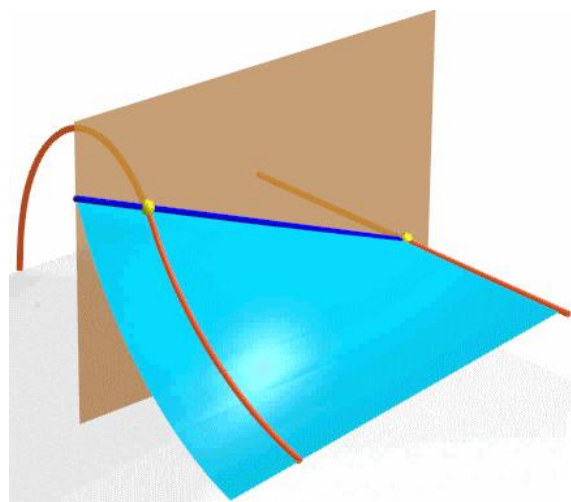


Рисунок 4.15 - Прямой коноид

одного семейства пересекает все образующие второго. Поэтому через каждую точку поверхности проходят две прямолинейные образующие разных семейств (рисунок 4.16).

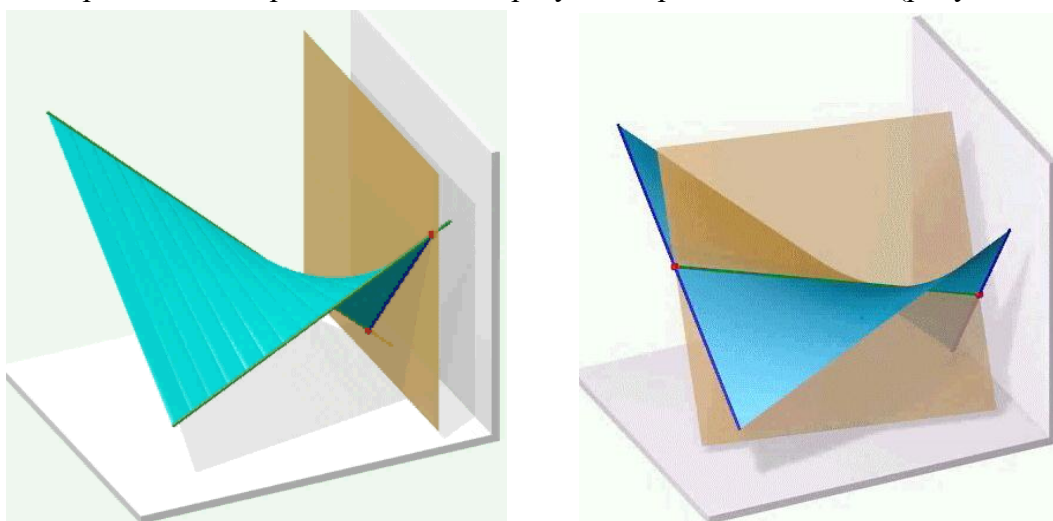


Рисунок 4.16 - Косая плоскость

Косую плоскость называют также гиперболическим параболоидом, так как при пересечении её соответствующими плоскостями в сечении можно получить параболы и гиперболы (рисунок 4.17а).

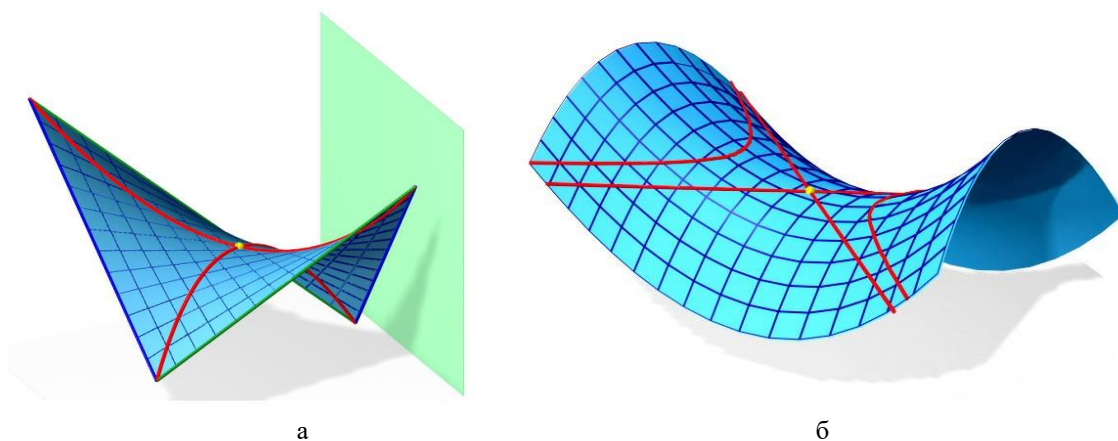


Рисунок 4.17 - Гиперболический параболоид и поверхность Каталана

Геометрическая часть определителя кривой плоскости состоит из направляющих прямых и плоскости параллелизма. Алгоритмическая часть определителя состоит из указания о том, что любая образующая поверхности может быть построена как прямая, пересекающая направляющие прямые и параллельная плоскости параллелизма.

Косая плоскость может быть получена как поверхность Каталана путем плоскопараллельного перемещения одной из парабол, как образующей, по второй параболе, как направляющей. Результат этого кинематического варианта формирования косой плоскости показан на рисунке 4.176. На рисунке 4.176 также показан пример гиперболического сечения рассматриваемой поверхности и его вырожденный случай - две прямые, проходящие через "седловую" точку.

Винтовые поверхности

Поверхность, образованная винтовым движением прямой линии, называется линейчатой винтовой поверхностью - геликоидом (винтовое движение характеризуется вращением вокруг некоторой оси i и поступательным перемещением, параллельным этой оси).

Прямой геликоид. Если в качестве кривой направляющей коноида взять цилиндрическую винтовую линию, в качестве прямой направляющей - ось винтовой линии, а за плоскость параллелизма принять плоскость, перпендикулярную оси винтовой линии, то поверхность, образованная при этих условиях, называется винтовым коноидом или прямым геликоидом (рисунок 4.18).

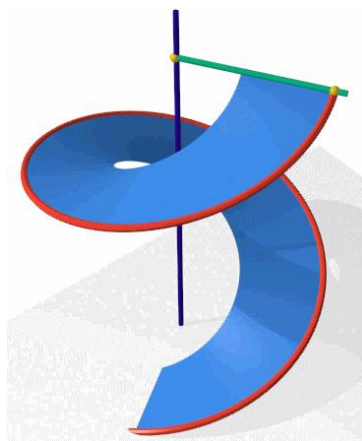


Рисунок 4.18 - Винтовой коноид (прямой геликоид)

На рисунке 4.19 приведен пример прямого открытого геликоида, имеющего конечную толщину поверхности.

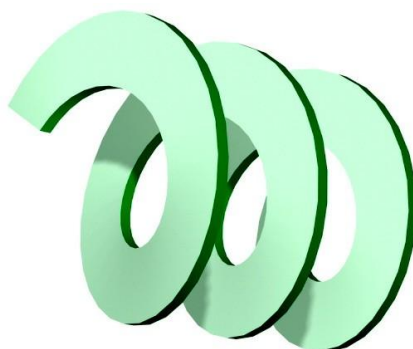


Рисунок 4.19 - Прямой открытый геликоид

Поверхности вращения

Если перемещение образующей линии представляет собой вращение вокруг некоторой неподвижной прямой (оси), то образованная в этом случае поверхность называется поверхностью вращения (рисунок 4.20).



Рисунок 4.20 - Поверхность вращения

Образующая линия может быть плоской или пространственной кривой, а также прямой. Каждая точка образующей линии при вращении вокруг оси описывает окружность, которая располагается в плоскости, перпендикулярной оси вращения. Эти окружности называются параллелями. Следовательно, плоскости, перпендикулярные оси, пересекают поверхность вращения по параллелям. Линия пересечения поверхности вращения плоскостью, проходящей через ось, называется меридианом. Все меридианы поверхности вращения конгруэнтны. Множество всех параллелей или меридианов представляет собой непрерывный каркас поверхности вращения. Через каждую точку поверхности проходит одна параллель и один меридиан.

Геометрическая часть определителя поверхности вращения состоит из оси вращения i и образующей линии l . Алгоритмическая часть определителя поверхности вращения состоит из операции вращения образующей l вокруг оси i и построения каркаса параллелей необходимой плотности.

При проектировании различных инженерных сооружений, машин и механизмов наибольшее распространение получили поверхности, образующиеся вращением прямой линии и кривых второго порядка.

Поверхности, образуемые вращением прямой (линейчатые поверхности вращения) (рисунок 4.21).

Вращением прямой линии образуются:

- цилиндр вращения, если прямая l параллельна оси i ;
- конус вращения, если прямая l пересекает ось i ;

-однополостный гиперболоид вращения, если прямая l (BC) скрещивается с осью i .

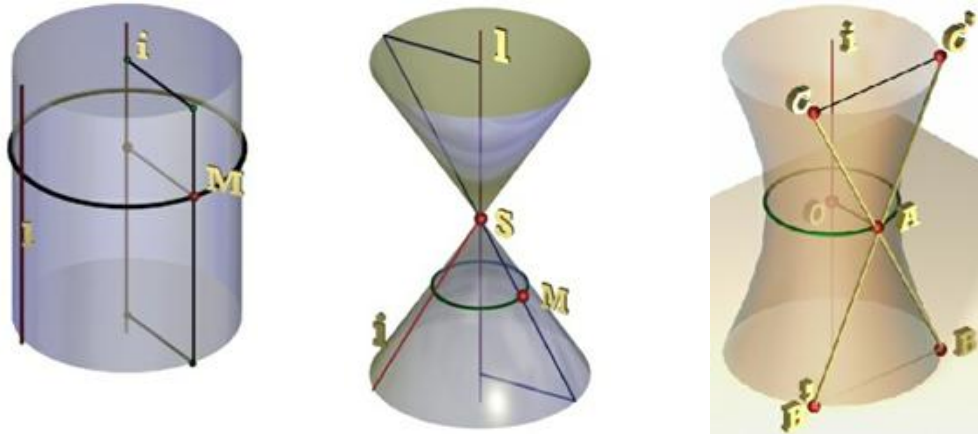


Рисунок 4.21 - Линейчатые поверхности вращения

Поверхность однополостного гиперболоида имеет две образующие линии l (BC) и l' (B'C'), наклоненные в разные стороны и пересекающиеся в точке (A), принадлежащей наименьшей параллели. Отрезок OA является кратчайшим расстоянием между образующей и осью. Таким образом, на поверхности однополостного гиперболоида располагаются два семейства прямолинейных образующих. Все образующие одного семейства - скрещивающиеся прямые. Каждая образующая одного семейства пересекает все образующие другого. Через каждую точку поверхности проходят две образующие разных семейств. Меридианом поверхности является гипербола.

Все рассмотренные линейчатые поверхности вращения являются поверхностями второго порядка.

Поверхности, образуемые вращением кривых второго порядка вокруг их осей.

Сфера образуется вращением окружности вокруг её диаметра.

Эллипсоид вращения образуется вращением эллипса вокруг большой или малой оси.

Параболоид вращения образуется вращением параболы вокруг её оси.

Однополостный гиперболоид вращения образуется вращением гиперболы вокруг её мнимой оси. Эта поверхность образуется также вращением прямой.

Двуполостный гиперболоид вращения образуется вращением гиперболы вокруг её действительной оси.

При вращении асимптот гиперболы образуется конус вращения, который называется асимптотическим по отношению к поверхности гиперболоида.

Все рассмотренные поверхности вращения являются поверхностями второго порядка.

Поверхности, образуемые вращением кривых второго порядка вокруг оси, не являющейся осью кривой, но расположенной в её плоскости. Существует теорема: "При вращении плоской или пространственной алгебраической кривой n -го порядка вокруг произвольной оси образуется алгебраическая поверхность вращения, имеющая в общем случае порядок $2n$ ". Из этой теоремы следует, что при вращении кривой второго порядка вокруг оси, не являющейся осью кривой, но расположенной в её плоскости, образуется

поверхность четвертого порядка. Наиболее распространенной поверхностью четвертого порядка является тор (рисунок 4.22).



Рисунок 4.22 - Тор

Тором называется поверхность, образованная вращением окружности вокруг оси, принадлежащей плоскости окружности, но не проходящей через её центр. При этом ось вращения может пересекать окружность, касаться её и располагаться вне окружности. В первых двух случаях тор называется закрытым, в последнем - открытым или кольцом.

Каналовые и циклические поверхности

Каналовой называют поверхность, образованную непрерывным каркасом замкнутых плоских сечений, определенным образом ориентированных в пространстве (рисунок 4.23). Площади этих сечений могут оставаться постоянными или монотонно изменяться в процессе перехода от одного сечения к другому.

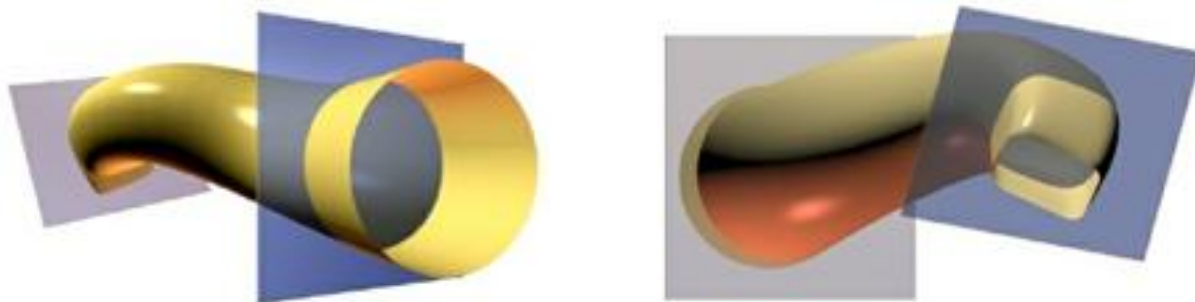


Рисунок 4.23 - Каналовые поверхности

В инженерной практике наибольшее распространение получили два способа ориентирования плоскостей образующих:

- параллельно какой-либо плоскости - каналовые поверхности с плоскостью параллелизма;
- перпендикулярно к направляющей линии - прямые каналовые поверхности.

Каналовая поверхность может быть использована для создания переходных участков между двумя поверхностями типа трубопроводов, имеющих:

- различную форму, но одинаковую площадь нормального сечения;
- одинаковую форму, но различные площади сечения;
- различную форму и различные площади поперечных сечений.

Поверхность Эшера - это каналовая поверхность, направляющей которой является пространственный трехлистник (рисунок 4.24). Последний можно представить как нить, намотанную по поверхности тора.

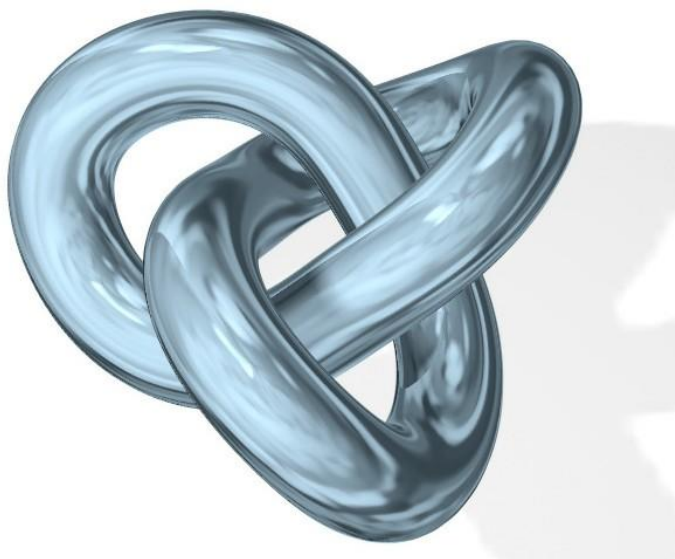


Рисунок 4.24 - Поверхность Эшера

Циклическую поверхность (рисунок 4.25) можно рассматривать как частный случай каналовой поверхности. Она образуется с помощью окружности, центр которой перемещается по криволинейной направляющей. В процессе движения радиус окружности монотонно меняется.

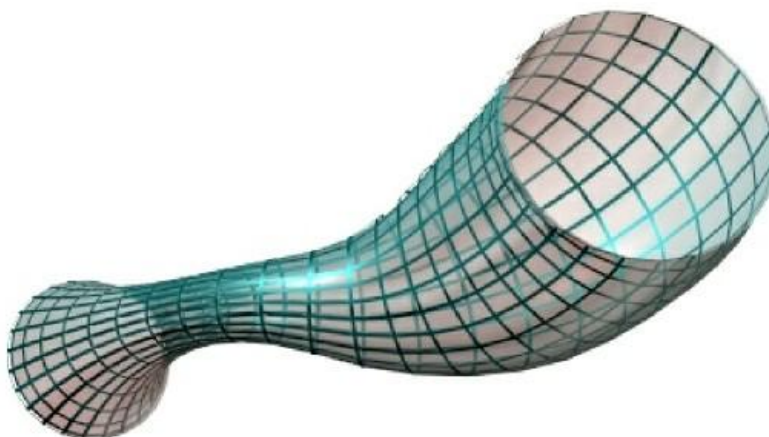


Рисунок 4.25 - Циклическая поверхность

Трубчатая поверхность (рисунок 4.26) относится к группе нелинейчатых поверхностей с образующей постоянного вида и является частным случаем циклической и каналовой поверхностей. Она обладает свойствами, присущими этим видам поверхностей. У циклической поверхности она позаимствовала форму образующей, а у каналовой - закон движения этой образующей.

Кинематический метод построения поверхностей, в частности, каналовых поверхностей представлен на рисунке 4.27.

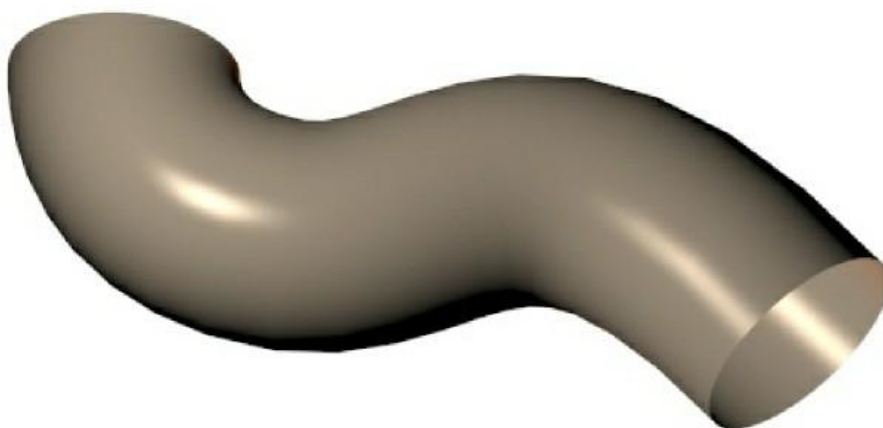


Рисунок 4.26 - Трубчатая поверхность

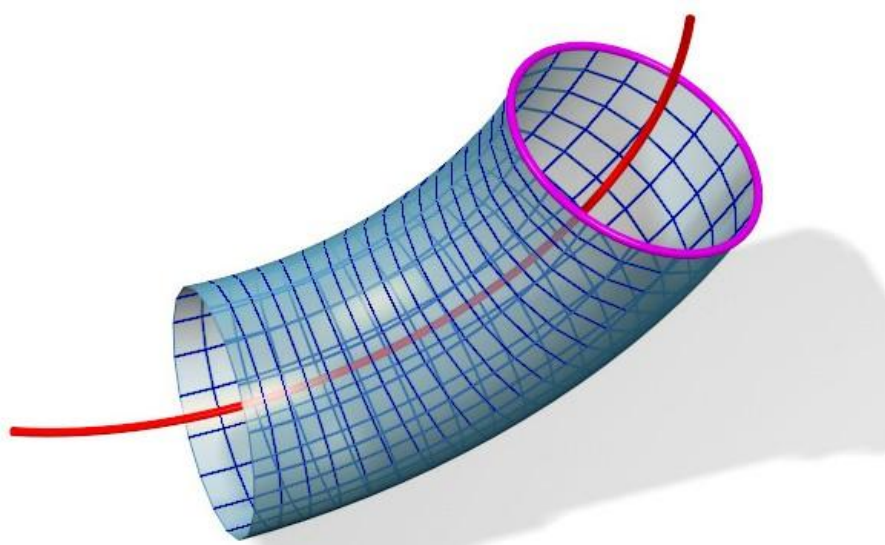


Рисунок 4.27 - Кинематический метод образования поверхности

Развёртки

Поверхность называется развертывающейся, если она путем изгибания может быть совмещена с плоскостью без образования складок и разрывов. При этом исходят из представления поверхности как гибкой, но нерастяжимой и несжимаемой пленки. Свойством развертываемости обладают многогранные поверхности и кривые линейчатые поверхности с ребром возврата: торсы, конические и цилиндрические. Линейчатые косые и нелнейчатые поверхности этим свойством не обладают. Существуют различные способы построения их условных разверток при помощи аппроксимации.

Плоская фигура, полученная в результате совмещения поверхности с плоскостью, называется разверткой. Между поверхностью и её разверткой существует взаимно-однозначное точечное соответствие (точке A на поверхности соответствует точка A' на развёртке, и наоборот), обладающее следующими свойствами:

- длина участка AB линии l на поверхности равна длине участка $A'B'$ соответствующей ей линии l' на развертке;

-угол α между кривыми m и n на поверхности равен углу α' между соответствующими им кривыми m' и n' на развертке (углом между кривыми называется угол между касательными к ним в точке пересечения);

-площадь отсека F поверхности равна площади соответствующего ему отсека F' развертки.

В дифференциальной геометрии доказывается, что второе и третье свойства являются следствием первого. Первое свойство вытекает из представления поверхности как гибкой, но нерастяжимой и несжимаемой пленки.

Из рассмотренных свойств следует (рисунок 4.28):

-прямой линии (a) на поверхности соответствует прямая (a') на развертке;

-прямым, параллельным ($a \parallel b$) на поверхности, соответствуют прямые, параллельные ($a' \parallel b'$) на развертке.

Однако оба указанных свойства обратной силы не имеют, т.е. не всякой прямой на развертке соответствует прямая на поверхности. Примерами этого могут служить цилиндрическая винтовая линия, параллели поверхности вращения. Если кривой линии, принадлежащей поверхности, соответствует прямая на развертке, то эта кривая линия является геодезической для данной поверхности.

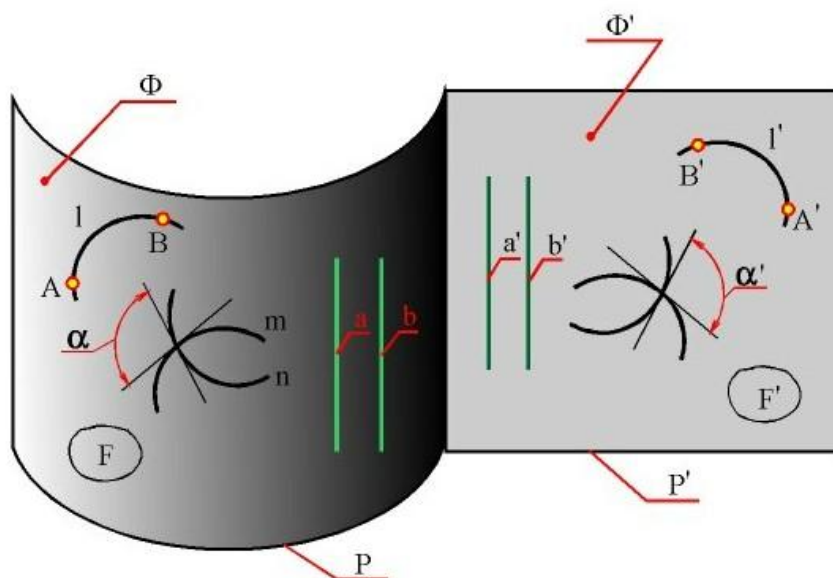


Рисунок 4.28 - Поверхность и ее развертка

Развертка многогранника представляет собой плоскую фигуру, полученную при совмещении всех его граней с плоскостью. Следовательно, построение развертки многогранника сводится к построению истинных величин его граней. Выполнение этой операции связано с определением натуральных величин его ребер, которые являются сторонами многоугольников - граней, а иногда и некоторых других элементов. Ребра многогранника условно разделяются на боковые и стороны основания.

Боковые грани любой пирамиды являются треугольниками (рисунок 4.29). Для построения развертки пирамиды необходимо предварительно определить натуральные величины боковых ребер и сторон основания.

Построение точных разверток кривых разветывающихся поверхностей традиционными методами сложно и, как правило, не вызывается практической необходимостью. Поэтому обычно строят приближенные развертки поверхностей. Основным способом построения приближенных разверток разветывающихся поверхностей (кроме цилиндрических) является способ триангуляции поверхности.

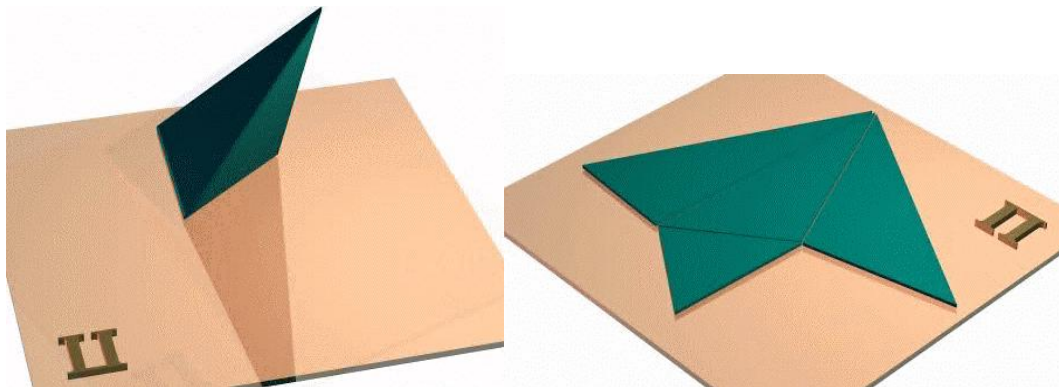


Рисунок 4.29 - Пирамида и ее развертка

Способ триангуляции состоит в том, что кривая поверхность заменяется многогранной поверхностью, состоящей из треугольных граней.

Развертка прямого кругового конуса, образующая которого равна l и радиус основания r , имеет форму кругового сектора с радиусом равным l и центральным углом $\alpha = 360^\circ r/l$ (рисунок 4.30).

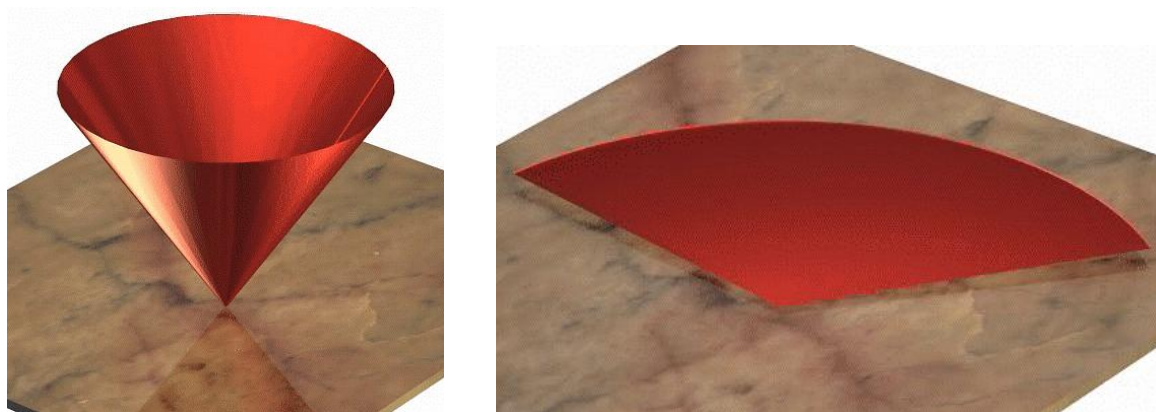


Рисунок 4.30 - Конус и его развертка

При построении разверток цилиндрических поверхностей способ триангуляции, как правило, не применяется. Цилиндрическая поверхность заменяется (аппроксимируется) вписанной в неё призматической поверхностью, которая определяется ломаной, вписанной в направляющую кривую цилиндра, и направлением образующих. Развёртка этой n -угольной призмы и принимается за развертку цилиндра. Ломаная линия, получающаяся на развертке призмы, заменяется плавной кривой. Развертка боковой поверхности прямого кругового цилиндра представляет собой прямоугольник со сторонами, соответственно равными $2\pi r$ и h , где r - радиус окружности основания цилиндра, а h - его высота (рисунок 4.31).

Развёртку неразветывающейся поверхности построить нельзя. Для построения условной развёртки такой поверхности применяют метод аппроксимации, который заключается в следующем. Данная неразветывающаяся поверхность Φ разбивается на некоторые отсеки.

Каждый из этих отсеков заменяется отсеком кривой разvertyвающейся поверхности. Совокупность всех отсеков разvertyвающихся поверхностей называется обводом Φ' поверхности Φ . С помощью триангуляции обвод Φ' заменяется обводом Φ'' гранных поверхностей. Развертка гранных поверхностей, образующих обвод Φ'' , принимается за условную развертку поверхности Φ . При свёртывании такой развертки, кроме изгибания, необходимо произвести частичное растяжение или сжатие отдельных её участков.

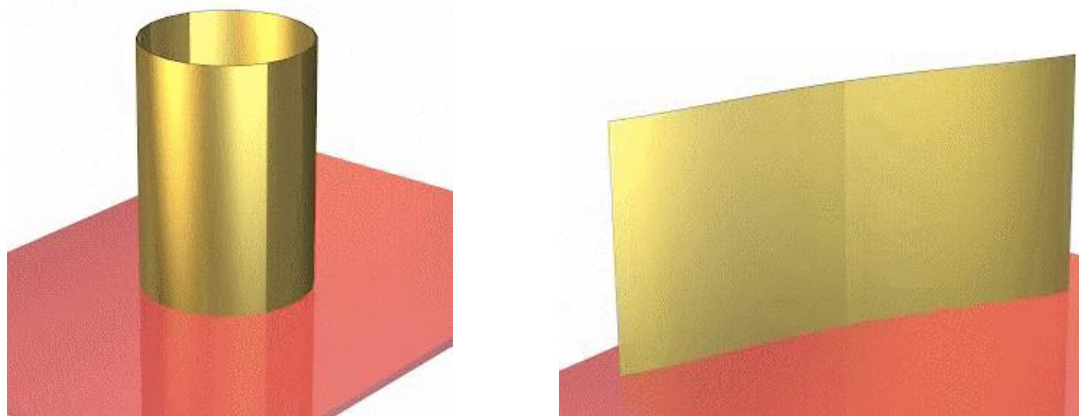


Рисунок 4.31 - Цилиндр и его развертка

Рассмотрим применение этого способа на примере построения условной развертки сферы. Разделим поверхность сферы (рисунок 4.32а) на некоторое число (например, шесть) одинаковых отсеков при помощи осевых плоскостей Σ , Δ , Λ . Поверхность каждого отсека сферы заменим отсеком описанной цилиндрической поверхности. В результате поверхность сферы заменяется обводом (составной поверхностью), составленным из отсеков прямых круговых цилиндров. Поверхность каждого отсека цилиндрической поверхности заменим отсеком вписанной призматической поверхности (рисунок 4.32б). В результате обвод, составленный из отсеков цилиндров, заменяется обводом, составленным из гранных поверхностей (отсеков прямых призм). Строим развертку каждого отсека призматической поверхности. На чертеже (рисунок 4.32в) показана развёртка одного из них. Затем ломаная 1-3-5-7... заменяется плавной кривой, проходящей через те же точки (рисунок 4.32г). Полученная фигура принимается за условную развёртку отсека сферы. Полная развертка будет состоять из шести таких фигур (рисунок 4.32д).

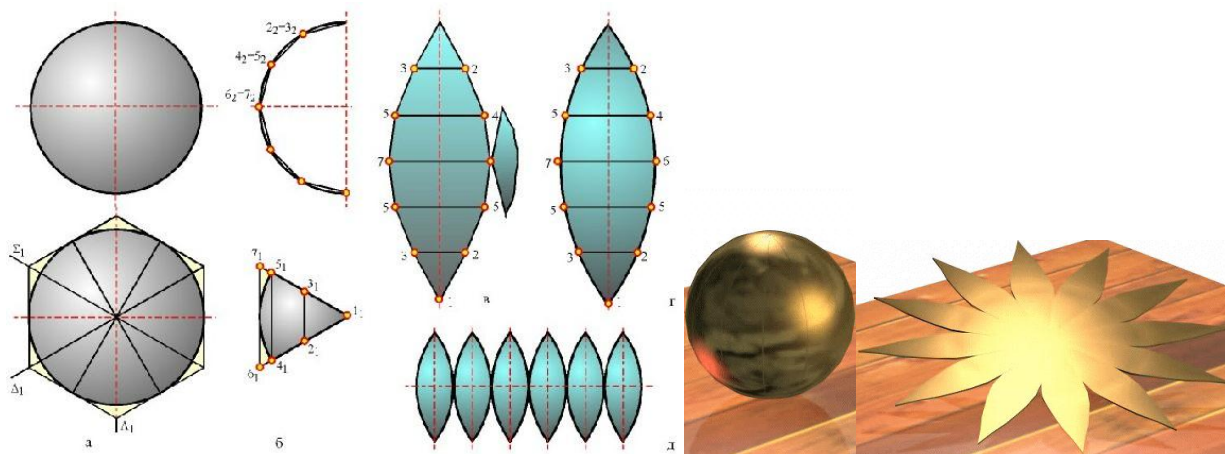


Рисунок 4.32 - Сфера и её развертка

Применяя в качестве аппроксимирующих поверхностей цилиндрические, конические или торсовые поверхности, аналогично можно строить условные развертки других неразвертывающихся поверхностей.

Вопросы для самоконтроля

1. Что собой представляет многогранник? Дайте определение многогранника.
2. Перечислите тела Платона (5). Что они собой представляют?
3. Перечислите способы (3) задания кривых поверхностей. В чём их особенности?
4. Что собой представляет определитель поверхности? Из каких частей (2) от состоит?
5. Приведите примеры образования цилиндрической и конической поверхностей вращения.
6. Приведите определитель построения сферы.
7. Дайте определение линейчатой поверхности.
8. Приведите пример формообразования торса.
9. Приведите примеры, алгоритмы формообразования косых (неразвёртывающихся) поверхностей.
10. Приведите примеры, алгоритмы формообразования поверхностей с плоскостью параллелизма.
11. Приведите примеры, алгоритмы формообразования винтовых поверхностей и поверхностей вращения.
12. Приведите примеры, алгоритмы формообразования каналовых и циклических поверхностей.
13. В чём состоит суть получения развёрток поверхностей?
14. Приведите примеры, алгоритмы получения развёрток многогранников.
15. Приведите примеры, алгоритмы получения развёрток кривых развёртывающихся поверхностей.
16. Приведите примеры, алгоритмы получения развёрток неразвёртывающихся поверхностей.

Лекция 5

Ядра геометрического моделирования

Рассмотрим состояние вопроса в хронологической последовательности.

Для начала определимся с понятиями. Что такое геометрическое ядро, какие функции оно реализует, насколько эти функции критичны для пользователей САПР?

Любая САПР представляет собой набор конечно-пользовательских приложений, разделяющих общую платформу, в основе которой лежит несколько базовых технологий, как правило, реализуемых в отдельных модулях, называемых технологическими компонентами. Каждая компонента отвечает за что-то свое: создание и редактирование трехмерной модели, ее визуализацию на дисплее, параметризацию, импорт/экспорт данных в определенном формате и т.д. Такие компоненты разработчики САПР либо проектируют, кодируют и поддерживают самостоятельно, либо лицензируют их у сторонних технологических поставщиков.

Технологические компоненты - это фундамент, на котором строится здание САПР. От этого фундамента зависит все остальное - возможности различных инструментов, их быстродействие, устойчивость к ошибкам, и даже общая интеллектуальность системы. Заложив в фундамент определенную компоненту, разработчик понимает, что впоследствии заменить ее на другую будет довольно трудно. Часто эта задача решается разборкой старого здания и строительством нового, т.е. выпуском нового поколения (не путать с очередной версией) программного продукта.

С точки зрения программиста каждая компонента - это набор библиотек с программным интерфейсом (API), который позволяет вызывать различные функции из приложения. В случае геометрического ядра речь идет об использовании базовых типов геометрических данных (точка, прямая, кривая, поверхность) и операций с ними (трансформация, проекция, пересечение), моделировании топологии граничной модели, реализации булевых операторов и типичных команд создания и редактирования трехмерных тел и поверхностей (заметание профиля, скругление ребра), построении треугольной сетки и экспорте/импорте данных в разных форматах (родном для ядра или нейтральном IGES/STEP).

Перечисление функций ядра занимает один абзац, но их реализация растягивается на десятки и сотни человеко-лет. Дело в том, что за каждой элементарной операцией (типа пересечения двух NURBS) стоит вычислительный алгоритм, реализация и отладка которого является весьма трудоемкой задачей. А таких операций в ядре - сотни (с учетом разнообразия типов данных, с которыми приходится работать). Далеко не все САПР-разработчики готовы инвестировать в таком объеме в базовую технологию и предпочитают лицензировать готовое ядро, осуществляя компании - разработчику ежегодные платежи (часто включающие определенную сумму с каждой проданной копии конечного продукта). Важно и то, что лицензированное ядро, как правило, уже было «обкатано» в других системах, поэтому обладает богатой функциональностью и высоким уровнем надежности. Наконец, выбрав стороннее ядро, поставщик САПР сможет быстрее закончить разработку своего продукта и вывести его на рынок. Иногда это время является определяющим - если опоздать с релизом на год-другой, то рынок может оказаться занятым конкурентами. Ярчайшим примером является выпуск САПР SolidWorks, ставшей первой системой параметрического твердотельного

моделирования для платформы Windows. Этим же путем впоследствии пошли разработчики MicroStation, T-FLEX, ADEM, SpaceClaim и множества других систем.

Немало и тех, кто ставит на возможность полного контроля над исходным кодом, оперативного исправления ошибок и наращивания функционала, быстрого переноса на новые платформы и готов расплачиваться за это собственными ресурсами. Этой трудной дорогой идут разработчики CATIA, NX, Creo Elements/Pro, Solid Edge, Autodesk Inventor, think3, КОМПАС-3D.

А началось все в далеком 1965 году, когда в компьютерной лаборатории Кембриджского университета было создано подразделение CAD Group. Эта команда ученых, возглавляемая Чарльзом Лэнгом (Charles Lang), проводила исследования в области создания программных средств, лежащих в основе систем MCAD/CAM. Вскоре к CAD Group присоединился Ян Брэйд (Ian Braid), который начал разрабатывать экспериментальную программу BUILD, систему геометрического моделирования на основе революционной для того времени технологии - граничного представления (BRep).

Здесь придется сделать небольшое отступление, чтобы рассказать об этой инновационной технологии - ведь ныне она лежит в основе большинства современных САПР. Граничные модели хранят информацию о границах объемного тела (гранях, ребрах и вершинах), что позволяет максимально точно описать деталь и рассчитать ее объемно-массовые характеристики. Для простоты манипулирования информация в граничной модели подразделяется на геометрические и топологические данные. Геометрические данные для каждой граничной сущности свои: для вершины - ее координаты, для ребра - параметрическое уравнение пространственной кривой (чаще всего - в виде NURBS - неоднородного рационального B-сплайна), для грани - параметрическое уравнение поверхности (опять же часто в виде NURBS). Топологические данные хранят информацию о смежности вершин и ребер, ребер и граней, а также о внутренних и внешних границах грани. Для удобного манипулирования топологической информацией было предложено несколько структур данных, обобщенно называемых BRep (от англ. Boundary Representation – граничное представление). Над разработкой первой в мире граничной структуры данных, которая позволяла описывать геометрию сложной формы, как раз и трудился в Кембридже Ян Брэйд.

В 1974 году ведущие участники CAD Group образуют компанию Shape Data Ltd., которая начинает разработку первого в мире коммерческого геометрического ядра ROMULUS.

Разработка ведется на языке Fortran и основывается на идеях, обкатанных в экспериментальных программах BUILD I и BUILD II. Вскоре продается первая коммерческая лицензия на ROMULUS. Покупатель - компания Hewlett-Packard - использует ее для создания своей CAD-системы ME30 (наследником которой является CoCreate OneSpace Modeler, ныне принадлежащая компании PTC и недавно переименованная в CREO Elements/Direct). В 1981 г. Shape Data становится частью компании Evans & Sutherland (одним из основателей которой был легендарный САПР-пионер Айвэн Сазерлэнд), а в 1985 году начинает разработку пакета Parasolid - прямого наследника геометрического ядра ROMULUS.

На этого наследника довольно скоро объявился серьезный претендент - компания Unigraphics, которая к тому времени осознала, что твердотельное моделирование на основе конструктивного подхода (CSG) - тупиковый путь развития. Основной продукт Unigraphics - САПР UniSolids (ныне всем известная под именем NX и принадлежащий Siemens PLM

Software), основанный на собственном CSG-ядре PADL, требовалось перевести на граничное представление твердых тел. Взвесив все варианты, в 1988 г. Unigraphics полностью выкупила Parasolid-бизнес у Evans & Sutherland и начала инвестировать в разработку ядра. Годом позже был выпущен конечно - пользовательский пакет UG/Solids, полностью основанный на Parasolid, но лицензировать Parasolid сторонним разработчикам компания начала лишь с середины 1990-х, когда на время стала частью компании EDS.

Вскоре после начала работы над Parasolid (активно поучаствовав в составлении соответствующего ТЗ) трое ведущих сотрудников Shape Data - Ян Брэйд, Чарльз Лэнг и Алан Грейер (Alan Grayer) - оставляют компанию и создают собственную под названием Three-Space Ltd., которая начинает разработку принципиально нового геометрического ядра - ACIS.

Разработка идет по заказу компании Spatial Technology (США), которая намеревается использовать ACIS в собственных целях. Дик Соуэр (Dick Sowar), основавший Spatial в 1986 г., сначала планировал использовать новое ядро для своей инновационной САМ-системы Strata, основанной на твердотельном моделировании, но через несколько лет понял, что бизнес по лицензированию самого ядра другим компаниям оказался более выгодным, и свернул остальные разработки. Дик вообще больше верил в перспективы компаний, поставляющих технологии, а не конечно-пользовательские продукты, поэтому инвестировал часть своих средств в проект еще одного кембриджского ученого - Джона Оуэна (John Owen), создавшего на эти деньги D-Cubed для разработки первого в мире коммерческого решателя геометрических ограничений. В дальнейшем базировавшиеся в Кембридже разработчики Three-Space (со временем поглощенной Spatial) и D-Cubed тесно сотрудничали друг с другом, что сыграло решающую роль в развитии событий вокруг ACIS в начале нового века.

Разработка ACIS с самого начала велась на C++, первая версия вышла в 1989 г. (она работала как на UNIX-платформах, так и на MS DOS), а первым покупателем снова стала HP. Вскоре лицензии на ACIS приобрели компании Control Data и Autodesk. Другими крупнейшими клиентами ACIS стали компании Bentley Systems и Intergraph.

За год до образования Spatial произошло одно знаменательное событие, которое вскоре перевернуло всю отрасль САПР: эмигрант из СССР Самуил Гейзберг основал компанию PTC (Parametric Technology Corp.) для разработки Pro/ENGINEER – первой в мире системы параметрического твердотельного моделирования на основе конструктивных элементов (использующей дерево построения для автоматической регенерации модели при внесении изменений). Pro/ENGINEER (ныне известный под именем CREO Elements/Pro) с самого начала базировался на собственном геометрическом ядре, спроектированном специально для поддержки твердотельного моделирования. Компания, впрочем, долго не решалась предложить свое ядро другим разработчикам, сделав это только в 2001 году. В это время рынок уже принадлежал ACIS и Parasolid, поэтому GRANITE (такое название получило ядро) лицензировался в основном для обмена данными с Pro/E, а не для создания собственных систем твердотельного моделирования. Такое положение это ядро занимает и поныне.

Успех Pro/E (новообразованная компания уже через несколько лет стала лидером отрасли по объемам продаж, став первой в истории САПР - компанией, пробившей планку в \$1 млрд. годового дохода) заставил всех конкурентов срочно реализовывать в своих системах аналогичный функционал. Самыми дальновидными оказались Джон Хёрштик (John Hirshtik) и Майкл Пейн (Michael Payne), которые, покинув PTC, в 1993 году создали собственную

компанию с целью разработки аналога Pro/E для платформы Windows. Стремясь ускорить разработку, они решили не разрабатывать собственное геометрическое ядро, а использовать готовое. Испробовав ACIS, разработчики остались неудовлетворенными его функционалом (многие функции, которые правильно работали в Pro/E, в ACIS давали ошибки). И, поскольку, попытки Майка договориться с руководителями PTC о лицензировании их геометрического ядра окончились неудачей (PTC прозорливо видела в будущем SolidWorks своего конкурента), а реакция Spatial на указанные ошибки в ACIS его не устроила, он решил протестировать Parasolid. В результате оказалось, что на наборе разработанных тестов Parasolid показал себя много лучше ACIS, и один из главных тендеров в мире САПР был решен в его пользу. Без сомнения, в успехе SolidWorks есть немалая заслуга разработчиков Parasolid.

Осознав, насколько выгодным может быть лицензирование своего ядра сторонним разработчикам, на рынок вышла японская компания Ricoh со своим продуктом Designbase. Ricoh использовала это ядро для дизайна своих фотоаппаратов, предъявляя высокие требования к сложности поверхностей. Designbase - первое ядро, в котором поверхности и твердые тела можно проектировать отдельно, легко переходя от одного представления к другому. Кроме того, Designbase использовало концепцию мета - моделирования, позволяющую автоматически создавать историю создания модели, и легко проводить параметрическую модификацию геометрии. В настоящее время ядро Designbase исчезло с рынка технологических компонент, но до сих пор можно встретить САПР, основанные на нем - это система твердотельного моделирования Helix, распространяемая в Японии компанией Fujitsu, CADRA (SoftTech), GMSWorks (C-Solutions) и другие.

Французской компании Dassault Systemes пришлось догонять ушедших вперед конкурентов. Будучи одним из пионеров в области трехмерного проектирования и реализовав возможности твердотельного моделирования еще в четвертой версии CATIA (выпущенной в 1993 г.), компания - как и многие другие старожилы рынка САПР - пропустила появление Windows. Осознав, что полноценный перенос на новую платформу CATIA 4, написанной на языке Fortran и ориентированной на архитектуру UNIX, невозможен, Dassault решила создать новое поколение CATIA с нуля. Новая разработка, получившая кодовое имя CNext, в 1998 году стала известной всему миру под названием CATIA V5 (после чего потребовалось еще несколько долгих лет на доведение V5 до функционального уровня своей предшественницы). Вместо устаревшего Fortran в качестве языка программирования был выбран C++, а геометрическое ядро новой системы получило название CGM - CNext/CATIA Geometric Modeler и было написано с нуля - ни строчки кода из прежней системы в нем не было.

Выпущенная в 2008 г. PLM-платформа V6 базируется на том же геометрическом ядре CGM. Все эти годы ядро непрерывно совершенствовалось и развивалось. С самого начала CGM позволяло моделировать геометрию, состоящую из многообразий разной размерности: нулевых (точки), 1-manifold (криволинейные каркасы), 2-manifold (поверхности) и 3-manifold (твердые тела), произвольно смешивая их в рамках одной модели.

Сделав ставку на новое современное ядро, Dassault выиграла - CATIA V5 и смежные PLM-продукты вывели компанию в абсолютные лидеры мирового рынка САПР (как когда-то PTC). Однако, за это пришлось заплатить высокую цену: данные, созданные в CATIA 4 (основанной на старом ядре), нельзя полноценно использовать в V5. Точнее расплатиться

пришлось крупнейшему клиенту Dassault – концерну Airbus, который, проектируя самый большой пассажирский самолет A380 одновременно в четвертой и пятой версиях CATIA, сорвал все сроки с вводом его в эксплуатацию и понес убытки в размере \$6 млрд. Причина – ошибки при переносе геометрических моделей между разнородными САПР.

Никто и не думал, что геометрическое ядро CATIA (святая святых компании) когда-нибудь будет доступно для лицензирования. Но летом 2010 г. компания всех удивила, начав через свою дочернюю структуру Spatial предлагать CGM всем желающим (конечно, для этого пришлось изменить расшифровку первой буквы: “C” теперь не “CATIA”, а “Convergence”). Таким образом, последнее ядро из «большой четверки» (Parasolid, ACIS, GRANITE, CGM) стало доступно для лицензирования.

Впрочем, CGM было не единственным французским ядром твердотельного моделирования. Еще в начале 1990-х другая компания из этой страны - Matra Datavision - начала работу над средой CAS.CADE, набором компонент, лежащих в основе САПР. Первым основанным на нем продуктом стала CAD/CAM - система EUCLID QUANTUM, выпущенная Matra в 1996 г. Впрочем, особого успеха на рынке она не снискала, а в 1999 Dassault Systemes выкупила у Matra большую часть САПР-бизнеса. Так как ядро CAS.CADE осталось невостребованным (у Dassault в руках уже было собственное ядро CGM), то Matra приняла решение опубликовать его исходный код и организовать сервисную компанию для обслуживания разработчиков, решившихся взять это ядро за основу своих продуктов. Так появился Open CASCADE.

Бизнес, основанный на открытом коде, давно доказал свою эффективность. Одно из подтверждений тому - судьба компании Open CASCADE SAS, которая, сменив несколько владельцев, по-прежнему продолжает свое существование: последний на сегодняшний день релиз ядра (6.4) датируется октябрём 2010 г.

Многие другие компании предпринимали попытки разработки геометрических ядер, но заметных результатов не достигли. Можно, например, вспомнить XOX Corporation со своим ядром Shapes (которое использовалось в первых версиях CADKEY, будучи затем заменено на ACIS). Американская компания IntegrityWare с 1996 г. разрабатывает набор библиотек для геометрических вычислений. С 1998 г. их распространением (под именем SMLib) занимается компания Solid Modeling Solutions, которая предлагает каждому купившему лицензию не только объектный, но и исходный код ядра вместе с правом модифицировать его в своих целях. А в 2004 г. IntegrityWare выпустила ядро геометрического моделирования SOLIDS++. Реализованное на языке программирования C++, ядро имеет модульную архитектуру, поддерживает в рамках одной модели топологию разной размерности, обладает настраиваемыми геометрическими допусками, позволяет выполнять триангуляцию модели, реализует скругления вершин со сложной топологией, может моделировать поверхности с контролем кривизны G2 и G3. Впрочем, о коммерческих САПР, созданных на этом ядре, пока ничего не известно.

Глядя на успехи англичан, французов, американцев и японцев, отечественные разработчики не могли сидеть, сложа руки. Движение возглавила компания АСКОН, выпустившая в 2000 г. первую версию КОМПАС-3D, полностью основанную на собственном геометрическом ядре. В 2002 почин поддержали в НТЦ АПМ, где создали собственное ядро APM Engine, используемое в системе твердотельного и поверхностного моделирования APM

Studio. Впрочем, оба этих ядра не предлагаются для лицензирования сторонним разработчикам.

С 2007 г. Правительство РФ осуществляет федеральную целевую программу «Национальная технологическая база» с целью создания новых передовых технологий и оборудования, внедрения разработанных технологий в производство, коммерциализации новых технологий, создания перспективного научно-технологического задела для разработки перспективной наукоемкой продукции, решения проблем улучшения экологической ситуации в стране.

В рамках этой программы Министерство промышленности и торговли РФ объявило тендер на выполнение научно-исследовательской и опытно-конструкторской работы «Создание отечественного лицензируемого программно-математического ядра трехмерного моделирования как базы для компьютерных систем автоматизированного проектирования сложной машиностроительной продукции». Такое ядро, будучи положено в основу нового поколения компьютерных систем проектирования, инженерного анализа, подготовки производства, создания технической документации, будет способствовать созданию российской промышленностью инновационных изделий, конкурентоспособных на мировом рынке, став частью национальной технологической базы.

Победителем тендера стало федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Московский государственный технологический университет «СТАНКИН», под руководством которого группа организаций в рамках проекта "Гербарий" (с 2011 по 2013 г.г.) разработала ядро RGK.

Конкуренция на рынке геометрических ядер между ACIS, Parasolid и Designbase, сложившаяся во второй половине 1990-х годов, привела к тому, что многие поставщики САПР получили возможность выбора: если они были не удовлетворены качеством используемого ядра или уровнем поддержки со стороны компании-разработчика, то могли сменить поставщика. Этой возможностью воспользовались Bentley Systems и Intergraph, переведя свои САПР - MicroStation и Solid Edge соответственно - с ACIS на Parasolid.

Компания 3D/EYE, разработчик IRONCAD, пошла еще дальше: встроив в 1995 г. в свою систему ядро ACIS, в 1999 она добавила туда Parasolid, разрешив пользователям свободно переключаться между ними. Система до сих пор сохраняет такую архитектуру, что помогает пользователям сократить затраты при работе с импортированной геометрией (созданной в другой САПР). Разные геометрические ядра обычно по-разному трактуют геометрические допуски, с разной точностью вычисляют пересечения кривых и поверхностей, в результате чего вполне корректная модель при переносе ее в систему, основанную на другом ядре, становится неработоспособной из-за многочисленных ошибок, которые приходится исправлять в полуавтоматическом режиме.

Компания Dassault пошла дальше. Приобретая в 2000 г. компанию Spatial, французы получили в свое распоряжение ядро ACIS – в дополнение к CGM, лежащему в основе CATIA, и лицензированному Parasolid, работающему в составе SolidWorks. Но руководители Dassault и не думали использовать ACIS в своих САПР. Вместо этого они поставили руководить Spatial Майкла Пейна, основателя SolidWorks, которому поручили провести переговоры с Autodesk – основным лицензиатом ACIS.

Изучив дела Spatial, Майк быстро понял, почему от этого ядра отказались Bentley и Intergraph (продукт был ненадежным, по возможностям он был полуфабрикатом, служба поддержки клиентов не существовала, цены все были разными в зависимости от того, сколько кто-то будет в состоянии заплатить в конце квартала, была нулевая система проверки качества).

На встрече с руководителем Autodesk Кэрл Барц Майк пообещал исправить ситуацию. Впрочем, в самой Autodesk тогда посчитали, что Майк, напротив, шантажирует их, угрожая вовсе прекратить развитие ядра, если Autodesk не примет предложение Dassault стать реселлером SolidWorks, убрав с рынка свой конкурирующий продукт Inventor. В результате Autodesk воспользовалась условием лицензионного соглашения и после единовременной выплаты \$6.5 млн. получила в свое распоряжение исходный код ACIS. Взяв его за основу, компания приступила к разработке собственного геометрического ядра, получившего название Shape Manager. Ответ на вопрос, где взять квалифицированных программистов для этой специфической работы, долго искать не пришлось - конечно в Кембридже, где же еще! Заключив соглашение с D-Cubed, Autodesk отдала эту разработку на аутсорсинг фактически авторам ACIS (разобраться в генетических связях Spatial и D-Cubed практически невозможно; можно лишь сказать, что компании тесно сотрудничали с 1986 г., а их ключевые сотрудники работали рука об руку в той самой CAD-лаборатории Кембриджского университета). Spatial подала на Autodesk в суд, но проиграла. Таким образом, кембриджское дерево принесло свой пятый плод (BUILD, ROMULUS, Parasolid, ACIS, Shape Manager).

Ядерные войны на этом закончились. А ядро ACIS, кстати, вполне успешно развивается в рамках Dassault. Регулярно выходят новые релизы, наращивается функционал по поддержке прямого редактирования геометрии. Неутомимый Майк Пейн, покинув Dassault, основал SpaceClaim, построив систему в этот раз не на Parasolid, а на ACIS. И благодаря личному знакомству разработчиков двух компаний, у них получилась отличная совместная работа, результатом которой теперь могут воспользоваться все клиенты Spatial. Не случайно, большинство систем прямого моделирования (SpaceClaim, KeyCreator, IRONCAD и частично Creo Elements/Direct) построены на ACIS.

Подведем итог. В таблице 1 представлены упомянутые геометрические ядра, включая те, которые никогда не были доступны для лицензирования сторонним разработчикам.

Таблица 1

Програмные продукты и ядра геометрического моделирования

Продукт	Производитель	Область	3D-ядро
4MCAD IntelliCAD	4M S.A., Греция	CAD, AEC	Open CASCADE Technology
Adams	MSC Software, США	CAE	Parasolid
ADEM	Группа компаний ADEM, Россия-Израиль- Германия	CAD, CAM, CAPP	ACIS

ADINA Modeler	ADINA R&D Inc., США	CAE	Parasolid и Open CASCADE Technology
Alibre Design	3D Systems, США	MCAD	ACIS
Allplan	Nemetschek AG, Германия	AEC/BIM	SMLib
AMPSolid	AMPS Technologies, США	CAE	ACIS
ANSYS	ANSYS Inc., США	CAE	ACIS и Parasolid
APM Studio	HTЦ АПИМ, Россия	MCAD	Собственное (APM Engine)
ArchiCAD	Graphisoft, Венгрия	AEC/BIM	Собственное
ARES	Graebert, Германия	CAD	ACIS
Ashlar-Vellum Cobalt, Xenon, Argon	Ashlar-Vellum, США	MCAD	ACIS
AutoCAD	Autodesk, США	CAD, AEC, GIS	Собственное (ASM), совместимое с ACIS
Autodesk Inventor	Autodesk, США	MCAD	Собственное (ASM), совместимое с ACIS
Autodesk Moldflow	Autodesk, США	CAE	Parasolid
Autodesk Revit Architecture	Autodesk, США	AEC/BIM	Собственное (ASM), совместимое с ACIS
bonzai3d	AutoDesSys, США	CAD	ACIS в комбинации с собственным ядром
Bricscad	Bricsys NV, Бельгия	AEC, MCAD	ACIS
BtoCAD	YuanFang Software Co., Ltd., Китай	CAD	ACIS
CADopia	CADopia Inc., США	CAD	ACIS
CATIA	Dassault Systemes, Франция	CAD/CAM/CAE, AEC	CGM

Cimatron	Cimatron Limited, Израиль	CAM	ACIS
CollabCAD	National Informatics Centre, Индия	CAD/CAM	Open CASCADE Technology
Creo (прежнее название – Pro/Engineer)	Parametric Technology, США	MCAD	GRANITE
Creo Elements/Direct Modeling (прежнее название – CoCreate)	Parametric Technology, США	CAD	ACIS
Edgecam	Planit Software, Великобритания	CAM	Parasolid и GRANITE
ESPRIT	DP Technology Corp., США	CAM	Parasolid
form-Z	AutoDesSys, США	CAD	ACIS в комбинации с собственным ядром
FreeCAD	Открытый онлайн-проект	CAD	Open CASCADE Technology
GibbsCAM	Cimatron, Израиль	CAD/CAM	Parasolid и GRANITE
GstarCAD	Suzhou Gstarsoft Co., Ltd, Китай	CAD	ACIS
IRONCAD	IronCAD LLC, США	MCAD	ACIS и Parasolid
KeyCreator	Kubotek USA Inc., Япония-США	CAD	ACIS
КОМПАС-3D	АСКОН, Россия	MCAD, AEC	Собственное ядро (C3D)
Mastercam	CNC Software, США	CAD/CAM	ACIS
Masterwork	Tecnos G.A., Италия	CAM	Open CASCADE Technology
MicroStation	Bentley Systems, США	AEC	Parasolid (ранние версии – ACIS)
Moment of Inspiration	Triple Squid Software Design, США	CAD	SOLIDS++
NX	Siemens PLM	CAD/CAM/CAE	Parasolid

	Software, Германия		
Patran	MSC Software, США	CAE	Parasolid
Power NURBS	Ideate Inc., США	CAD	SOLIDS++
PowerSHAPE	Delcam plc, Великобритания	CAD/CAM	Parasolid
progeCAD	progeCAD Srl Uninominale, Италия	CAD	ACIS
Radan	Planit, Великобритания	CAD/CAM	ACIS
Rhinoceros	Robert McNeel and Associates, США	CAD	SOLIDS++ (отдельные модули)
Shark LT	Encore, США	CAD	ACIS
SmartCAM	SmartCAMcnc	CAM	ACIS
Solid Edge	Siemens PLM Software, Германия	MCAD	Parasolid (ранние версии – ACIS)
SolidWorks	Dassault Systemes, Франция	MCAD	Parasolid
SpaceClaim	SpaceClaim Corp., США	MCAD	ACIS
STAR-CCM+	CD-adapco, Великобритания-США	CAE	Parasolid
StruCad	AceCad Software, Великобритания	AEC/BIM	Собственное
T-FLEX	Топ Системы, Россия	MCAD	Parasolid
ThinkDesign	Versata, США	MCAD	Собственное ядро
TopSolid	Missler Software, Франция	CAD/CAM	Parasolid
TurboCAD	IMSI/design, США	AEC, MCAD	ACIS
Vectorworks	Nemetschek, Германия	AEC	Parasolid (ранние версии - SMLib)
ViaCAD 2D/3D	Encore, США	CAD	ACIS
ZW3D (прежнее название – VX CAD/CAM)	ZWCAD Software, Китай	MCAD	Собственное ядро (VX Overdrive)

ZWCAD	ZWCAD Software, Китай	CAD	ACIS
-------	--------------------------	-----	------

Вопросы для самоконтроля

1. За что отвечают технологические компоненты в САПР?
2. Перечислите функции геометрического ядра.
3. Перечислите основных разработчиков геометрических ядер (фирмы и продукты).
3. Перечислите преимущества лицензированных ядер.
4. Где, когда и кем разработано первое коммерческое геометрическое ядро (ROMULUS)?
5. На каком представлении геометрии твёрдого тела и каком алгоритмическом языке создано ядро ROMULUS?
6. Как соответствуют между собой ядра ROMULUS и Parasolid?
7. На каком представлении геометрии твёрдого тела написано ядро PADL?
8. На каком представлении геометрии твёрдого тела и каком алгоритмическом языке создано ядро ACIS?
9. В чём состоят особенности ядра параметрического моделирования GRANITE?
10. Перечислите особенности геометрического ядра Designbase?
11. В чём состоят особенности ядра CGM?
12. Что представляет собой ядро Open CASCADE?
13. Перечислите особенности геометрического ядра SOLIDS++.
14. Какие ядра геометрического моделирования разработаны в РФ?
15. Как соответствуют между собой ядра ACIS и Shape Manadger?
16. Перечислите наиболее распространённые САПР. На основе каких геометрических ядер они созданы?

Лекция 6

Параметрическое моделирование

Процесс проектирования и конструирования, как правило, итеративный и предполагает перебор нескольких вариантов, поэтому упрощение и автоматизация построения модели будущего изделия являются одной из важнейших задач САПР.

Одним из широко распространенных методов решения этой задачи является параметрическое проектирование (или просто параметризация), основанное на моделировании деталей и изделий с использованием параметров элементов модели и соотношений между этими параметрами. Параметризация позволяет за короткое время перебрать с помощью изменения параметров или геометрических отношений различные конструктивные схемы, выбрать оптимальные решения и избежать принципиальных ошибок.

Определить цель конструирования достаточно просто, однако процесс поиска рационального решения сложен и требует гармоничного сочетания различных методов автоматизированного конструирования изделий. Параметрическое конструирование как методология автоматизированной разработки является основой для параллельного ведения проектно-конструкторских работ и позволяет уточнить конечную цель конструирования уже на ранних стадиях реализации проекта, что и определяет эффективность совмещения процессов конструирования, инженерного анализа и производства на едином временном интервале и их взаимной интеграции. Параметризация подразумевает использование различных видов взаимосвязей между компонентами модели и приложениями, которые используют данную модель.

Использование технологии параметрического конструирования позволяет, при необходимости, легко изменять форму модели, в результате чего пользователь имеет возможность быстро и эффективно получать альтернативные конструкции или пересмотреть концепцию изделия в целом. При отсутствии средств обеспечения параметрического конструирования модель определена однозначно только своей геометрией, поэтому внесение даже малейших изменений требует значительных трудовых затрат. Изменения же параметрической модели выполняются так же легко, как и изменения значения размеров на чертежах.

Параметризация - концепция, которая охватывает все методы для решения задач конструирования. Важной особенностью современной концепции параметрического конструирования является прежде всего возможность создания геометрической модели с использованием связей и правил, которые могут переопределяться и дополняться на любом этапе ее создания. Связи представляются в виде размерных, геометрических и алгебраических соотношений. Правила же определяются как условия выполнения базовой операции (например, сквозное или «глухое» отверстие).

Параметрическое проектирование существенно отличается от обычного двухмерного черчения или трехмерного моделирования. В случае параметрического проектирования создаётся по сути математическая модель объектов с параметрами, при изменении которых происходят изменения конфигурации и размеров деталей, их взаимного расположения в сборках и т. п.

На практике применяется достаточно много различных методов параметризации, на сегодняшний день нет однозначно преобладающего решения. Рассмотрим наиболее часто применяемые на практике методы.

Табличная параметризация

Табличная параметризация заключается в создании таблицы параметров типовых деталей (рисунок 6.1). Создание нового экземпляра детали производится путем выбора из таблицы типоразмеров. Возможности табличной параметризации весьма ограничены, поскольку задание произвольных новых значений параметров и геометрических отношений обычно невозможно.

Однако табличная параметризация находит широкое применение во всех параметрических САПР, поскольку позволяет существенно упростить и ускорить создание библиотек стандартных и типовых деталей, а также их применение в процессе конструкторского проектирования.

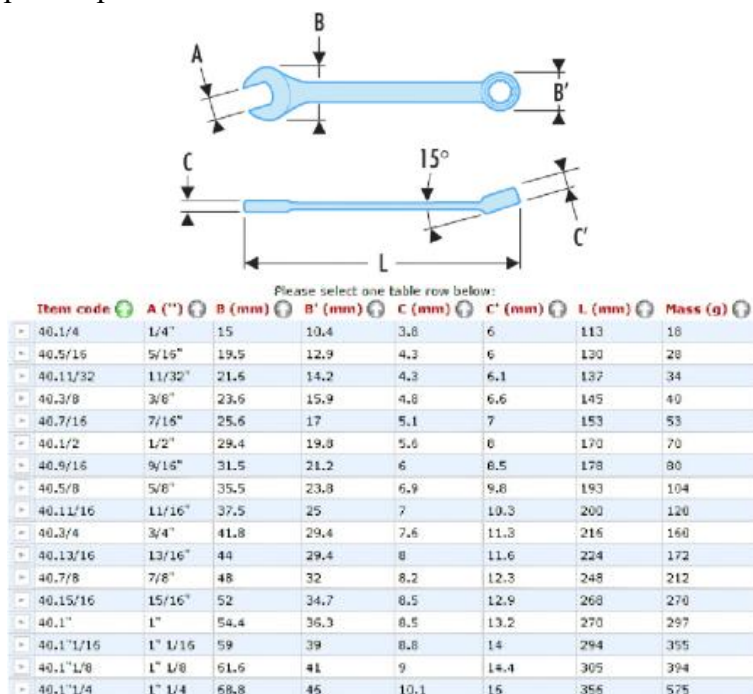


Рисунок 6.1 - Табличная параметрическая модель гаечного ключа

Иерархическая параметризация

Иерархическая параметризация (параметризация на основе истории построений) заключается в том, что в ходе построения модели вся последовательность построения отображается в отдельном окне в виде «дерева построения» (рисунок 6.2). В нем перечислены все существующие в модели вспомогательные элементы, эскизы и выполненные операции в порядке их создания.

Помимо «дерева построения» модели, система запоминает не только порядок ее формирования, но и иерархию ее элементов (отношения между элементами). (Например: сборки => под сборки => детали). Параметризация на основе истории построений присутствует практически во всех САПР, использующих трехмерное твердотельное

параметрическое моделирование. Обычно такой тип параметрического моделирования сочетается с вариационной и/или геометрической параметризацией.

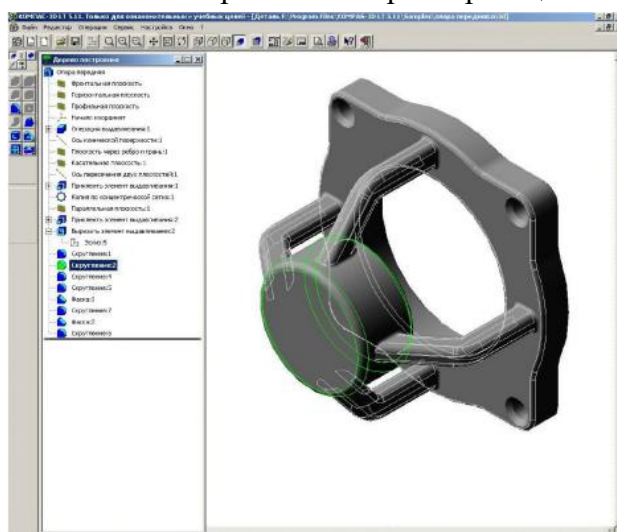


Рисунок 6.2 - Геометрическая модель с «деревом построения»

Вариационная (размерная) параметризация

Вариационная, или размерная, параметризация основана на построении эскизов (с наложением на объекты эскиза различных параметрических связей) и наложении пользователем ограничений в виде системы уравнений, определяющих зависимости между параметрами (рисунок 6.3).

Процесс создания параметрической модели с использованием вариационной параметризации выглядит следующим образом:

1. На первом этапе создается эскиз (профиль) для трехмерной операции. На эскиз накладываются необходимые параметрические связи.

2. Затем эскиз «образмеривается». Уточняются отдельные размеры профиля. На этом этапе отдельные размеры можно обозначить как переменные (например, присвоить имя Length) и задать зависимости других размеров от этих переменных в виде формул (например, $\text{Length}/2$).

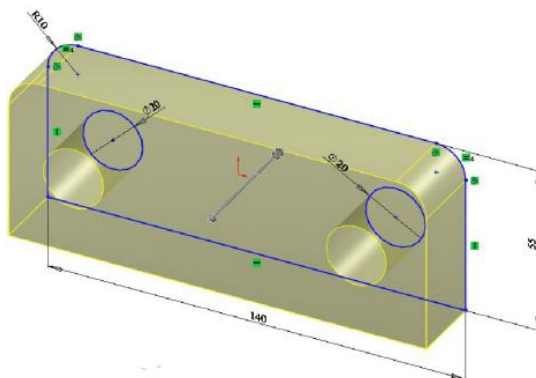


Рисунок 6.3 - Геометрическая модель с вариационной параметризацией

3. Затем производится трехмерная операция (например, смещение), значение атрибутов операции тоже служит параметром (например, величина смещения).

4. В случае необходимости создания сборки взаимное положение компонентов сборки задается путем указания сопряжений между ними (совпадение, параллельность или

перпендикулярность граней и ребер, расположение объектов на расстоянии или под углом друг к другу и т. п.).

Вариационная параметризация позволяет легко изменять форму эскиза или величину параметров операций, что позволяет удобно модифицировать трехмерную модель.

Частным случаем размерной параметризации является так называемая "эвристическая" параметризация, когда последовательно выполняется эскиз, который "образмеривается"; операцией смещения получают геометрическую модель (рисунок 6.4а). Затем производят редактирование значений размеров и выполняют операцию "эвристическая параметризация", в результате которой происходит автоматическое изменение геометрии профиля (рисунок 6.4б). Результат (рисунок 6.4в) получают после регенерации объемной модели.

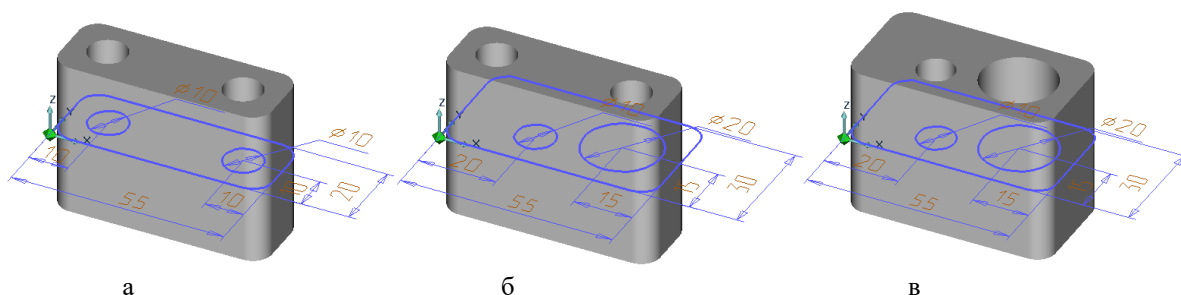


Рисунок 6.4 - "Эвристическая" параметризация

Геометрическая параметризация

Геометрической параметризацией называется параметрическое моделирование, при котором геометрия каждого параметрического объекта пересчитывается в зависимости от положения родительских объектов, его параметров и переменных.

Параметрическая модель в случае геометрической параметризации состоит из элементов построения и элементов изображения (рисунок 6.5). Элементы построения (конструкторские или вспомогательные линии) задают параметрические связи. К элементам изображения относятся линии изображения (которыми обводятся конструкторские линии), а также элементы оформления (размеры, надписи, штриховки и т. п.). Одни элементы построения могут зависеть от других элементов построения.

Элементы построения могут содержать и параметры (например, радиус окружности или угол наклона прямой). При изменении одного из элементов модели все зависящие от него элементы перестраиваются в соответствии со своими параметрами и способами их задания.

Процесс создания параметрической модели методом геометрической параметризации выглядит следующим образом:

1. На первом этапе конструктор задает геометрию профиля конструкторскими линиями, отмечает ключевые точки.
2. Затем проставляет размеры между конструкторскими линиями. На этом этапе можно задать зависимость размеров друг от друга.
3. Затем обводит конструкторские линии линиями изображения - получается профиль, с которым можно осуществлять различные трехмерные операции.

Последующие этапы в целом аналогичны процессу моделирования с использованием метода вариационной параметризации.

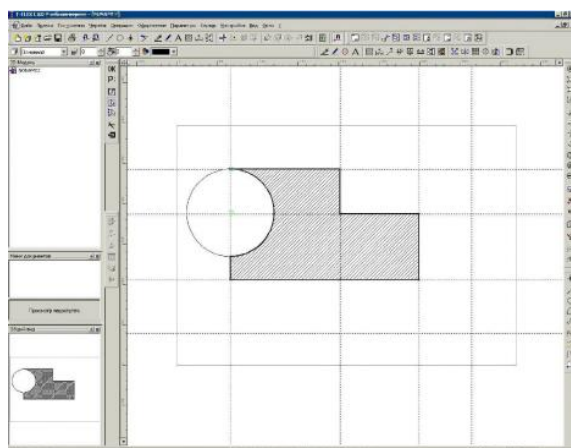


Рисунок 6.5 - Геометрическая параметризация основного контура с использованием направляющих линий

Геометрическая параметризация обеспечивает возможность более гибкого редактирования модели. В случае необходимости внесения незапланированного изменения в геометрию модели необязательно удалять исходные линии построения (это может привести к потере ассоциативных взаимосвязей между элементами модели), можно провести новую линию построения и перенести на нее линию изображения.

Ассоциативное конструирование

Ассоциативное конструирование (Associative Design) - это обобщающее название технологии параметрического конструирования, обеспечивающей единую, в том числе и двустороннюю, информационную взаимосвязь между геометрической моделью, расчетными моделями, программами для изготовления изделия на станках с ЧПУ, конструкторской документацией, базой данных проекта.

Использование технологии ассоциативного конструирования позволяет, при необходимости, изменять форму модели и получать автоматически перестроенные чертежи (рисунок 6.6) или траектории инструмента для обработки на станках с ЧПУ.

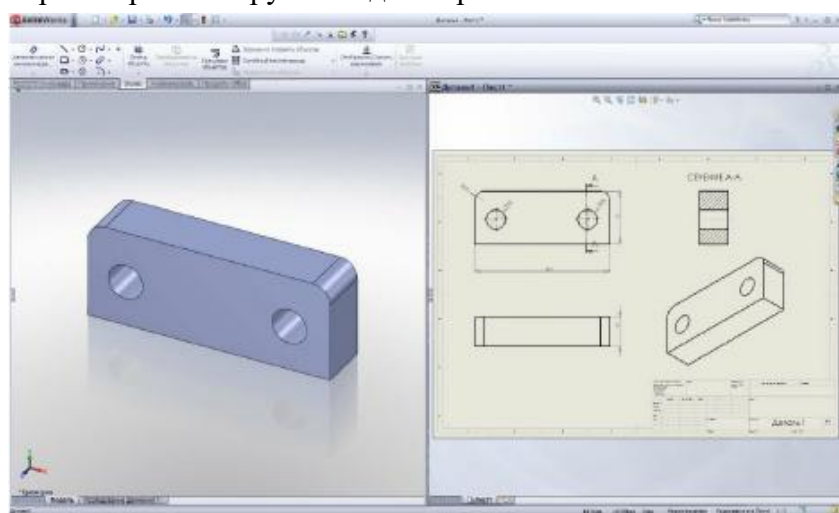


Рисунок 6.6 - Конструкторский чертеж, ассоциативно связанный с геометрической моделью

Частным случаем ассоциативного конструирования является технология ассоциативной геометрии, иногда именуемая как направленная ассоциативность (idirected associativity), - это

технология ассоциативного конструирования, которая базируется на непосредственных взаимосвязях между объектами. Простейший пример - определение параллельности двух отрезков. Отрезок А может быть определен как параллельный отрезку В. В результате при перемещении отрезка В отрезок А также изменит свое положение с сохранением ориентации по отношению к отрезку В. Собственное же положение отрезка А не может быть непосредственно изменено. Можно определить отрезки А и В как параллельные и другим способом, так что можно будет изменять положение любого из этих отрезков, удовлетворяя условиям других наложенных связей, - это случай так называемой «мягкой» ассоциативности. Преимущество использования ассоциативной геометрии - скорость.

Недостаток же заключается в том, что пользователь должен полностью определить размеры и ориентацию элемента, прежде чем приступить к созданию следующего элемента.

Объектно-ориентированное конструирование

Объектно-ориентированное конструирование (Feature-Based Modeling) основано на том, что конструктивные элементы геометрии "фьючерсы" (features) представляют собой объекты с предопределенным поведением и структурой данных. Это один из подходов ассоциативного конструирования, с помощью которого определяется поведение геометрической формы при дальнейших изменениях.

Этот подход реализован на основе определенного набора правил и атрибутов, задаваемых при выполнении базовой операции, в дополнение к уже заданным связям и ассоциативной геометрии. Базовые операции являются высокоэффективным инструментом для создания геометрической модели конструкции, инженерного анализа или изготовления. Объектно-ориентированное моделирование предоставляет в распоряжение пользователя макрофункции, ранее определенные как последовательность действий, использующих булевы операции. Например, сквозное отверстие (рисунок 6.7) - может быть представлено как булева операция вычитания и цилиндр достаточной длины, большей, чем текущая толщина детали.

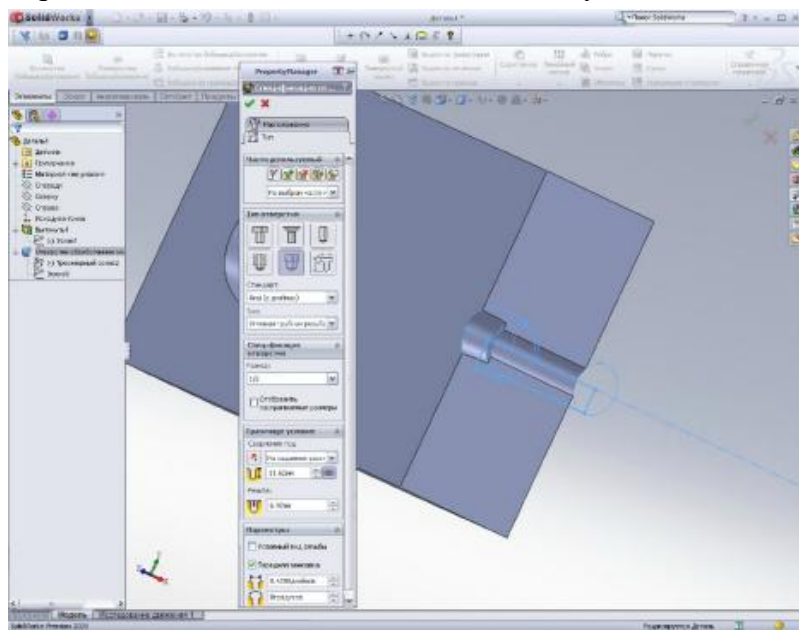


Рисунок 6.7 - Конструирование отверстия как специального объекта построения

Но если модель станет толще, то цилиндр уже не будет обладать достаточной длиной и отверстие превратится в «глухое». Однако под сквозным отверстием понимается дополнительное правило, которое определяет сквозной проход в указанном месте через тело модели, независимо от того, изменилась форма модели или нет. Базовые операции также могут иметь и дополнительные атрибуты, которые используются в других приложениях, таких как анализ и изготовление.

Обязательные требования к базовым операциям при объектно-ориентированном моделировании:

1) используемая базовая операция должна быть полностью определена. После выполнения базовой операции ее топология должна сохраняться и распознаваться как базовая операция (отверстие, паз и т. д.), а также предоставлять возможность изменения определяющих ее геометрических параметров (диаметр, глубина, и т. д.);

2) определение базовой операции должно включать в себя правила, определяющие поведение геометрической формы, а также средства контроля за соблюдением этих правил после выполнения базовой операции. Например, сквозное отверстие должно оставаться таковым, в то время как форма модели подвергается изменению;

3) для повышения эффективности процесса параллельной разработки приложения для инженерного анализа и изготовления должны иметь доступ к описанию объекта, не требуя при этом от пользователя информации об объекте, использованной ранее при выполнении базовой операции.

Уже существующие типы конструктивных элементов могут быть использованы для создания новых типов путем наследования всех свойств исходных объектов и добавления новых атрибутов и поведения. Обязательным компонентом объектно-ориентированного конструирования являются механизмы создания конструктивного элемента и его обновления путем изменения данных каждого элемента. Запуск механизма обновления при изменении данных автоматически инициирует операцию его создания, а так как эти механизмы наследуются всеми конструктивными элементами от базового типа, обеспечивается совместимость структур данных для всего набора элементов. Конструктивные элементы включены в общий цикл обновления, таким образом, любое изменение данных приводит к автоматическому обновлению модели в соответствии с правилами построения и данными для каждого элемента.

Конструирование на основе использования параметрической модели комплексного представителя типовой детали

В этом случае геометрические параметры модели могут быть связаны как переменные и можно задать зависимости других размеров от этих переменных в виде формул. Реализация геометрической модели осуществляется в среде CAD модуля путём последовательного использования языка высокого уровня (например, VBA или Си++) и табличного процессора Excel.

Каждая группа деталей имеет свои особенности конструкции, определяющие выбор метода создания объемной модели. Несмотря на это, можно перечислить основные этапы работы, общие во всех вариантах:

-анализ конструкции деталей группы (рисунок 6.8);

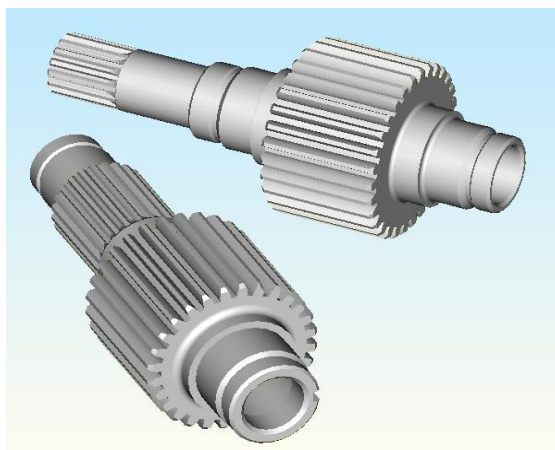


Рисунок 6.8 - Анализ конструкции деталей группы

- формирование комплексного представителя группы деталей;
- построение исходных контуров для создания объемной модели;
- простановка размеров (координация поверхностей) (рисунок 6.9);

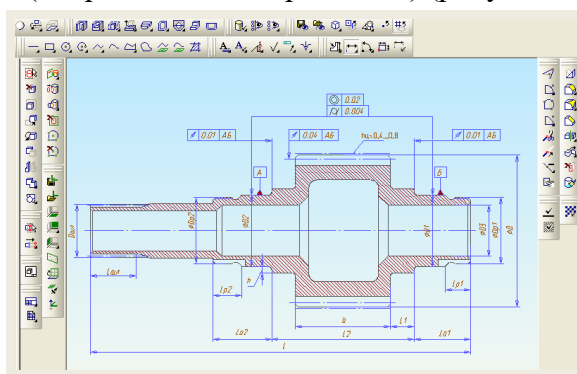


Рисунок 6.9 - Простановка размеров

- преобразование размеров в параметрические;
- построение объемной модели с помощью выполнения операций над контурами (рисунок 6.10);
- создание электронной табличной формы для внесения информации из конструкторского чертежа детали (рисунок 6.11);
- написание алгоритма расчета параметрических размеров с использованием данных с чертежа, оформление таблицы расчета;

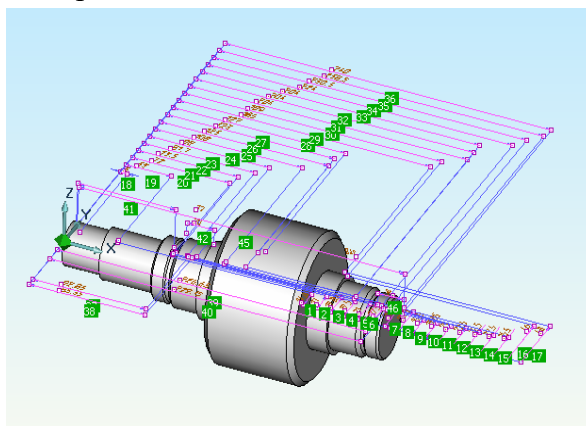


Рисунок 6.10 - Параметрическая объемная модель комплексного представителя

	A	B	C	D	E	F	G	H	I	J	K
1		Зубчатый венец			Диаметр вала под подшипники		А - в зав. от серии				
2		<i>m</i>	<i>z</i>	<i>b</i>	<i>D1</i>	<i>D2</i>	<i>h</i>	<i>L1</i>	<i>L2</i>	<i>Lo1</i>	<i>Lp1</i>
3	деталь №1	3	30	66	45	45	3	15	96	36	16
4	деталь №2	6	20	65	50	50	4	15	105	38	18
5	деталь №3	3	28	77	55	55	4	8	96	38,5	12,5
6	деталь №4	3	30	60	45	45	4	15	90	37	18
7	деталь №5	4,5	25	38	40	40	3	10	85	28	16
8	деталь №6	3,5	50	45	45	45	4	12	80	25	15
9	деталь №7	5	20	65	50	50	5	10	92	32	18

Рисунок 6.11 - Табличная форма Excel

-подключение таблицы значений параметрических размеров к параметрической модели, запись параметрического фрагмента.

Вопросы для самоконтроля

1. В чем суть параметризации?
2. Перечислите виды параметризации.
3. В чём заключается табличная параметризация?
3. Перечислите особенности иерархической параметризации.
4. Поясните процесс создания параметрической модели с использованием вариационной (размерной) параметризации.
5. Поясните алгоритм создания параметрической модели методом "эвристической" параметризации.
6. В чём суть геометрической параметризации? Порядок создания параметрической модели.
7. Поясните различия между иерархической параметризацией и вариационной (размерной) параметризацией.
8. Поясните различия между размерной и геометрической параметризацией.
9. Для чего используется ассоциативная параметризация?
10. Приведите пример конструирования на основе использования технологии ассоциативной геометрии.
11. В чём преимущества и недостатки использования ассоциативной геометрии?
12. В чем суть метода объектно-ориентированного конструирования?
13. Перечислите обязательные требования к базовым операциям при объектно-ориентированном моделировании.
14. Какими механизмами осуществляется изменение модели при изменении данных входящего в нее конструктивного элемента?
15. Поясните суть конструирования на основе параметризации комплексного представителя.
16. Перечислите основные этапы создания параметрической модели комплексного представителя группы деталей.

Лекция 7

Прямое моделирование

Когда в 1985г. Семен Гейзберг, эмигрант из СССР, бывший доцент Ленинградского университета, основал в США компанию Parametric Technology Corporation (PTC) для разработки машиностроительной CAD (MCAD) - системы Pro/Engineer, он не предполагал, что положенная в ее основу технология параметрического моделирования на основе конструктивных элементов (*parametric feature-based modeling*) будет доминировать на протяжении четверти века и что все ведущие MCAD-системы (CATIA, NX, а также SolidWorks, Inventor и Solid Edge) станут идейными наследниками Pro/Engineer. Мог ли он догадаться, что добившись небывалого коммерческого успеха, PTC неожиданно для всех примет решение принести в жертву святая святых - сам бренд Pro/Engineer?

Аналитики небезосновательно полагают, что это решение PTC прямо связано с недавним поглощением компании CoCreate, являющейся пионером альтернативного подхода - прямого моделирования. Новый бренд (Creo) и новые приложения, разделяющие общий формат данных (Creo Parametric и Creo Direct), теперь отражают видение PTC современного состояния MCAD-рынка - доминирующей технологии на нем больше нет, а значит и «забронзовевший» бренд (Pro/Engineer) уже не нужен.

Сами названия своих новых приложений PTC подчеркнула наличие двух подходов к твердотельному моделированию - параметрического и прямого, у каждого из которых имеется своя ниша применения. В появлении второго подхода просматривается прямая заслуга CoCreate.

Мало кто слышал о компании CoCreate до того, как в 2007 г. ее приобрела корпорация PTC. К тому времени CoCreate обслуживала потребности пяти тысяч клиентов (среди которых фигурировали такие громкие имена как Fujitsu, HP, Liebherr, NEC, Panasonic, Epson). Компания CoCreate в то время имела за плечами славную историю, уходящую корнями в Отдел механического проектирования (Mechanical Design Division) корпорации Hewlett-Packard. Именно в этом отделе в 1992 г. была разработана система трехмерного моделирования SolidDesigner. В отличие от других известных MCAD-систем того времени, SolidDesigner (рисунок 7.1) опиралась на технологию динамического моделирования - как альтернативу моделированию на основе истории.

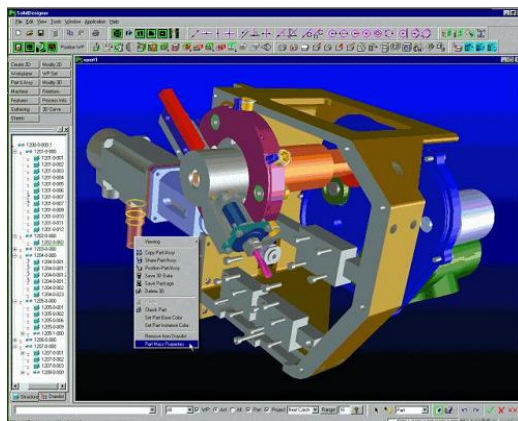


Рисунок 7.1 - MCAD-система HP Precision Engineering SolidDesigner — пионер рынка прямого моделирования

Здесь необходимо сделать технологическое отступление, чтобы внести терминологическую ясность, т.к. ранее мы уже использовали четыре термина, относящихся к геометрическому твердотельному моделированию: параметрическое моделирование, прямое моделирование, динамическое моделирование и моделирование на основе истории.

Под *параметрической моделью* логично понимать геометрическую модель с параметрами, изменяя значения которых, можно получать разные варианты модели. Параметрическое моделирование - это фундаментальная концепция САПР, позволяющая существенно сократить затраты на внесение изменений в проект, создание новых модификаций изделий и т.п.

Однако, в силу того, что первые реализации параметрических моделей были основаны на истории построения геометрии, между этими двумя совершенно разными понятиями возникла устойчивая связь. Действительно, *историю построения модели* (т.е. последовательность операций, которые были использованы для создания геометрической формы тела с нуля) легко превратить в параметрическую модель, если с каждой операцией ассоциировать набор параметров. Координаты и типы элементов плоского контура, высота его «смещения», диаметр и форма отверстия являются примерами таких параметров. Изменив их значения и заново «проиграв» (регенерировав) историю построения, можно получить другую геометрию. Параметрические конструктивные элементы Семена Гейзберга в системе Pro/Engineer образовывали дерево, которое автоматически строилось в соответствии с историей создания модели и отражало взаимосвязи между элементами. Позднее такой способ был повторен почти во всех MCAD-системах.

Данный способ параметризации - при всей его простоте и универсальности - имеет серьезные недостатки. Ключевой из них - сложность и непрозрачность для пользователя. Чтобы изменить геометрическую форму, требуется найти в дереве построения нужный конструктивный элемент и понять, значение какого параметра необходимо изменить. При таком подходе не работает фундаментальный принцип пользовательского интерфейса WYSIWYG (*What You See Is What You Get* - вы видите ровно то, что получаете в результате), редактируется текстовое или численное значение параметра элемента, а в результате меняется геометрическая форма тела.

Хорошо известно, что для моделей со сложной геометрией история построения может быть весьма длинной, а процесс ее регенерации может занять длительное время, поставив пользователя MCAD в положение ожидающего и заставив его напрасно тратить свое время и нервные клетки. Например, медленная регенерация дерева построения в системе SolidWorks вынуждает пользователей прибегать к ухищрениям - типа «заморозки» регенерации отдельных элементов - с целью ускорения общего процесса.

Другой известный недостаток параметризации на основе истории состоит в том, что решение о том, какие параметры модели можно менять, принимается в процессе ее создания. Если потом вам вдруг потребуется поменять какой-то параметр, который отсутствует в дереве построения, то решение для вас будет непростым - либо перестроить модель с нуля, либо применять сложные оптимизационные алгоритмы, которые путем варьирования значений определяющих параметров пытаются подобрать желаемое значение требуемого параметра.

Этот недостаток отражает общую проблему *процедурного подхода* к параметризации, разновидностью которого является метод регенерации истории построения. Процедурный

подход предполагает, что вы заранее делите все параметры модели на входные и выходные. Изменять можно только значения параметров первой группы, а значения выходных рассчитываются в соответствии с predetermined процедурами, формулами, историей построения и т.п.

Наконец, еще одной ключевой проблемой параметризации на основе истории построений является невозможность применения этой технологии при работе с разнородными (*multi-CAD*) и унаследованными (*legacy*) данными. Дело в том, что при трансляции модели из одного формата в другой история построения обычно теряется - транслируется только сама геометрия, которая в таком случае называется «немой» (*dumb*). Некоторые дорогостоящие трансляторы способны конвертировать конструктивные элементы из одной системы в другую, но они не являются панацеей, т.к. номенклатура конструктивных элементов в каждой системе своя, и трансляция один-в-один невозможна в принципе. То же самое можно сказать и о методах автоматического распознавания конструктивных элементов (*automated feature recognition*) в «немой» геометрии: они работают лишь в простейших случаях, и общей проблемы не решают.

Все отмеченные выше недостатки параметризации на основе истории построений были успешно преодолены в системе динамического моделирования SolidDesigner, которая впервые дала пользователю средства прямой манипуляции элементами геометрической модели в трехмерном пространстве. Это стало возможным в первую очередь благодаря ядру твердотельного моделирования ACIS, которое первым лицензировала у Spatial Tehnology в 1989 г. как раз HP.

Ядро ACIS моделирует геометрию твердого тела посредством *граничного представления* (BRep) и реализует булевы операции между телами (которые служат основой для процедурных конструктивных элементов). Булевы операции глобальны в том смысле, что объединение, пересечение или разность двух тел требует трудоемких действий над их полными граничными структурами (собственно, в этом-то и кроется проблема с производительностью параметрических систем на основе истории построения). Однако, Spatial Tehnology в своем ядре реализовала также так называемые *локальные операции*, для выполнения которых достаточно работать лишь с некоторой окрестностью границы тела (рисунок 7.2).

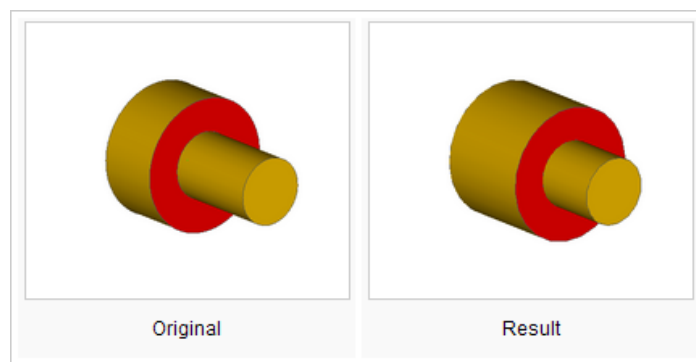


Рисунок 7.2 - Локальные операции в ACIS — основа для эффективной реализации прямого моделирования

Во многих случаях локальные операции не требуют изменения топологии модели (числа и связности ее граней, ребер и вершин), а значит, могут выполняться весьма эффективно.

Динамическое моделирование позволяет пользователю выбрать в модели одну или несколько граней и перенести или повернуть их. Система SolidDesigner помогала пользователю автоматизировать выбор смежных граней, образующих один конструктивный элемент (карман, выступ или отверстие). Система позволяла также скопировать конструктивный элемент с одной грани тела на другую.

Параметрический контроль над геометрией осуществлялся с помощью, так называемых, *управляющих значений*. Пользователь мог задать в модели одну или несколько *3D-меток*, специфицирующих требуемые расстояния и углы между гранями, а система автоматически выполняла соответствующие локальные операции, чтобы удовлетворить эти метки по очереди.

В 1996 г. HP под напором своих заказчиков, многие из которых разрабатывали собственные MCAD-системы, выделила свой отдел механического проектирования в отдельную компанию CoCreate, в 2000 продала ее инвестиционным фондам, а семь лет спустя она попала под контроль PTC (прямой наследник продукта SolidDesigner теперь известен под именем Creo Elements/Direct, а наработанные идеи реализуются в новом продукте Creo Direct, являющимся - как и Creo Parametric - ничем иным, как реинкарнацией Pro/Engineer).

Подход, предложенный компанией CoCreate, получил название **прямое моделирование** (термин динамическое моделирование так и не прижился). Прямое моделирование геометрии позволяет модифицировать ее независимо от истории построения. При этом не стоит путать способ моделирования геометрии (с деревом построения или без такового) и способ ее редактирования.

Прямое редактирование - это и есть перенос/вращение/копирование/удаление одной или нескольких граней тела (например, образующих один конструктивный элемент). А вот использование управляющих размеров или конструктивных элементов - это уже косвенное редактирование. Прямое редактирование типично для систем прямого моделирования, косвенное - для систем на основе истории построения. Однако в том же SolidDesigner присутствовали оба вида редактирования.

А разработчики IRONCAD впервые показали прямое редактирование в рамках моделирования на основе истории построения.

Повторное открытие прямого моделирования

Все изменилось в 2007 г. В декабре того года PTC объявила о своей сделке с CoCreate, а в апреле вышла первая версия абсолютно новой системы прямого моделирования SpaceClaim (рисунок 7.3). Одноименная компания, созданная бывшими сотрудниками PTC Блейком Куртером и Дэвидом Тейлором, громко прозвучала на рынке, когда ее директором был назначен легенда отрасли Майкл Пейн. Сооснователь PTC и SolidWorks, он помог новой компании привлечь внимание серьезных инвесторов и набрать солидный портфель клиентов. Настоящим успехом стало заключение соглашений с крупнейшим в мире производителем металлорежущих станков TRUMPF и ведущим поставщиком ПО для инженерного анализа ANSYS. Тем самым SpaceClaim четко обозначила две ниши, где прямое моделирование может приносить существенную пользу: подготовка геометрических моделей для CAM и CAE. Третьей нишей стало концептуальное проектирование. В принципе, SpaceClaim не предложил ничего нового, чего не было бы в SolidDesigner: то же ядро (ACIS), та же концепция «умного

выбора» граней (smart selection), те же операции прямого и косвенного моделирования. Однако, поскольку система была с нуля разработана для платформы Windows, она выглядела не в пример современнее CoCreate, а маркетингового шума произвела больше в разы: каждый год компания SpaceClaim рапортовала о трехкратном росте клиентской базы (не раскрывая, впрочем, абсолютных показателей).

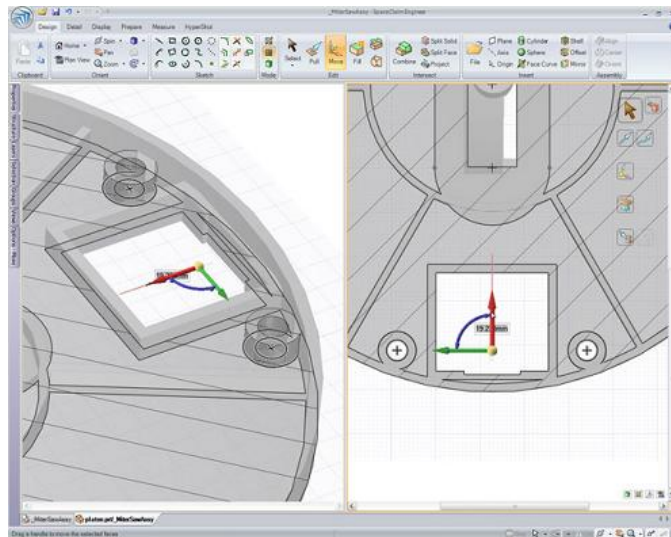


Рисунок 7.3 - Катализатор современного прямого моделирования — система SpaceClaim

Год спустя Siemens PLM Software объявила о своей синхронной технологии (признавшись, что она разрабатывалась еще в те времена, когда компания носила имя Unigraphics).

Синхронная технология - метод трехмерного параметрического моделирования, анонсированный компанией Siemens PLM Software в 2008 г. Синхронная технология комбинирует возможности параметрического моделирования на основе конструктивных элементов со средствами прямого редактирования элементов геометрической формы. Синхронная технология возможна благодаря использованию вариационного подхода к проектированию изделия, состоящего в связывании граничных элементов геометрической модели логическими и параметрическими ограничениями - как заданными пользователем, так и автоматически распознанными системой CAD.

Синхронная технология основана на возможностях ядра геометрического моделирования Parasolid и вариационного геометрического решателя DCM и воплощена в последних версиях продуктов Solid Edge и NX.

В 2009 г. Dassault Systemes выпустила CATIA V6 LiveShape, а Autodesk - Inventor Fusion. Это были приложения для прямого моделирования, совместимые по формату данных с системами на основе истории построения.

Поскольку LiveShape толком никто не видел (на просторах интернета нет ни одного сколь-нибудь исчерпывающего обзора возможностей этого приложения), то будем говорить далее лишь о подходах Siemens и Autodesk.

Вариационное прямое моделирование

Когда мы говорим о недостатках процедурного подхода к параметризации (который состоит в априорном разделе параметров на входные и выходные), мы забываем сказать о

давно известной альтернативе, называемой **вариационным моделированием**. В рамках вариационного моделирования параметрические связи в модели задаются декларативно - перечислением ограничений, связывающих ее элементы. При этом ограничения (в отличие от процедур, формул и истории построения) не являются направленными - они не определяют, какие из связанных параметров являются входными, а какие выходными. Тем самым ограничения легко могут образовывать циклические зависимости между параметрами (например, $x=2*y$ и $y=x/2$), а, значит, требуется специальный итеративный решатель, который возьмет и решит скопом все заданные ограничения, вычислив новые значения для всех входящих в них параметров.

Подобные решатели хорошо известны. Одними из первых коммерческих продуктов стали решатели 2D/3D DCM, разрабатываемые компанией D-Cubed (ныне часть Siemens PLM Software) с конца 1980-х г.г. Сейчас они конкурируют на рынке с компонентами LGS 2D/3D, разрабатываемыми в российской компании ЛЕДАС с 2001г.

Традиционными областями применений таких решателей были подсистемы двумерного параметрического черчения и трехмерного проектирования сборок. Однако, решатели - в комбинации с эффективным ядром твердотельного моделирования - могут работать и с BRep-структурами, открывая возможность параметрического контроля твердотельной геометрии.

В 2008 г. компания ЛЕДАС выпустила первую, ограниченную в возможностях реализацию вариационного прямого моделирования для системы Google SketchUp, в 2009 начала более продвинутую разработку для Rhino (первая коммерческая версия выпущена в апреле 2011 г. под названием RhinoWorks), а в 2010-2011 г. заключила контракты на разработку аналогичной функциональности в Bricscad и КОМПАС-3D.

Самую эффективную на сегодняшний момент реализацию технологии вариационного прямого моделирования можно наблюдать в Bricscad (рисунок 7.4), и немалая заслуга здесь принадлежит ядру ACIS - тому самому, что было положено в основу SolidDesigner и SpaceClaim.

Важной частью предложенной технологии является возможность избежать ручной спецификации большого количества ограничений, которые гарантировали бы сохранение *конструктивной концепции (design intent)* проектируемого изделия. В системах прямого моделирования эту концепцию легко потерять. Например, если вы возьметесь редактировать модель стола путем вращения верхней грани его столешницы (чтобы сделать столешницу наклонной к уровню пола), то, скорее всего, будете неприятно удивлены результатом.

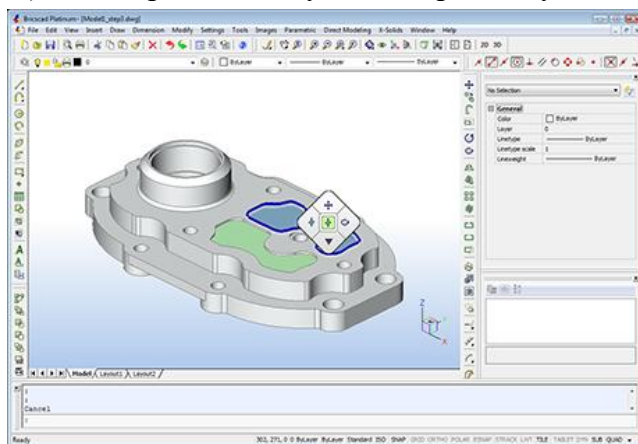


Рисунок 7.4- Вариационное прямое моделирование в Bricscad V12

Для корректной модификации такой модели требуется одновременно вращать не только верхнюю, но и нижнюю, и боковые грани столешницы, чтобы сохранить их взаимную параллельность и перпендикулярность соответственно. В системах SolidDesigner и SpaceClaim такой функционал обеспечивается за счет применения шаблонов, позволяющих выбрать сразу несколько граней в модели (а в системе TopSolid грани приходится выбирать вручную, что существенно усложняет выполнение операций прямого редактирования). Однако мультिवыбор для сложных моделей может оказаться неэффективным и несравнимым по мощи с распознаванием геометрических ограничений.

Вариационное прямое моделирование - это комбинация лучшего: параметрического моделирования на основе истории построений и «чистого» прямого моделирования (таблица 1):

Таблица 1

Зависимость возможностей от способов геометрического моделирования

	На основе истории построения	«Чистое» прямое моделирование	Вариационное прямое моделирование
Легкость освоения и использования	Нет	Да	Да
Работа в режиме WYSIWYG	Нет	Да	Да
Время отклика системы при внесении изменений	Медленно	Быстро	Быстро
Возможность задания конструктивной концепции	Да, с помощью конструктивных элементов	Нет	Да, с помощью геометрических и размерных ограничений
Распознавание конструктивной концепции	Ограниченное простыми элементами	Нет	Полное
Прямое редактирование	Ограниченное	Полное	Полное, с сохранением конструктивной концепции
Параметрическое редактирование	Ограниченное деревом построения	Частичное	Полное
Редактирование импортированной геометрии	Нет	Да	Да

Синхронная технология

То, что компания Siemens PLM Software явила миру под названием синхронной технологии, на самом деле было известно специалистам и раньше. В книге «Parametric and Feature-Based CAD/CAM» Джамии Шах и Марти Мянтыля описывают два подхода к определению конструктивных элементов - *процедурный* и *декларативный*.

В рамках *процедурного* подхода задается процедура построения конструктивного элемента - эта же процедура работает и при изменении пользователем значений его

параметров - элемент просто строится заново с помощью булевых операций внутри геометрического ядра.

Декларативный подход предполагает отдельные спецификации для создания и редактирования элемента. Если создание по-прежнему выполняется с помощью глобальных функций ядра, то редактирование происходит путем локального изменения границы тела после решения геометрических и размерных ограничений, определяющих конструктивный элемент.

Например, элемент «вытянутый профиль» можно определить заданием ограничений перпендикулярности между плоскостью профиля и боковыми гранями, образованными в результате его вытягивания, а также ограничением расстояния между плоскостью профиля и вытянутой гранью. Такая декларативная спецификация конструктивных элементов позволяет применить к модели операции прямого редактирования, т.е. трансформации граней тела. В этом случае каждая трансформация выполняется динамически с одновременным удовлетворением всех определяющих модель ограничений. В результате после каждой трансформации все ограничения остаются удовлетворенными, а это значит, что элементы модели сохранили свою конструктивную концепцию.

Комбинируя возможности ядра геометрического моделирования Parasolid и решателя геометрических ограничений 3D DCM, специалисты компании Siemens PLM Software реализовали в Solid Edge (рисунок 7.5) так называемые синхронные (декларативно заданные с помощью ограничений) конструктивные элементы, предоставив своим пользователям выбор: либо сразу проектировать свои модели с использованием синхронных элементов, или строить гибридные модели с деревом построения, включающим как классические процедурные (*упорядоченные* в терминологии Siemens), так и новые синхронные элементы.

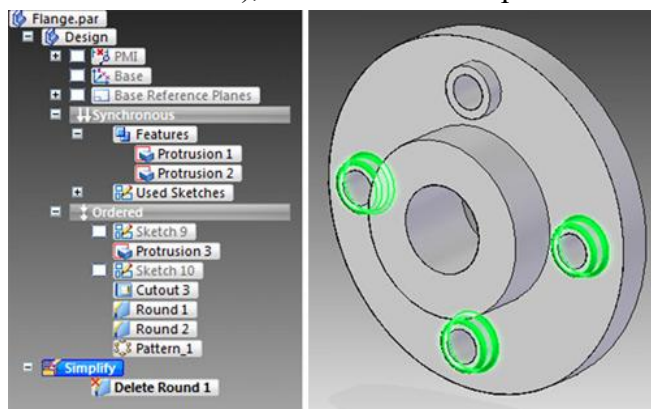


Рисунок 7.5 - Синхронная технология в Solid Edge

Комбинация прямого моделирования с деревом построения

Компании Autodesk и PTC предложили своим пользователям другой подход, состоящий в том, что прямое моделирование геометрии осуществляется в отдельных приложениях (Inventor Fusion и Creo Direct соответственно), где пользователи применяют к модели операции прямого и косвенного (с помощью управляющих размеров) редактирования, не имея доступа к дереву ее построения. Но затем та же самая модель может быть загружена в «классическое» приложение на основе истории построения (Inventor и Creo Parametric), где изменения, сделанные с ней в системе прямого моделирования, могут быть проинтегрированы в ее дерево построения.

PTC применяет в этом месте довольно бесхитростный подход, дописывая псевдо-элементы типа MoveFace и EditRound к концу дерева построения, а Autodesk реализовала довольно хитроумный менеджер изменений, который способен перевести некоторые операции прямого редактирования в изменение параметров конструктивных элементов (но в общем случае скатываясь к тому же MoveFace).

Интересно, что Autodesk в рамках Inventor Fusion и Autodesk 123D реализовала еще и простенький решатель ограничений, что сделало эти приложения немного похожими на разрабатываемые компанией ЛЕДАС— за исключением того, что в продуктах Autodesk отсутствует модуль автоматического распознавания ограничений, а сама номенклатура ограничений существенно беднее. Однако, не вызывает сомнений, что этот функционал будет нарастать по мере выпуска новых версий.

Редактирование импортированной геометрии

Подходы компаний Siemens, Autodesk и PTC нацелены прежде всего на то, чтобы дать пользователям их «классических» приложений возможности прямого моделирования.

Вернемся, однако, к еще одной проблеме моделирования на основе истории построений: работе с разнородными и унаследованными данными. Необходимость работы с такими данными не подлежит никакому сомнению: согласно недавнему опросу 82% опрошенных проектных подразделений используют в своей работе три, и более форматов данных CAD, а 42% - более пяти. Причины тому (рисунок 7.6) вполне объективны: необходимость тесной кооперации со смежниками и OEM-поставщиками, долгие жизненные циклы проектируемых изделий (превосходящие жизненные циклы ПО), исторические причины развития бизнеса.



Рисунок 7.6 - Причины, вынуждающие проектировщиков работать с разнородными данными

А главной проблемой при работе с разнородными данными сами опрошенные назвали потерю интеллектуальности, присущей исходной модели в той системе, где она была создана. В «чужой» системе теряется история построения модели, теряются параметрические связи между ее элементами. Именно в этом месте родился термин «немая» геометрия, который метко объясняет суть проблемы: геометрия есть, но «рассказать» о своей конструктивной концепции она ничего не может.

Системы прямого моделирования научили «немую» геометрию «говорить», но разговор получается неполноценным: как гарантировать, что в процессе редактирования не потеряется

заложенная в модель конструктивная концепция? Ответ - в методах автоматического распознавания геометрических ограничений в граничной модели. Если две плоских грани тела расположены параллельно или перпендикулярно друг другу, такое положение скорее всего не случайно. Видимо, они должны оставаться параллельными (перпендикулярными) при любой операции редактирования модели (за исключением того случая, когда пользователь захочет явно задать другой угол между этими гранями). Аналогично можно сказать про отверстия равного диаметра, про карманы одинаковой глубины, про различные виды симметрии модели и т.п. Все это можно распознать и сохранить.

Распознавание ограничений не в пример легче распознавания конструктивных элементов или реконструкции дерева построения, ведь возможные геометрические соотношения легко проверить простым перебором граничных элементов. Именно в этом видится ключ к интеллектуальности систем прямого моделирования. И конечно возможности добавления к «немой» геометрии пользовательских спецификаций (будь то распознанные конструктивные элементы или ограничения) существенно помогают обогатить ее конструктивную концепцию.

Недостатки прямого моделирования

Ахиллесовой пятой всех современных систем прямого моделирования все еще остается редактирование сложных NURBS-поверхностей (рисунок 7.7). Одно дело - распознать в модели с отсутствующей историей построения плоские, цилиндрические, сферические, конические, торические грани. Другое дело - понять, что данная грань является результатом применения операций *sweep* (заметания плоского контура при движении вдоль заданной кривой), *loft* (построение тела по его плоским сечениям), *blend* (гладкого сопряжения двух поверхностей) и т.п.

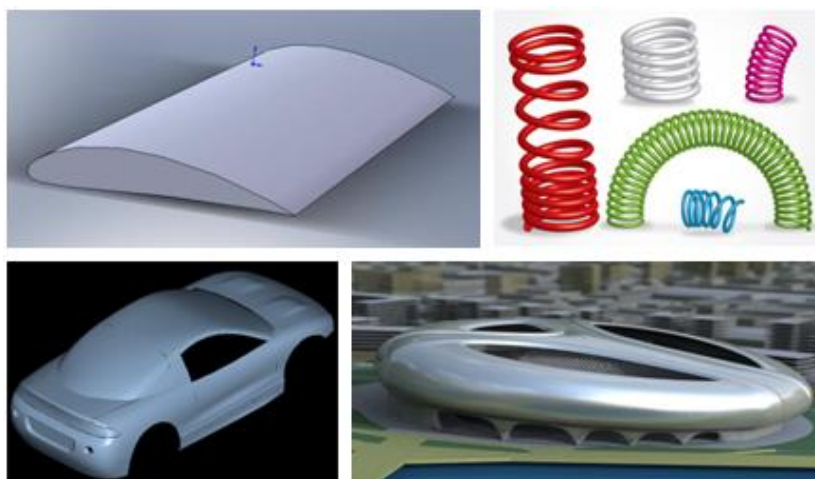


Рисунок 7.7 - Редактирование сложных NURBS-поверхностей — вызов для современных систем прямого моделирования

Вопросы для самоконтроля

1. Назовите компанию - пионера прямого моделирования.
2. В какой системе впервые использовано прямое моделирование?
3. Как ассоциативно связаны параметрическая модель с деревом построения?
4. Назовите преимущества и недостатки параметрических моделей.
5. В чём состоит процедурный подход и каков его недостаток?

6. В чём состоит суть прямого моделирования?
7. Какие операции доступны при использовании динамического моделирования?
8. В чём особенности прямого и косвенного редактирования?
9. В чём суть синхронного моделирования?
10. Объясните, что собой представляет вариационное моделирование?
11. Для чего предназначены итеративные решатели?
12. В чём состоит особенность сохранения конструктивной концепции изделия?
13. Объясните разницу в подходах к определению конструктивного элемента: процедурном и декларативном.
14. Расскажите, в каких средах геометрического моделирования реализована комбинация прямого моделирования с деревом построений.
15. Почему передаваемую из одной системы геометрического моделирования в другую геометрическую модель называют "немой"?
16. Как научить "немую" геометрию "говорить"?
17. В чём проявляется недостаток прямого моделирования?

Лекция 8

Методы проецирования. Техническое рисование.

Для построения изображений (проекций) геометрических фигур начертательная геометрия применяет метод проецирования. Получающиеся при этом чертежи называются проекционными.

Метод проекций - отображение геометрической фигуры на плоскость путем проецирования точек этой фигуры. Проецированием называется процесс построения изображения с помощью проецирующих прямых.

Существует два вида проецирования - центральное и параллельное и соответственно два вида проекций - центральные и параллельные. Построение проекций предмета сводится к построению проекций некоторого множества его точек. Поэтому изучение метода проецирования начинают с построения проекций точки.

Центральное проецирование

Для того чтобы построить проекцию некоторой точки A (рисунок 8.1), выбирается произвольная плоскость Π_1 , называемая плоскостью проекций, и точка S , не принадлежащая плоскости Π_1 , называемая центром проекций. Операция проецирования состоит в том, что через точки S и A проводится прямая до пересечения с плоскостью Π_1 . Прямая SA называется проецирующей прямой, а A_1 точка пересечения проецирующей прямой с плоскостью проекций Π_1 , - центральной проекцией точки A . На плоскости Π_1 , можно построить центральные проекции всех точек пространства, за исключением тех, которые принадлежат плоскости Π_1' , проходящей через центр проекций S и параллельной Π_1 . В этом случае проецирующие прямые оказываются параллельными плоскости Π_1 (прямая SC) и точек пересечения их с плоскостью в обычном смысле нет.

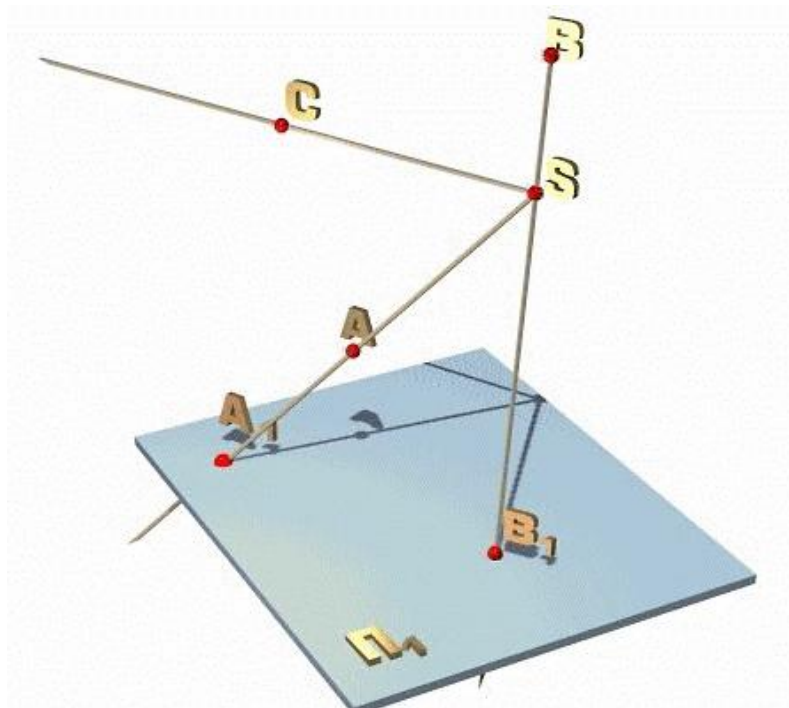


Рисунок 8.1 - Центральные проекции точек

Описанным методом центрального проецирования может быть построена проекция любой точки геометрической фигуры, а следовательно, и проекция самой фигуры. Например, центральной проекцией отрезка $[AB]$ на плоскости Π_1 (рисунок 8.2) является множество центральных проекций всех точек отрезка $[AB]$ - $[A_1B_1]$

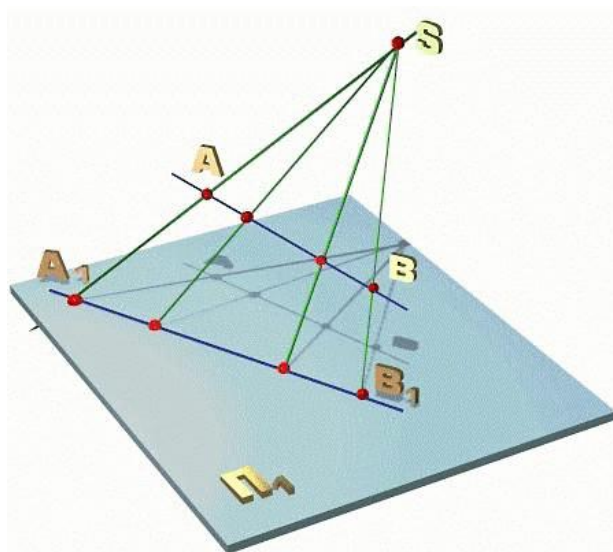


Рисунок 8.2 - Центральная проекция прямой

При центральном проецировании происходит искажение формы, размеров и некоторых других свойств предмета (рисунок 8.3).

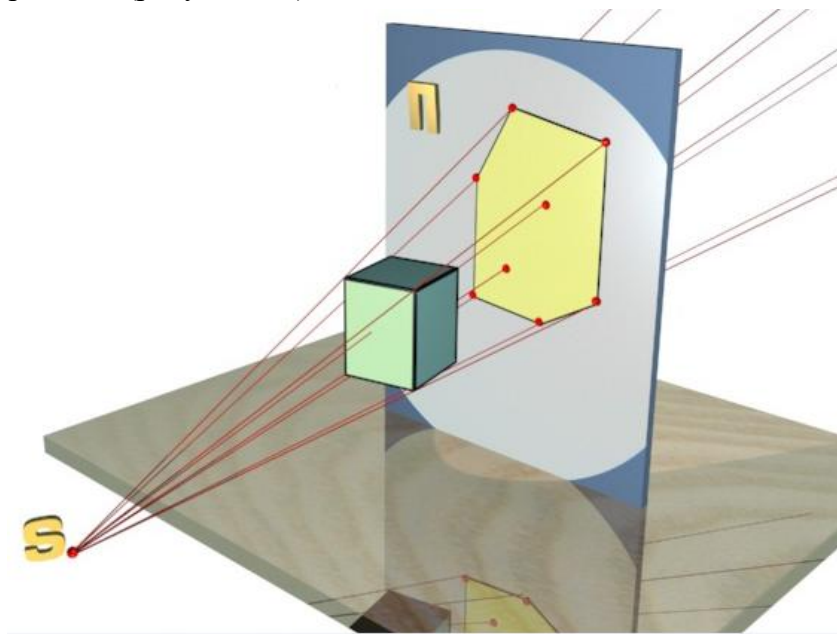


Рисунок 8.3 - Центральная проекция призмы

Вместе с тем, нетрудно заметить, что часть свойств сохраняется, например, проекция точки является точкой; проекция прямой - тоже прямая линия; если точка принадлежит прямой, то проекция точки принадлежит проекции той же прямой; точка пересечения прямых проецируется в точку пересечения их проекций. Проекция предмета, построенная методом центрального проецирования, называется перспективой.

Построение проекций заданного объекта называется прямой задачей начертательной геометрии. Нетрудно заметить, что метод центрального проецирования позволяет решать её однозначно: каждая точка имеет на плоскости Π_1 единственную проекцию, так как проецирующая прямая пересекается с плоскостью Π_1 в одной точке.

Определение формы и размеров объекта по его чертежу называется обратной задачей начертательной геометрии. Одна проекция - точки не определяет ее положения в пространстве, так как может быть проекцией любой точки, принадлежащей проецирующей прямой, то есть одно проекционный чертеж является необратимым. Для определения местоположения точки в пространстве нужно иметь как минимум две проекции точки, полученные из разных центров проекции.

Параллельное проецирование

Если за центр проекций принять бесконечно удалённую (несобственную) точку S пространства, то проецирующие прямые AA_1, BB_1, \dots будут параллельными между собой. Для их построения вместо отсутствующей на чертеже точки S задают направление проецирования s (рисунок 8.4).

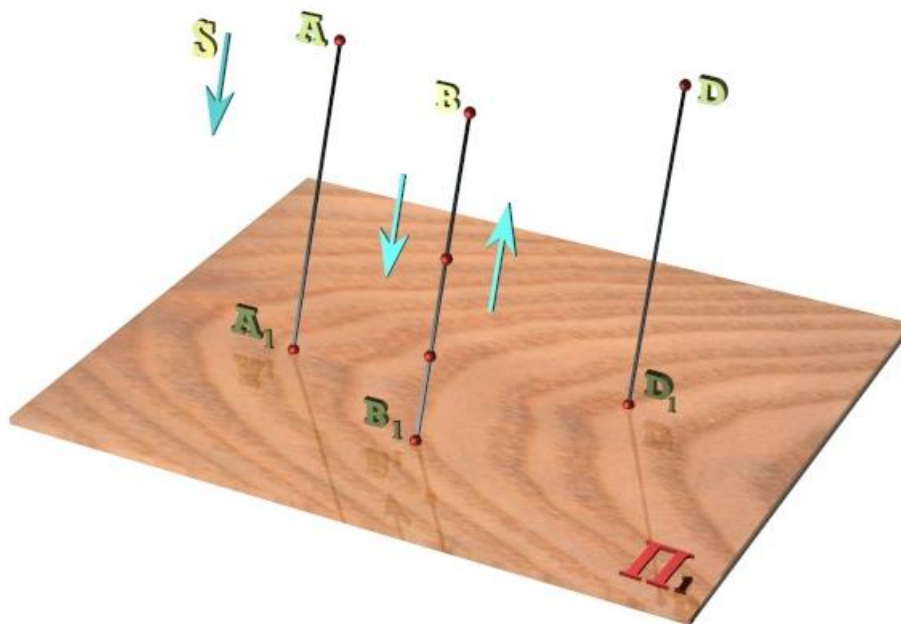


Рисунок 8.4 - Параллельные проекции точек

Такой вид проецирования называется параллельным, а точки A_1, B_1, D_1, \dots пересечения проецирующих прямых с плоскостью проекций Π_1 - параллельными проекциями точек A, B, D, \dots пространства.

При параллельном проецировании, так же как и при центральном, каждая точка пространства имеет на плоскости Π_1 одну проекцию, но эта проекция не определяет положения точки в пространстве (рисунок 8.5).

Следовательно, одно проекционный чертеж, полученный методом параллельного проецирования необратим. Метрические характеристики геометрических фигур при параллельном проецировании в общем случае не сохраняются (происходит искажение линейных и угловых величин).

В зависимости от угла, образованного направлением проецирования с плоскостью проекций различают **прямоугольное** (ортогональное) [угол равен 90°] и **косоугольное** [угол не равен 90°] параллельное проецирование.

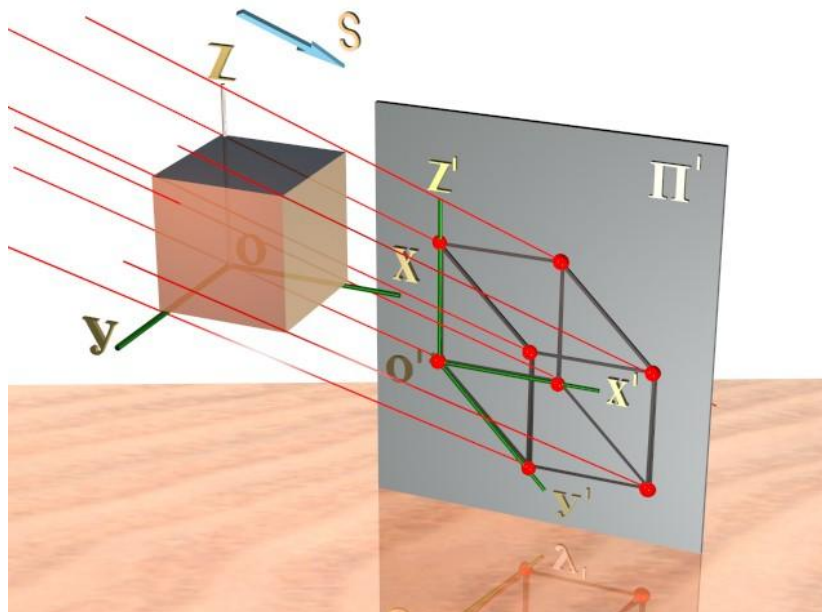


Рисунок 8.5 -Параллельная проекция призмы

Перечислим **основные инвариантные (независимые) свойства** параллельного проецирования:

1. Проекция точки на плоскость есть точка $A \rightarrow A_1$.
2. Проекция прямой в общем случае прямая $l \rightarrow l_1$. Она вырождается в точку, если прямая параллельна направлению проецирования (рисунок 8.6).
3. Если точка принадлежит линии, то проекция точки принадлежит проекции линии $A \in l \rightarrow A_1 \in l_1$.

Отсюда следует, что для построения проекции прямой необходимо построить

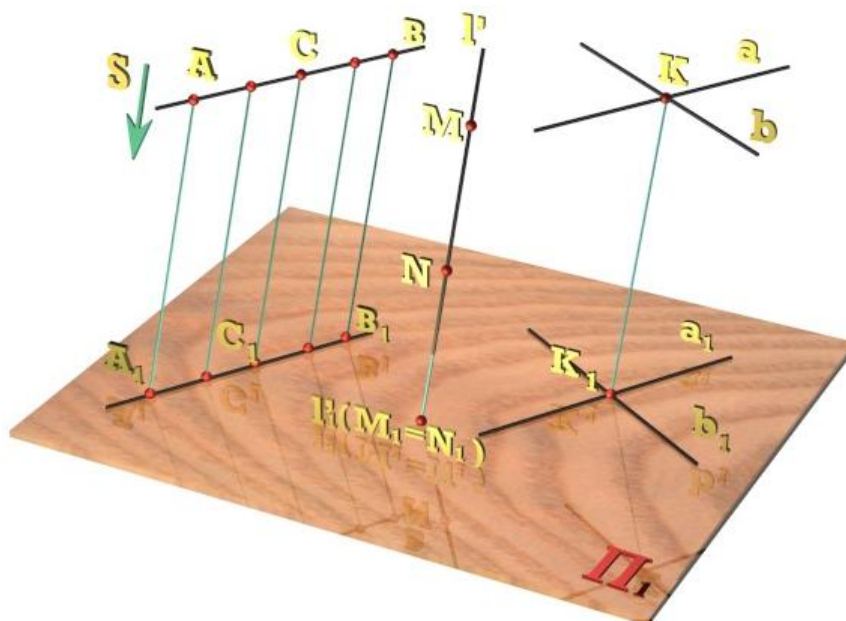


Рисунок 8.6 - Параллельные проекции точек и прямых

проекция двух принадлежащих ей точек.

4. Точка пересечения линий проецируется в точку пересечения их проекций

$$K = a \cap b \Rightarrow K_1 = a_1 \cap b_1.$$

5. Проекция параллельных прямых параллельны $l \parallel l' \Rightarrow l_1 \parallel l'_1$ (рисунок 8.7).

Следствия, вытекающие из этих свойств:

- отношение длин отрезков параллельных прямых равно отношению длин их проекций
- если точка, принадлежащая отрезку прямой, делит его в некотором отношении, то проекция точки делит проекцию отрезка в том же отношении

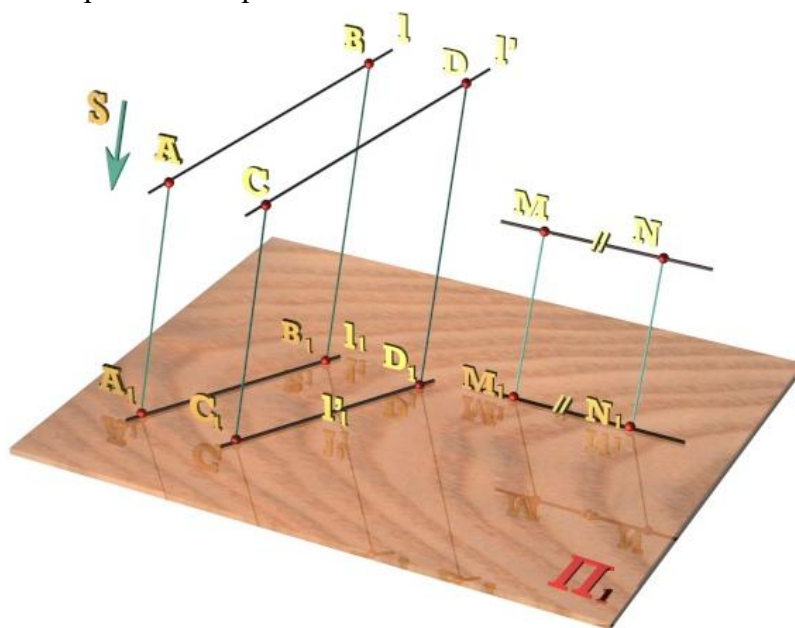


Рисунок 8.7 - Параллельные проекции параллельных прямых

6. Если геометрическая фигура Φ принадлежит плоскости, параллельной плоскости проекций, (например, Π_1), то проекция этой фигуры на плоскость Π_1 конгруэнтна самой фигуре ($\Phi \in \Sigma \wedge (\Sigma \parallel \Pi_1) \Rightarrow \Phi_1 \cong \Phi$) (рисунок 8.8).

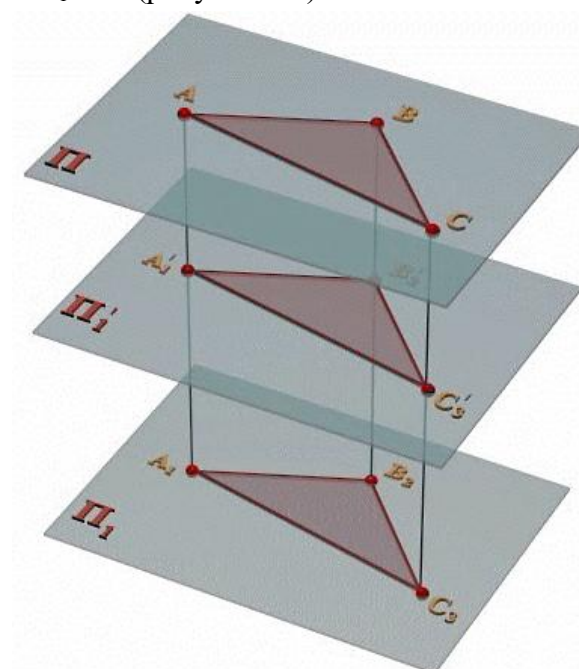


Рисунок 8.8 - Параллельные проекции плоских фигур

Проекция геометрической фигуры не изменяется при параллельном переносе плоскости проекций.

Ортогональное проецирование

Если направление проецирования перпендикулярно плоскости проекций, то параллельное проецирование называется ортогональным (прямоугольным) (рисунок 8.9).

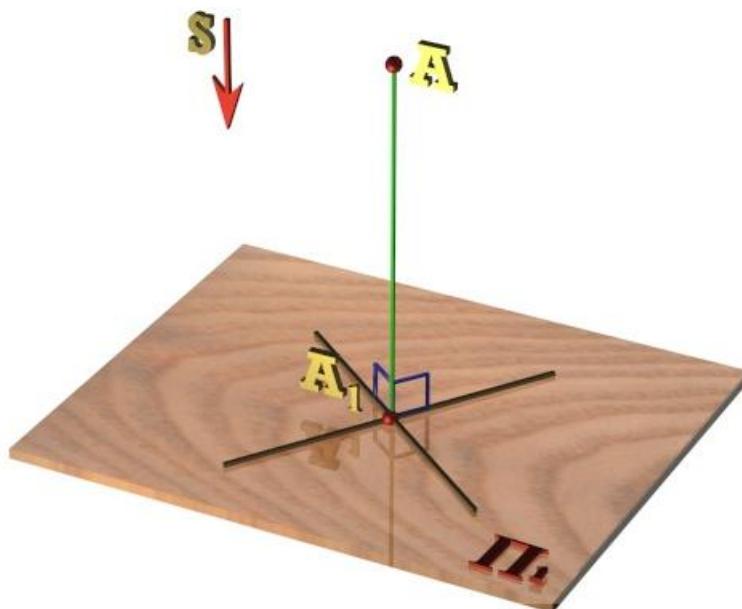


Рисунок 8.9 - Ортогональная проекция точки

В этом случае проекция A_1 точки A называется ортогональной, или прямоугольной. В противном случае проецирование называется косоугольным.

Ортогональное проецирование, являясь частным случаем параллельного, значительно упрощает построение проекций геометрических фигур и является основным при выполнении комплексных чертежей технических форм (рисунок 8.10).

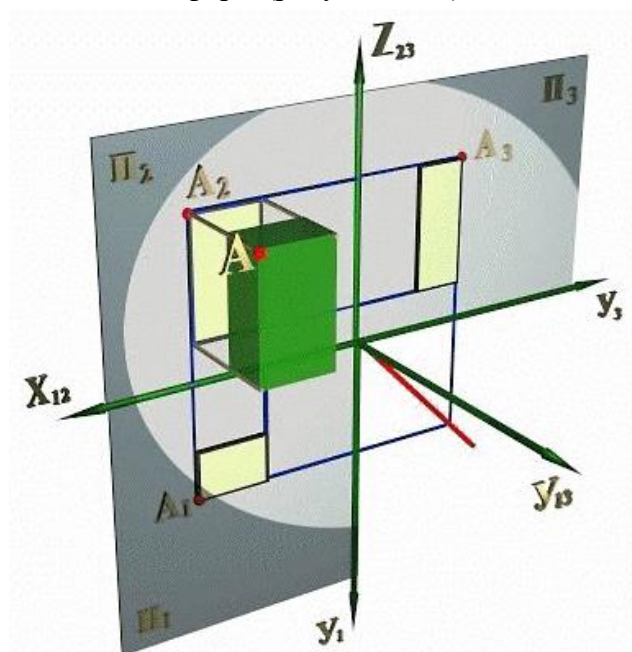


Рисунок 8.10 - Ортогональные проекции призмы

Рассмотренные в предыдущих параграфах одно проекционные чертежи геометрических фигур являются необратимыми. По ним нельзя мысленно воссоздать пространственную форму и размеры изображенного объекта. Существуют различные способы устранения этого недостатка одно проекционных чертежей в зависимости от принятого вида проецирования. Например, при центральном проецировании точку можно проецировать из двух различных центров, при параллельном - при помощи двух различных направлений, при ортогональном - на две пересекающиеся плоскости (рисунок 8.11).

Нетрудно заметить, что в каждом из этих случаев получаются две проекции A_1 , и A_1' , точки A , однозначно определяющие её положение в пространстве. Следовательно, обратимый чертеж геометрической фигуры должен содержать не менее двух проекций каждой ее точки.

В технике применяются следующие виды обратимых чертежей:

- комплексные
- аксонометрические
- перспективные
- чертежи с числовыми отметками

В дальнейшем мы будем рассматривать только комплексные чертежи.

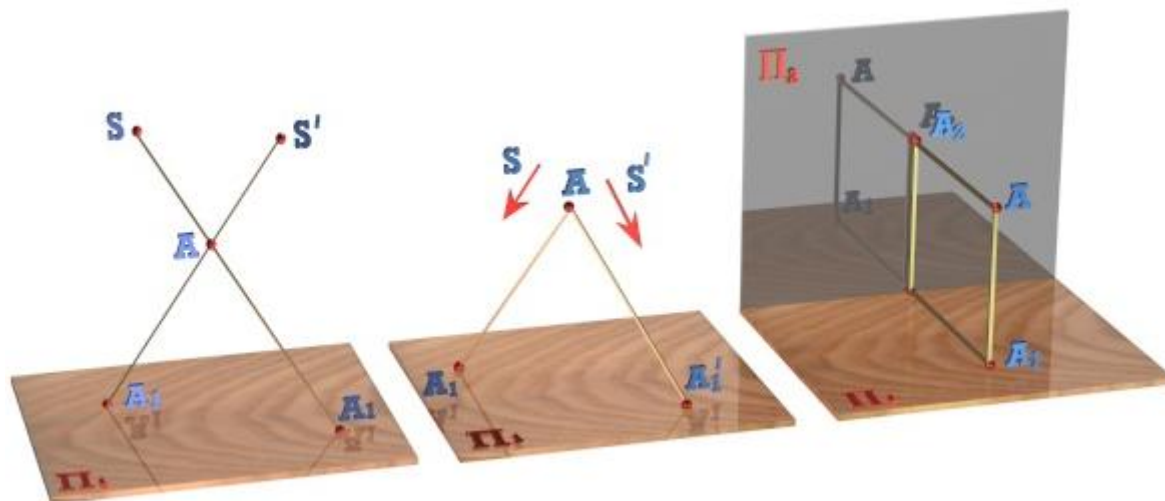


Рисунок 8.11 - Построение обратимых чертежей

Сущность аксонометрического проецирования. Виды проекций.

В ряде случаев необходимо, наряду с чертежом объекта, выполненном в ортогональных проекциях, иметь его наглядное изображение, состоящее только из одной проекции.

Способ проецирования, при котором заданная геометрическая фигура вместе с декартовой системой координат, к которой она отнесена в пространстве, параллельно проецируется на одну плоскость проекций так, что ни одна ось не проецируется в точку (а значит, сам предмет будет спроецирован в трёх измерениях), называется **аксонометрическим**, а полученное с его помощью изображение - **аксонометрической проекцией** или **аксонометрией**. Плоскость, на которую производится проецирование, называется **аксонометрической** или **картинной**.

Аксонометрическая проекция называется **прямоугольной**, если при параллельном проецировании проецирующие лучи перпендикулярны картинной плоскости ($\varphi=90^\circ$) и **косоугольной**, если лучи составляют с картинной плоскостью угол $0<\varphi<90^\circ$.

Возьмём в пространстве координатные оси с единичными отрезками на них и спроецируем на картинную плоскость P параллельно и в направлении проецирования S (т.е. с заданным углом проецирования ψ) (рисунок 8.12).

Т.к. ни одна из координатных осей не параллельна картинной плоскости, то единичные отрезки на плоскости P будут меньше единичных отрезков на декартовых осях.

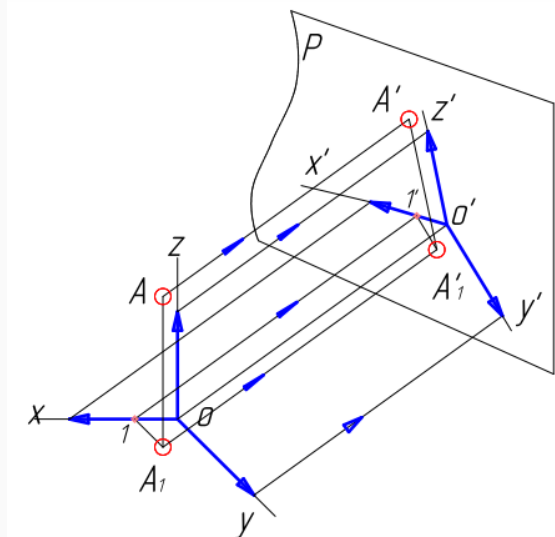


Рисунок 8.12 - Аксонометрическое проецирование

Отношение единичных отрезков на аксонометрических осях к единичным отрезкам на координатных осях называется **коэффициентом искажения** по аксонометрическим осям.

Очевидно, принимая различное взаимное расположение декартовой системы координат и картинной плоскости и задавая разные направления проецирования, можно получить множество аксонометрических проекций, отличающихся друг от друга как направлением аксонометрических осей, так и величиной коэффициента искажения вдоль этих осей.

Если коэффициенты искажения приняты различными по всем трём осям, то эта аксонометрическая проекция называется **триметрической**. Если коэффициенты искажения одинаковы по двум осям - **диметрической**. Если коэффициенты искажения равны между собой - **изометрической**.

В машиностроении наибольшее распространение получили (см. ГОСТ 2317-69) (рисунки 8.13 и 8.14):

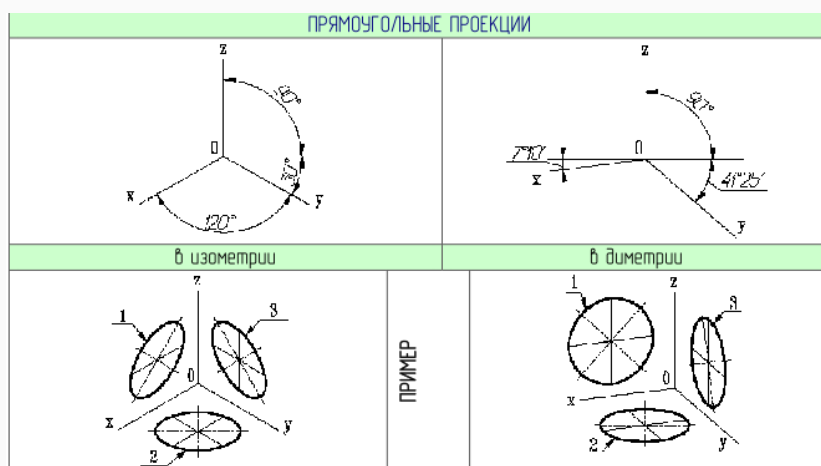


Рисунок 8.13 - Прямоугольные проекции

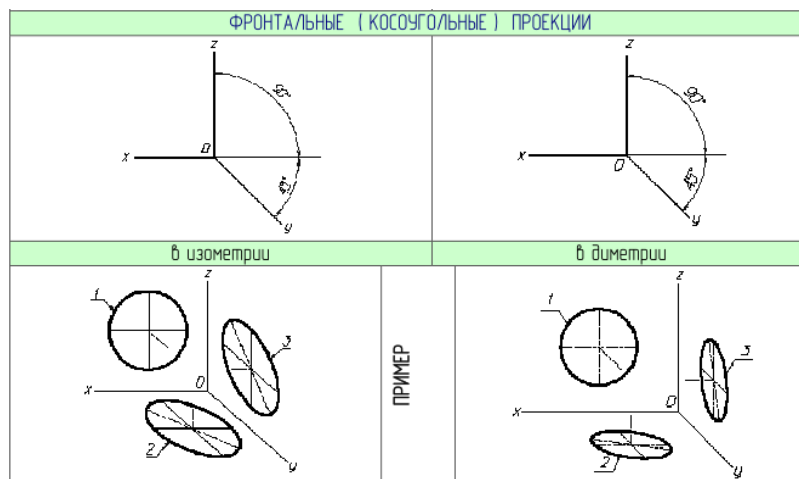


Рисунок 8.14 - Косоугольные проекции

Особенности технического рисунка

Технические рисунки чаще всего применяются в конструкторской практике. Ими пользуются для того, чтобы быстро выразить свою мысль в наиболее наглядной форме или пояснить сложный чертеж. Каждый инженер, мастер, студент технического учебного заведения должны уметь делать рисунки деталей.

Несколько предварительно выполненных рисунков позволяют выбрать лучший вариант будущей формы или конструкции предмета. Эскизы и рисунки являются первичной формой изображения предмета.

Рисунки уступают чертежу в точности, но превосходят его в наглядности.

Технический рисунок предмета выполняется по правилам аксонометрии от руки на глаз без применения чертежных инструментов.

Построение технического рисунка основано на свойствах параллельных проекций; этим он отличается от рисунка, построенного по правилам перспективы, основанным на свойствах центральных проекций.

Перед выполнением технического рисунка детали необходимо тщательно рассмотреть эту деталь, изучить ее геометрическую форму, взаимное положение и размерное соотношение его составных частей.

Для равномерного заполнения поля выбранного формата следует продумать компоновку рисунка.

Рисование геометрических тел

Детали всегда можно мысленно расчленить на основные геометрические тела – параллелепипед, призму, пирамиду, цилиндр, конус, шар.

Рисунки геометрических тел выполняют в той же последовательности, в какой составляют аксонометрические чертежи.

Для **призмы** прежде всего рисуют основание. При вертикальном расположении призмы удобнее начинать рисунок с верхнего основания, а при горизонтальном с переднего.

Нарисовав основание, из его вершин проводят ребра призмы; видимые стороны второго основания проводят параллельно сторонам первого. Если на рисунке предполагают сохранить

невидимые линии, то меняют последовательность и рисуют, начиная с оснований (рисунок 8.15).

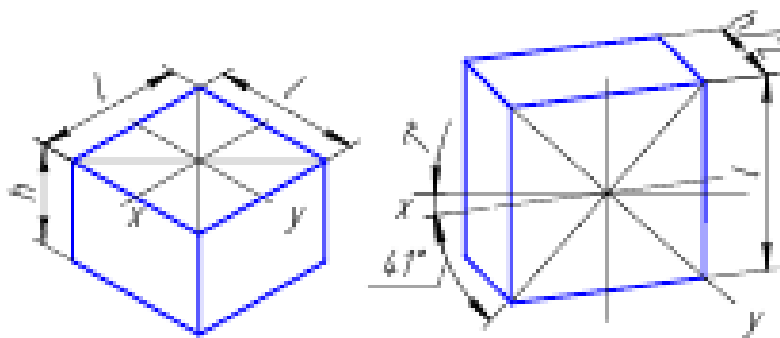


Рисунок 8.15 - Рисунок призмы

Пирамиду рисуют, начиная с основания, затем проводят линию высоты и намечают на ней вершину S, которую соединяют с вершинами основания (рисунок 8.16).

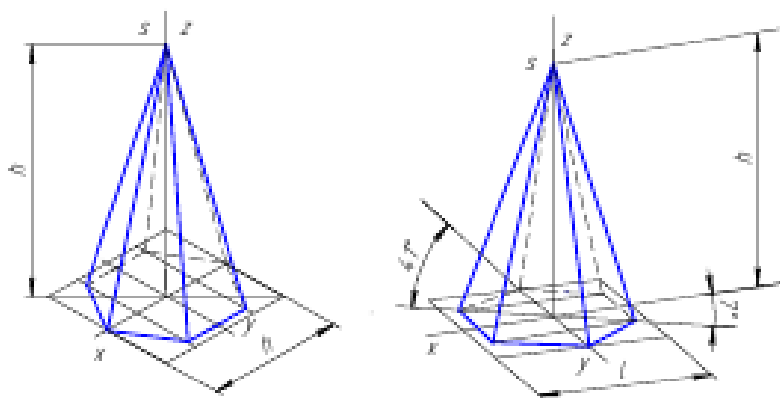


Рисунок 8.16 - Рисунок пирамиды

Вертикально расположенный **цилиндр** начинают рисовать с верхнего основания (рисунок 8.17), затем проводят касательные к эллипсу, образующие цилиндра; параллельно построенному основанию строят нижнее основание.

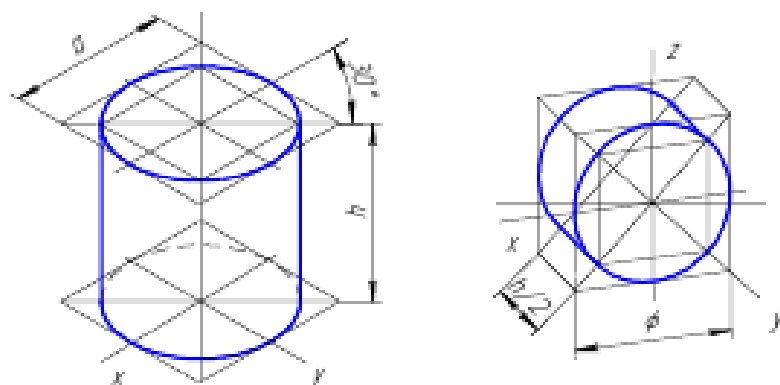


Рисунок 8.17 - Рисунок цилиндра

На рисунке 8.18 приведен рисунок **конуса**. Рисунок следует начинать с изображения основания конуса. Затем рисуют вершину конуса и крайние видимые образующие, которые касаются эллипса.

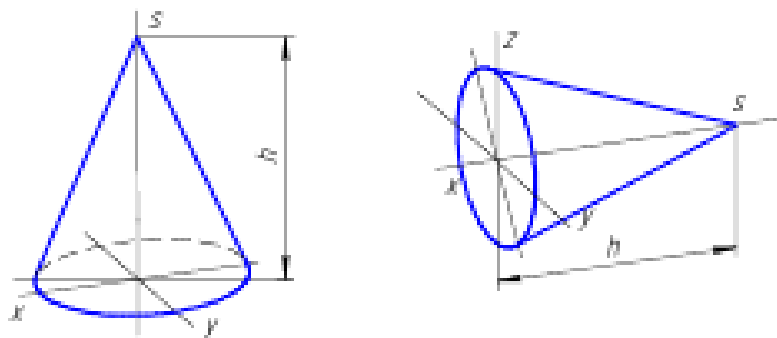


Рисунок 8.18 - Рисунок конуса

Шар в прямоугольной аксонометрии изображается окружностью (рисунок 8.19).

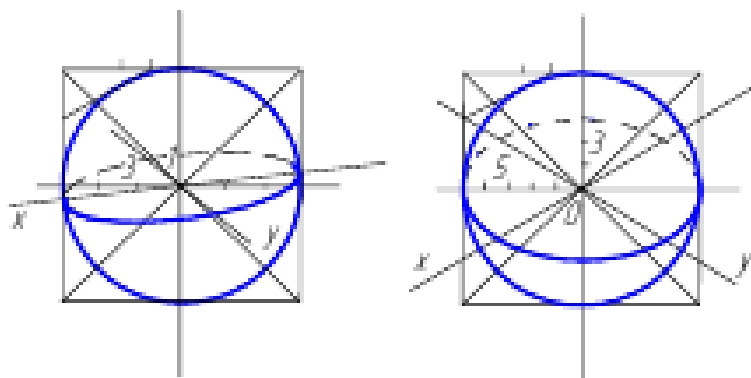


Рисунок 8.19 - Рисунок шара

Нанесение светотени

Для придания рисунку большей наглядности на нём наносят светотени. Светотень показывает распределение света на поверхности предмета, она способствует восприятию объемной формы предмета.

Светотень состоит из собственной тени, падающей тени, рефлекса, полутени и блика.

Собственная тень – тень, образующая в неосвещенной части предмета.

Падающая тень – тень, отбрасываемая самим предметом на какую-либо поверхность. Она на техническом рисунке не показывается.

Полутень – слабоосвещенные места, постепенный переход от тени к свету.

Рефлекс – высветление собственной тени отраженными лучами. На неосвещенную часть предмета падают отраженные лучи, которые ее подсвечивают, создавая рефлекс, значит рефлекс – отраженный свет на поверхности предмета.

Свет – наиболее освещенная часть предмета.

Блик – наиболее освещенное место (пятно) на поверхности предмета. В техническом рисунке блики показывают на поверхности вращения.

Наиболее освещенными частями поверхности являются те части, к которым лучи света перпендикулярны.

В техническом рисовании принято направление лучей света сверху, слева направо.

Наиболее освещенные участки поверхности оттеняют светлее, чем расположенные от света дальше (рисунок 8.20).

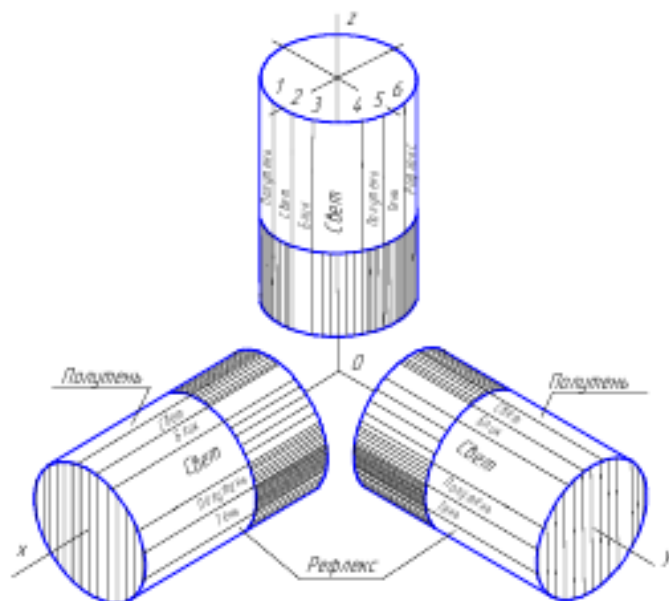


Рисунок 8.20 - Схема освещения цилиндра

Для нанесения теней на техническом рисунке применяют параллельную штриховку, шраффировку, точки (рисунок 8.21а, б, в).

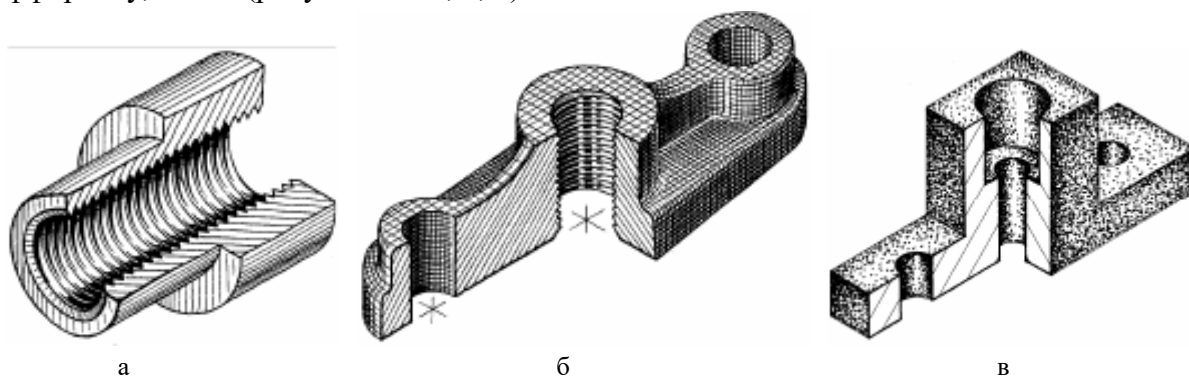


Рисунок 8.21 - Нанесение теней: а - штриховка, б - шраффировка, в - оттенение точками

Шраффировка (рисунок 8.22) - штриховка в виде сетки, её линии наносят параллельно линиям образующей и направляющей изображаемой поверхности. На рисунке цилиндра и конуса линии штриховки параллельны образующим их оснований.

На рисунке граненого тела линии шраффировки параллельны ребрам и сторонам основания. На наклонных плоскостях проводят сначала наклонные линии, затем параллельные сторонам основания.

При работе над рисунком отдельные штрихи могут получаться слишком яркими. Яркость штрихов можно уменьшить с помощью мягкой резинки. Резинку следует прикладывать как промокашку, но не растирать, иначе рисунок может получиться смазанным.

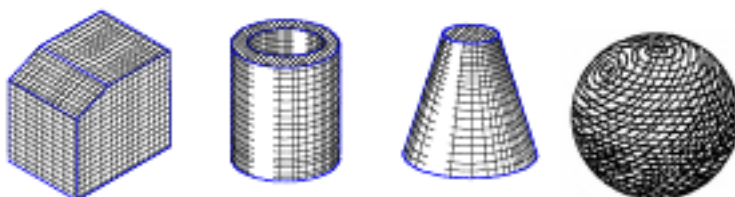


Рисунок 8.22 - Нанесение теней шраффировкой

Рисование деталей

Последовательность выполнения технического рисунка детали:

- выбирают вид аксонометрической проекции, строят аксонометрические оси, намечают положение детали относительно осей;
- определяют габаритные размеры детали, строят параллелепипед;
- расчленяют вспомогательный габаритный параллелепипед на отдельные составляющие его геометрической формы и рисуют их тонкими линиями;
- рисуют видимые элементы детали;
- наносят светотень (рисунок 8.23).

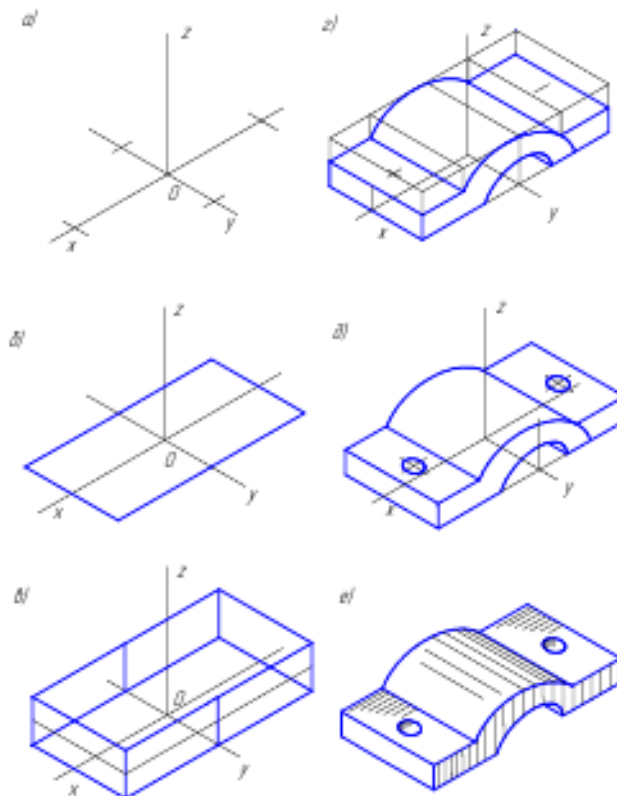


Рисунок 8.23 - Последовательность построения технического рисунка детали

Рисование сборочных единиц

Последовательность выполнения технического рисунка сборочной единицы:

- выбирают вид аксонометрической проекции, строят аксонометрические оси, намечают положение сборочной единицы относительно осей;
- выбирают на глаз соотношение габаритных размеров сборочной единицы, сопоставляют отдельные части между собой и со всей сборкой в целом;
- намечают тонкими линиями границы рисунка, т.е. определяют его композицию на листе;
- намечают положение отдельных деталей и эллипсов - изображений окружностей;
- уточняют форму каждой детали и прорисовывают их;
- после выполнения линейного рисунка всей сборочной единицы приступают к нанесению светотени.

Пример рисунка сборочной единицы (рисунок 8.24).

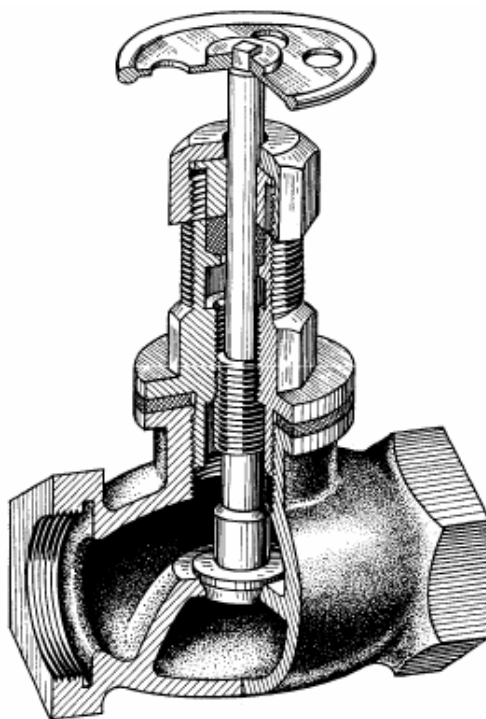


Рисунок 8.24 - Технический рисунок сборочной единицы

Выбор аксонометрических проекций

Правильный выбор аксонометрической проекции обеспечивает большую наглядность изображения и простоту построения рисунка. Под наглядностью понимают наиболее отчетливую видимость на рисунке основных частей детали и наименьшее искажение ее формы.

Простота построения рисунка связана с выбором вида аксонометрической проекции.

Выбор аксонометрической проекции зависит от формы детали. Главной задачей является обеспечение видимости всех основных элементов изображаемой детали. Отдельные части детали не должны закрывать друг друга.

В качестве примера для выбора аксонометрической проекции на рисунке 8.25 приведены аксонометрические проекции (изометрическая, диметрическая, кабинетная).

На изображении куска угольника с отверстием изометрическая проекция имеет наибольшие преимущества: на ней видно, что толщина вертикальной и горизонтальной полок одинаковая. На диметрической и кабинетной проекциях вертикальная полка кажется значительно толще горизонтальной; видно, что отверстие сквозное. На диметрической и кабинетной проекции этого не видно (рисунок 8.25, а).

На изображении стойки, преимущество имеет диметрическая проекция, она дает полное представление о форме и размерах предмета, изометрическая проекция показывает только две стороны предмета. В кабинетной проекции квадратная форма граней кажется вытянутой, торец стержня изображен искаженным (перекошенный эллипс) (рисунок 8.25, б).

На изображении фланца преимущества за кабинетной проекцией: на ней видно, что у патрубка сквозное отверстия и толщина стенок трубы везде одинаковая, на изометрической проекции она кажется неодинаковой.

Кабинетная проекция просто выполняется, т.к. все торцевые очертания изображены окружностями, а не эллипсами, как в изометрической и диметрической проекциях (рисунок 8.25,в).

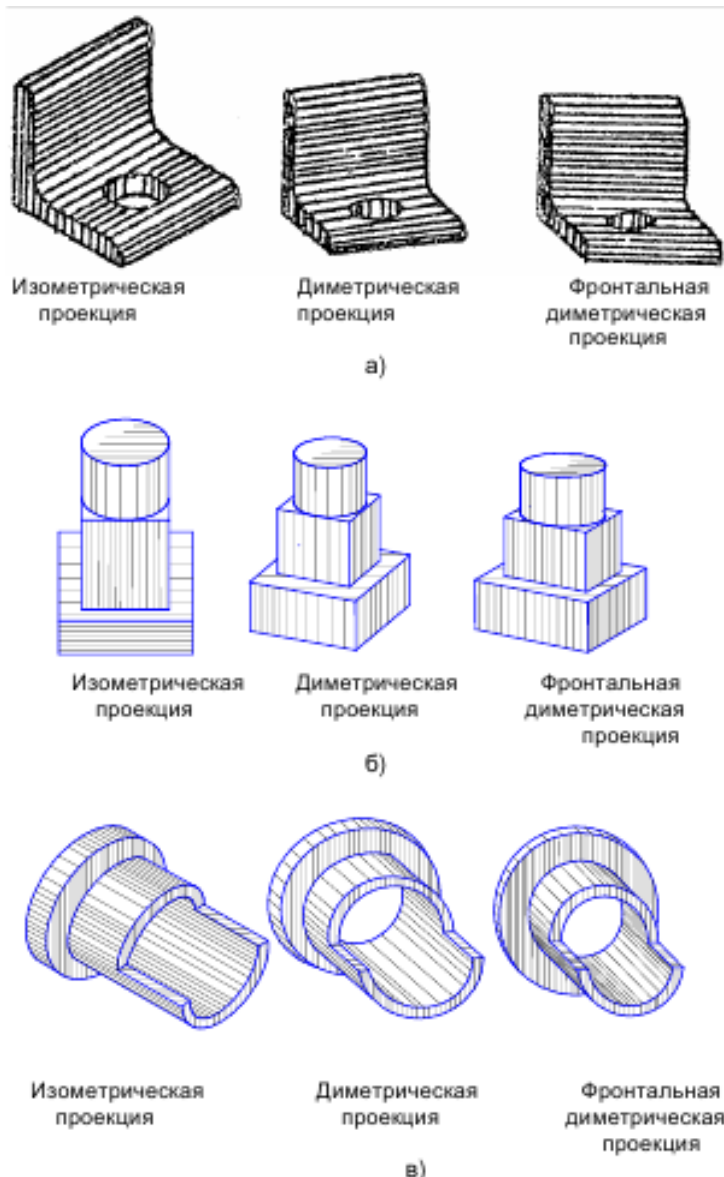


Рисунок 8.25 - Технические рисунки деталей в различных проекциях

Вопросы для самоконтроля

1. Сформулируйте понятие (определение) метода проекций.
2. Перечислите виды проецирования и виды проекций.
3. Объясните, в чём суть метода центрального проецирования.
4. Дайте понятие прямой и обратной задач проецирования.
5. Объясните, в чём суть метода параллельного проецирования.
6. Объясните, в зависимости от чего проецирование является прямоугольным или косоугольным.
7. Перечислите основные инвариантные свойства (б) и следствия, вытекающие из этих свойств.
8. Объясните, в чём суть метода ортогонального проецирования.

9. Объясните, в чём суть метода аксонометрического проецирования.
10. В чём отличие аксонометрических проекций: триметрии, диметрии и изометрии.
11. Приведите примеры стандартных прямоугольных проекций окружностей.
12. Приведите примеры стандартных косоугольных проекций окружностей.
13. Приведите порядок рисования геометрических тел. Приведите примеры.
14. Объясните, для чего на технический рисунок наносят светотень.
15. Объясните, какие аспекты (б) содержит понятие светотень.
16. Приведите схему освещения технического рисунка цилиндра в изометрии и пример его оттенения штриховкой.
17. Приведите примеры оттенения технического рисунка шраффировкой.
18. Объясните порядок выполнения технического рисунка детали.
19. Объясните, от каких геометрических особенностей деталей зависит выбор аксонометрических проекций для получения технического рисунка.

Лекция 9 Основные концепции графического программирования

Графические библиотеки

Интерактивное манипулирование формами составляет значительную часть работы в системах геометрического моделирования. Следовательно, важнейшей составляющей таких систем является программное обеспечение, создающее графическое изображение на экране монитора. Поэтому нам придется изучить терминологию и основные понятия графического программирования.

Термин *программирование на компьютере (computer programming)* раньше означал написание «сочинения» на языке компьютерных команд в соответствии с predetermined правилами грамматики. В ответ на вводимые числа выполняемое «сочинение» порождало ожидаемые числа и символы на терминале или в файле данных. В наши дни на входе и выходе «сочинений» все чаще находится графическая информация. Такое программирование называется *графическим (graphics programming)*, а область его применения - *компьютерной графикой (computer graphics)*.

Помимо основного программного обеспечения, необходимого для обычного программирования (операционная система, редактор и компилятор), графическое программирование требует наличия специальных графических программ. Графические программы делятся на два класса: драйверы устройств и графические библиотеки.

Драйвер устройства может рассматриваться как набор аппаратно-зависимых кодов, непосредственно управляющих процессором графического устройства таким образом, что электронный пучок оказывается направленным в нужное место. Драйверы обязательно являются аппаратно-зависимыми, то есть жестко привязанными к конкретным графическим процессорам. Примерно то же самое можно сказать об ассемблере, конкретный вид которого может выполняться только на процессорах одной и той же модели. То же происходит, если графическая программа использует драйвер устройства непосредственно (рисунок 9.1).

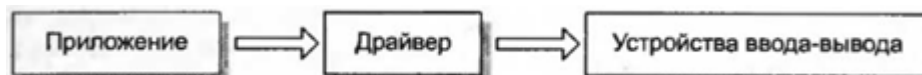


Рисунок 9.1 - Непосредственное использование драйвера устройства

Такую графическую программу при переходе на другое графическое устройство придется переписывать с использованием новых команд драйвера. Более того, команды драйвера устройства весьма примитивны, поэтому такая программа была бы очень длинной, если бы она должна была решать какую-либо сложную задачу. К тому же программа эта получилась бы плохо читаемой. Программисты предпочитают писать программы на языках высокого уровня. Графическое программирование не могло стать исключением, особенно если представить все неудобства, связанные с использованием команд драйвера низкого уровня. Поэтому с графическими устройствами стали поставляться библиотеки, получившие название *графических (graphics libraries)*. **Графическая библиотека**, как, например, и математическая, **представляет собой набор подпрограмм**, предназначенных для решения определенных задач. Конкретная подпрограмма может изображать на экране прямую, круг или иной объект. Графическая библиотека основывается на командах драйвера устройства (рисунок 9.2).

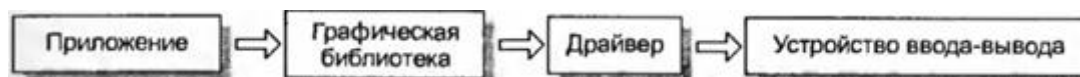


Рисунок 9.2 - Использование графической библиотеки

Каждая подпрограмма создается с использованием поддерживаемого набора команд драйвера. Например, подпрограмма, изображающая окружность, может быть составлена из отдельных команд драйвера, рисующих на экране точки или короткие отрезки.

Подпрограммы графической библиотеки могут использоваться точно так же, как подпрограммы математической. Нужная подпрограмма вызывается из основной программы аналогично тому, как вызываются функции синуса и косинуса, когда программисту требуется вычислить их значения. Одна из проблем использования подпрограмм графической библиотеки связана с тем, что их названия и способы вызова (входные и выходные аргументы) у каждой библиотеки свои. Это не создавало бы трудностей, если бы одна графическая библиотека могла работать со всеми существующими устройствами, что теоретически было бы возможным, если бы все существующие драйверы устройств поддерживали ее. Однако по некоторым причинам производители программного обеспечения не хотят или не могут создать графическую библиотеку, которая могла бы работать со всеми драйверами, а потому у каждой библиотеки имеется свой круг поддерживаемых драйверов. Следовательно, такая библиотека может работать лишь с ограниченным набором графических устройств, а графические программы, рассчитанные на работу со множеством устройств, приходится переписывать с использованием нескольких библиотек.

Чтобы обойти эту проблему, разработчики графических библиотек могли бы использовать одинаковые наборы подпрограмм с одинаковыми названиями, аргументами и возможностями. (На практике каждая подпрограмма реализуется разработчиками по-разному). Тогда графические программы не требовали бы изменения на уровне исходного кода даже при изменении графических устройств. Одним из примеров такого подхода является графическая система CORE, предложенная в 1977 г. группой SIGGRAPH (Special Interest Group on Computer Graphics) Ассоциации вычислительной техники (Association for Computing Machinery - ACM). Однако графическая система CORE не предоставляет достаточного набора команд для использования всех возможностей растровых устройств, потому что эти устройства не были широко распространены во времена разработки системы. Примерно в то же время Международная организация по стандартизации (International Standards Organization — ISO) разработала систему графического ядра (Graphics Kernel System — GKS). GKS считается стандартом в двумерной графике, а позже эта система была расширена до GKS-3D с поддержкой трехмерной графики.

И у CORE, и у GKS есть определенные недостатки в отношении динамического отображения и гибкого взаимодействия с пользователем. Поэтому организация ISO предложила еще один стандарт, получивший название *Иерархическая система программирования интерактивной графики (Programmer's Hierarchical Interactive Graphics System — PHIGS)*. Этот стандарт стал стандартом де-факто для большинства графических рабочих станций. Затем он был расширен и получил название «расширение PHIGS для X» (PHIGS extension to X — PEX), потому что в него были включены функции работы с окнами системы X window. Графические программы, написанные на PEX, в сетевой среде могут

использоваться независимо от типа рабочей станции. Это достоинство было унаследовано от системы X window.

Коммерческая графическая библиотека OpenGL развивалась независимо от организаций по стандартизации, однако с течением времени она набирает все большую популярность благодаря гибкости в управлении рабочими станциями и персональными компьютерами в сетевой среде. OpenGL - расширение графической библиотеки GL (фирменной графической библиотеки для компьютеров Silicon Graphics²). Благодаря популярности компьютеров Silicon Graphics в областях, связанных с компьютерной графикой, библиотека OpenGL постепенно приобрела статус графического стандарта де факто.

Системы координат

Для вывода изображения объекта на экран графического устройства необходимо решить две основные задачи:

- указать положение всех точек объекта в пространстве;
- определить положение их образов на мониторе.

Для задания положения точек в пространстве и на мониторе используются системы координат. Важно понимать, как связаны между собой различные системы координат. Особенно это важно для проектирования трехмерного объекта на плоский экран. Проекция на экране строится по тем же законам, что и проекция реального объекта на сетчатке человеческого глаза.

Первой среди систем координат мы рассмотрим *систему координат устройства* (*device coordinate system*), которая определяет положение точки на экране. Эта система состоит из горизонтальной оси u и вертикальной оси v (рисунок 9.3).

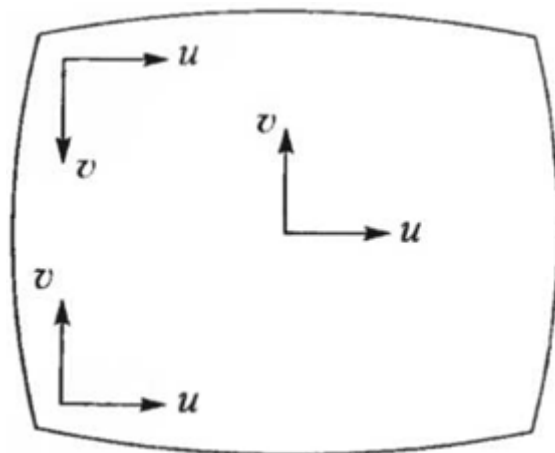


Рисунок 9.3 - Системы координат устройства

Обратите внимание, что начало отсчета может выбираться произвольно. Осей u и v достаточно для задания положения любой точки экрана, поэтому третья ось, перпендикулярная первым двум, не определяется. Положение любой точки задается двумя целыми числами u и v , равными числу пикселей между началом координат и точкой по осям u и v . Однако одна и та же точка может задаваться разными парами u и v в зависимости от положения начала координат, направления осей и масштаба. Эти параметры для разных

² Silicon Graphics — компания, производившая графические рабочие станции, такие как Indigo и 0₂.

графических устройств устанавливаются достаточно произвольно. Поэтому аппаратные координаты в графической программе могут потребовать изменения при смене графического устройства.

Виртуальная система координат устройства (*virtual device coordinate system*) позволяет избежать описанной выше проблемы. Виртуальная система координат устройства фиксирует точку отсчета, направление и масштаб осей для всех рабочих станций. Слово «виртуальный» означает, что данная система отсчета существует только в воображении программиста. Обычно начало отсчета располагается в левом нижнем углу экрана, ось *u* откладывается вправо, а ось *v* - вверх. Обе координаты могут изменяться в диапазоне от нуля до единицы. Точка, положение которой задается в виртуальной системе координат, на любом экране будет попадать в одно и то же место. Это дает программисту возможность единообразно определять формы, не заботясь о конкретных системах координат устройств. Графическая программа передает виртуальные координаты подпрограмм драйвера устройства и преобразует их в координаты конкретного устройства.

Виртуальная и обычная системы координат устройства позволяют задавать положение точки на плоском экране.

Займемся теперь системами координат для работы с трехмерным пространством. Основных **трехмерных систем** координат всего **три**: **внешняя система координат** (*world coordinate system*), **система координат модели** (*model coordinate system*) и **система координат наблюдателя** (*viewing coordinate system*).

Внешняя, или мировая система координат (*world coordinate system*), — это опорная система, используемая для описания интересующего нас мира. Внешней она является по отношению к объектам этого мира. Например, такая система может использоваться для описания расположения и ориентации парт, стульев и доски, если интересующий нас мир представляет собой класс.

Следующим шагом является описание формы каждого объекта мира. **Форма объекта определяется координатами всех или некоторых характеристических точек объекта по отношению к системе координат, связанной с ним, - системой координат модели** (*model coordinate system*). Координаты точек объекта, определенные таким образом, не изменяются даже тогда, когда объект перемещается или вращается в пространстве. Они действительно зависят только от формы объекта. Система координат модели перемещается вместе с тем объектом, к которому она привязана. Поэтому **форма каждого объекта определяется в его собственной системе координат модели**. Расположение и ориентация любого объекта задаются относительным положением и ориентацией модельной системы координат данного объекта по отношению к внешней системе координат. Относительное расположение и ориентация систем координат определяются матрицей преобразования (см. курс линейной алгебры). **Наличие внешней системы координат и модельных систем для всех объектов полностью определяет мир, то есть расположение и форму всех объектов данного мира.** Другими словами, **применение матриц преобразования позволяет получить координаты любой точки любого объекта во внешней системе.**

Следующий шаг - проецирование трехмерных объектов или их точек на монитор подобно тому, как они проецируются на сетчатку человеческого глаза. В компьютерной графике используется два вида проекций: **перспективная и параллельная** (рисунок 9.4).

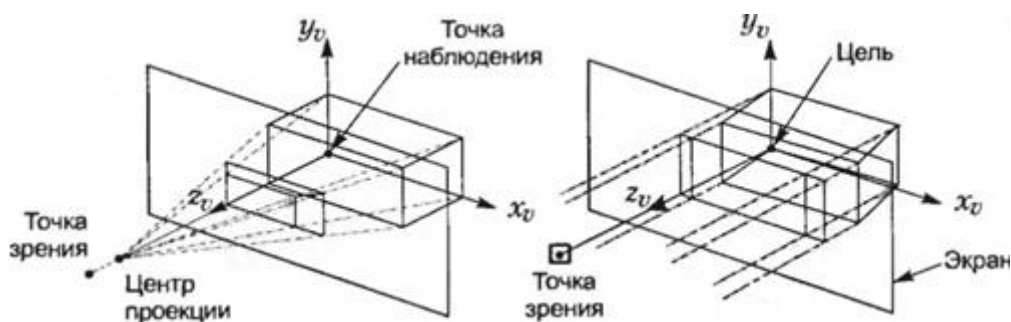


Рисунок 9.4 - Два вида проекций: перспективная и параллельная

Оба вида требуют задания двух точек: точки зрения и точки наблюдения. **Точка зрения** (*viewpoint*) - это глаз наблюдателя. **Точка наблюдения** (*viewsite*) - это точка объекта, определяющая направление «луча зрения». **Вектор, проведенный от точки зрения к цели, задает направление наблюдения.**

В перспективной проекции (*perspective projection*) все точки рассматриваемого объекта соединяются с центром проекции, который обычно лежит на линии, соединяющей точку зрения и цель. Точки пересечения этих линий с экраном образуют проекцию. Экран располагается между точкой зрения и целью. **В параллельной проекции** (*parallel projection*) линии от всех точек объекта проводятся в направлении наблюдателя параллельно направлению наблюдения, а точки пересечения этих линий с экраном формируют проекцию.

Экран, как и в перспективной проекции, располагается перпендикулярно направлению проецирования. Такая проекция называется ортогональной.

Точки проекции, получаемые любым из описанных методов, легко могут быть рассчитаны, если координаты точек проецируемого объекта даны в системе координат x_v, y_v, z_v (см. рисунок 9.4).

Начало координат наблюдательской системы координат может находиться в точке зрения, а ось z_v при этом должна указывать на точку наблюдения. Оси x_v и y_v определяются так же, как обычно, в результате чего система становится левосторонней. Следствием этого является то, что точки с большими значениями Z оказываются дальше от наблюдателя. По принятому нами соглашению объекты с большими значениями Z расположены ближе к наблюдателю.

Например, координаты точек параллельной проекции объекта попросту равняются соответствующим координатам X_v и Y_v точек объекта. Система координат x_v, y_v, z_v называется **наблюдательской** (*viewing coordinate system*), поскольку она облегчает расчет проекции наблюдения. Наблюдательская система координат строится таким образом, чтобы обладать перечисленными ниже характеристиками. Начало этой системы координат располагается в рассматриваемой точке, ось z_v направлена из начала координат в точку зрения, а ось y_v параллельна вертикальной оси экрана (см. рис. 9.4). Третья ось, x_v определяется как векторное произведение первых двух. У большинства людей вертикальное направление в пространстве естественным образом ассоциируется с вертикальным направлением на экране, поэтому ось y_v считается проекцией вертикального вектора из пространства на экран. В большинстве графических библиотек пользователю приходится задавать вектор вертикали в пространстве в мировых координатах. Положение точки зрения и точки наблюдения также задается в мировых координатах (рисунок 9.5).

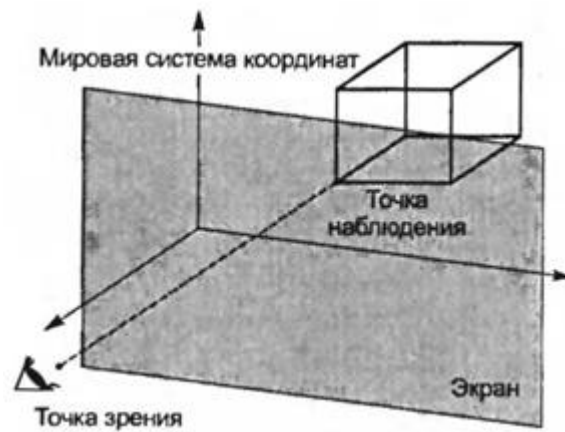


Рисунок 9.5 - Точка зрения и точка наблюдения

После определения наблюдательской системы координат и вычисления координат точек всех интересующих нас объектов остается только вычислить положение их проекций на экране.

Взаимоотношения перечисленных выше систем координат иллюстрирует рисунок 9.6.

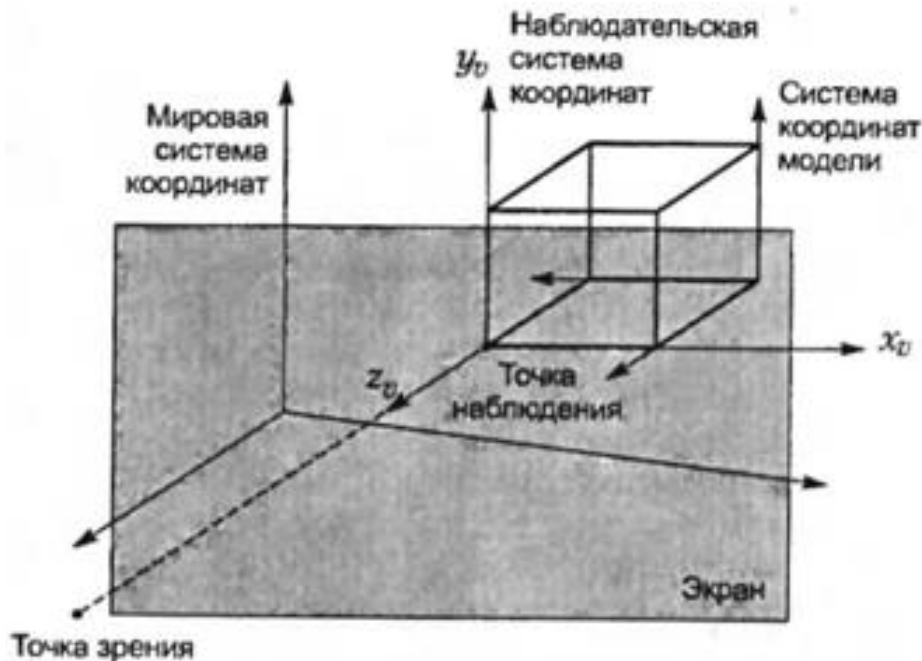


Рисунок 9.6 - Взаимоотношение систем координат

Как уже говорилось, системы координат связаны друг с другом матрицами преобразования. Так, положение и ориентация каждой из модельных систем координат по отношению к мировой задаются соответствующими матрицами преобразований. Наблюдательская система координат также может быть определена относительно мировой при помощи матрицы преобразования, если задать положение точек зрения и наблюдения, а также вектор вертикали. Процедура расчета точек проекции с использованием матриц преобразования выглядит следующим образом. **Сначала** координаты проецируемой точки преобразуются **из модельных в мировые** при помощи матрицы преобразования, определяющей переход от модели, к которой относится точка, к мировой системе координат. Эта операция (рисунок 9.7) называется **преобразованием модели** (*model transformation*).

Затем координаты этой точки преобразуются из мировой системы координат в наблюдательскую. Эта операция (см. рисунок 9.7) называется *преобразованием просмотра* (*viewing transformation*). Наконец, координаты в наблюдательской системе координат преобразуются в значения X_s и Y_s , а затем - в виртуальные координаты устройства.



Рисунок 9.7 - Преобразования между системами координат

Эта операция называется *преобразованием проецирования* (*projection transformation*). Наконец, **виртуальные координаты** устройства преобразуются в **обычные** подпрограммой драйвера.

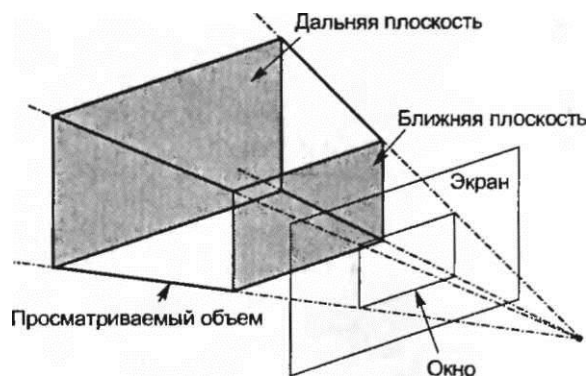
Все эти преобразования обычно выполняются внутри графической библиотеки, а программисту приходится только указывать сведения, необходимые для проведения преобразований. Например, трансляции и повороты объектов учитываются при преобразовании модели, положение точки зрения, точки наблюдения и вектора вертикали - при преобразовании наблюдения, а тип проекции, расположение центра проекции и экрана - при преобразовании проецирования. Однако графические библиотеки примитивного уровня могут потребовать от программиста самостоятельного написания кода для всех этих преобразований.

Окно и видовой экран

Термин *окно* (*window*) в сетевой компьютерной среде обозначает область экрана монитора, посредством которого пользователь взаимодействует с вычислительными ресурсами, подключенными к той же сети. В компьютерной графике этот термин имеет иное значение. Окно - это область пространства, проецируемая на монитор. Объекты, находящиеся вне окна, на мониторе не появляются. В этом смысле оно подобно окну дома, через которое человеку, сидящему внутри дома, видна лишь часть внешнего мира. Вероятно, эта аналогия

Diagram illustrating the viewing frustum in a 3D coordinate system. The coordinate system has axes x_v , y_v , and z_v . The viewing frustum is defined by a viewing volume (Просматриваемый объем) and a viewing point (Точка наблюдения). The viewing frustum is projected onto a screen (Экран) through a window (Окно). The viewing point is labeled as Точка зрения (Viewing point).

Просматриваемый объем при проецировании может давать довольно сложное изображение, поскольку в него могут попадать ненужные объекты, расположенные вблизи наблюдателя или вдали от него. Иногда бывает удобно ограничить этот объем двумя плоскостями, из которых одна располагается ближе, а другая - дальше (рисунок 9.10). Для параллельной проекции ближняя и дальняя плоскости определяются так же, как и для перспективной.



122

Видовой экран (viewport) - это область экрана, где будет отображаться проецируемое изображение (рисунок 9.11).

В эту область проецируется просматриваемый объем, определяемый «обычным» окном. Отображение состоит из трансляции и масштабирования, учитывающих расстояние между центром видового экрана и центром монитора, а также разницу размеров окна и видового экрана. Другими словами, значения X_s и Y_s , должны быть увеличены или уменьшены таким образом, чтобы центр окна попадал в центр видового экрана, а не в центр монитора.



Рисунок 9.11 - Видовые экраны

Кроме того, они должны быть подвергнуты масштабированию, чтобы четыре граничные точки окна стали четырьмя граничными точками видового экрана. Соотношение сторон у окна должно быть таким же, как и у видового экрана, в противном случае изображение будет искажено, и окружность, например, превратится в эллипс.

Примитивы

Примитивы (primitives) - это элементы графики, которые могут отображаться графической библиотекой. В каждой библиотеке набор примитивов свой, поэтому в данном разделе мы рассмотрим только наиболее общие примитивы, поддерживаемые большинством графических библиотек.

Отрезок

Для отображения *отрезка прямой (линии - line)* необходимо задание координат двух его концов. В большинстве графических библиотек координаты концов могут задаваться в трехмерном пространстве; проецирование на плоскость экрана осуществляется автоматически. Можно указывать атрибуты отрезка: тип, толщину, цвет и другие. Типы отрезков, поддерживаемых большинством графических библиотек, изображены на рисунке 9.12. Для систем автоматизированной разработки чертежей поддержка этих типов линий совершенно необходима, поскольку они часто используются в машиностроительных и архитектурных чертежах и электрических схемах.

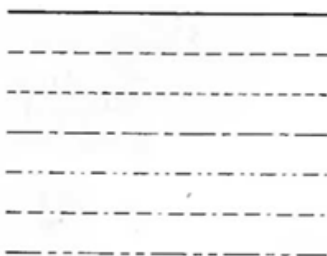


Рисунок 9.12 - Различные виды отрезков

В библиотеках GKS, PHIGS и OpenGL одной из базовых функций является *ломаная* (*polyline*), представляющая собой набор соединенных друг с другом отрезков. Координаты концов отрезков, составляющих ломаную, задаются в виде матрицы. В случае ломаной, состоящей всего из одного отрезка, в матрицу помещаются координаты двух его концов.

Многоугольник

Многоугольник - это то же самое, что и ломаная, за небольшим исключением: первая и последняя строки матрицы вершин должны быть одинаковыми (соответствующие им точки совпадают). Того же результата можно было бы достичь и с использованием функции построения ломаной, однако многоугольник, построенный при помощи специальной функции, распознается системой как объект, имеющий внутреннюю и внешнюю части. Внутренняя площадь многоугольника может быть заполнена штриховкой различных видов (рисунок 9.13).

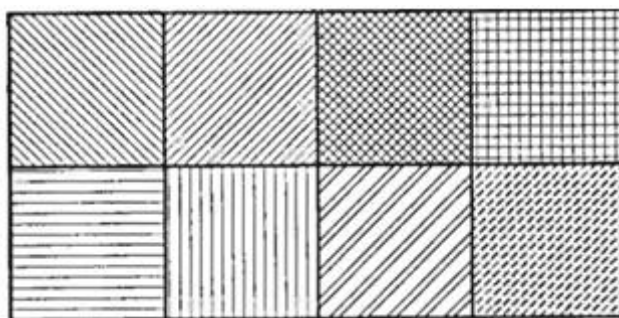


Рисунок 9.13 - Примеры различных штриховок

Атрибутами многоугольника могут быть цвет внутренней области (цвет заполнения), а также тип, ширина и цвет ломаной, ограничивающей эту область. Хотя функция построения многоугольников может использоваться и для построения кругов и прямоугольников, в большинстве графических библиотек существуют специальные функции, требующие гораздо меньше входных параметров (например, центр и радиус круга или два конца диагонали прямоугольника). Тем не менее, внутри библиотек эти функции реализованы через функцию построения многоугольников.

Текст

Большинство графических библиотек поддерживают два вида текста: текст для пояснений (экранный или двумерный текст) и трехмерный текст. Текст для пояснений всегда располагается в плоскости экрана, поэтому его форма не искажается вне зависимости от угла, на который он повернут. Трехмерный текст может быть расположен на любой плоскости в трехмерном пространстве. Его положение и ориентация задаются в мировых координатах. Для текста любого вида необходимо задание таких параметров, как шрифт, отношение высоты к ширине и угол наклона букв, а также положение и направление строки текста. Текст может быть представлен символами двух видов: аппаратными и программными. Программный шрифт строится соответствующими графическими программами, заранее сохраняемыми в памяти компьютера. Построение его занимает больше времени, чем построение символов аппаратного шрифта, но зато форма может быть гораздо более

замысловатой. Символы аппаратного шрифта состоят из отдельных отрезков, формирующих буквы.

Ввод графики

Как уже отмечалось, графической программе может потребоваться поддержка ввода графических элементов, таких как точки, отрезки и многоугольники, а не только чисел и текстовых строк. Например, если пользователь хочет вычислить площадь многоугольника на экране или увеличить его размеры, он должен сначала указать интересующий его многоугольник среди прочих объектов, видимых на экране. Для ввода графики используется два вида физических устройств: локатор (устройство ввода координат) и кнопка. *Локатор (locator)* передает графической программе информацию о своем положении, то есть о положении курсора. *Кнопка (button)* сообщает о действиях пользователя (включении и выключении) в месте текущего положения курсора. В наши дни наиболее популярным устройством графического ввода является мышь, которая выполняет обе функции. Шарик в нижней части корпуса мыши позволяет вводить координаты, а кнопки наверху корпуса передают программе действия пользователя.

Устройство графического ввода может работать в трех режимах: опрос, запрос и выбор. В режиме *опроса (sampling)* осуществляется постоянное считывание состояния устройства ввода, прежде всего положения локатора. Например, если вы свободно рисуете на экране, перемещая мышь, она работает в режиме опроса. Перемещение мыши приводит к непрерывному перемещению курсора по экрану. В режиме *запроса (requesting)* положение локатора считывается только при отправке запроса, который обычно производится при нажатии на кнопку мыши. Чтобы прояснить различие между режимами опроса и запроса, рассмотрим процесс построения многоугольника путем графического задания координат его вершин при помощи мыши. В этом случае мы перемещаем мышь до тех пор, пока курсор не окажется в нужном месте, после чего нажимаем кнопку. Курсор перемещается по экрану согласно движениям мыши, которая находится при этом в режиме опроса. Координаты вершин передаются графической программе в режиме запроса. У этих режимов есть общее свойство: графической программе передаются координаты мыши или курсора. В режиме *выбора (picking)* устройство графического ввода идентифицирует элемент экрана, на который указывает курсор в момент нажатия кнопки. Графические элементы можно идентифицировать по именам, присвоенным им программистом во время составления программы. Режим выбора очень удобен при редактировании чертежа, уже имеющегося на экране (например, для удаления многоугольников или изменения координат их вершин).

Дисплейный файл

Дисплейный файл (display list) - это группа команд графической библиотеки, сохраненная для последующего выполнения. Большинство команд графических библиотек могут либо помещаться в дисплейный файл, либо выполняться немедленно. Дисплейный файл обеспечивает удобство и эффективность упорядочения и обработки команд библиотеки. Рассмотрим, например, перемещение изображения дома по экрану. Предположим, что рисунок состоит из нескольких сотен отрезков. Если бы эти отрезки существовали по отдельности, нам пришлось бы написать команду перемещения несколько сотен раз - для

каждого из них. Однако если команды построения отрезков, образующих рисунок, объединены в дисплейный файл, команду перемещения достаточно написать только один раз. Чтобы поместить графические элементы в дисплейный файл, нужно открыть этот файл перед первой командой, которая должна в него попасть, и закрыть его после последней команды:

OpenGL

```
glNewList(AREA_FILL, GL_COMPILE_AND_EXECUTE):
```

```
/* Открытие дисплейного файла с именем AREA_FILL. */
```

```
glBegin(GL_POLYGON):
```

```
glVertex2fv(point1):
```

```
glVertex2fv(point2):
```

```
:
```

```
:
```

```
glEnd( );
```

```
/* Определение многоугольника. */
```

```
glEndList( );
```

```
/* Закрытие дисплейного файла. */
```

Дисплейный файл OpenGL ориентирован на оптимизацию производительности, в частности, при работе по сети, но не за счет производительности на отдельном компьютере. Оптимизация обеспечивается благодаря тому, что дисплейный файл хранится в виде списка команд, а не в виде динамической базы данных. Другими словами, созданный дисплейный файл изменить уже нельзя. Если бы его можно было изменять, производительность упала бы из-за накладных расходов на поиск внутри списка и управление памятью. Изменение отдельных частей дисплейного файла потребовало бы перераспределения памяти, что могло бы привести к ее фрагментации. Дисплейные файлы, как правило, работают так же быстро, как и обычные последовательности команд, не объединенные в группы. В случае OpenGL дисплейные файлы могут значительно повысить производительность, в особенности при передаче подпрограмм OpenGL по сети, поскольку файлы эти хранятся на сервере, благодаря чему сокращается сетевой трафик. К созданному дисплейному файлу могут быть применены следующие операции:

- множественное выполнение (multiple execution) - один и тот же файл можно выполнять много раз;

- иерархическое выполнение (hierarchical execution) - иерархическим называется дисплейный файл (родительский), вызывающий другие дисплейные файлы (дочерние). Иерархические дисплейные файлы удобны для объектов, состоящих из отдельных компонентов, особенно если некоторые компоненты входят в объект в нескольких экземплярах;

- удаление (deletion) - дисплейный файл может быть удален.

Матрица преобразования

Как говорилось выше, проецирование точек на объект в трехмерном пространстве требует преобразования координат из одной системы в другую. Сначала нужно перевести координаты точек объекта из модельной системы в мировую. Текущее положение объекта

обычно задается через повороты и смещения относительно исходного положения, в котором модельная система координат совпадала с мировой. Следовательно, мировые координаты точек объекта можно получить трансляцией и поворотом соответствующих точек из их исходного положения, в котором их модельные координаты совпадали с мировыми. Большинство графических библиотек выполняют эти преобразования самостоятельно, а программисту остается задать только смещение и поворот для интересующего его объекта. Однако проектировщику все равно нужно знать законы преобразований, чтобы рисовать объекты в нужных местах без проб и ошибок, в особенности если эти объекты перемещаются достаточно сложным образом.

Получив мировые координаты всех точек объекта в его текущем положении, мы должны вычислить координаты этих точек в наблюдательской системе. Перевод координат из одной системы в другую называется *отображением (mapping)*. Отображение между мировой и наблюдательской системами координат обычно также осуществляется графической библиотекой самостоятельно, по заданным программистом координатам точки зрения, точки наблюдения и направлению вектора вертикали (в мировых координатах).

Удаление невидимых линий и поверхностей

Проекция на экран станет более наглядной, если будет содержать только видимые линии и поверхности. *Удаление невидимых линий (hidden-line removal)* заключается в блокировании отображения отрезков, скрытых от наблюдателя, а *удаление невидимых поверхностей (hidden-surface removal)* есть то же самое по отношению к поверхностям. Удаление невидимых линий иллюстрирует рисунок 9.14. Очевидно, что эта процедура значительно облегчает восприятие объекта.

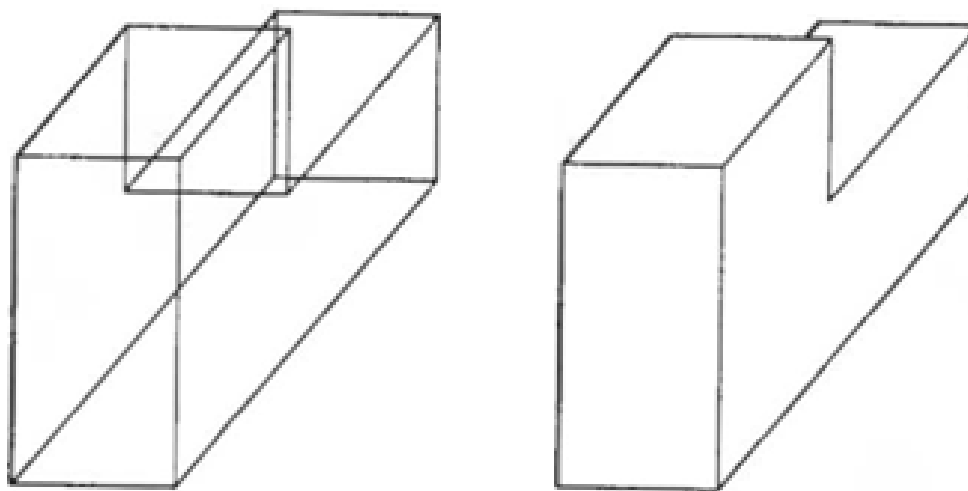


Рисунок 9.14 - Изображение до и после удаления невидимых линий

Опубликовано множество программных алгоритмов удаления невидимых линий и поверхностей. Их авторы пытаются повысить вычислительную эффективность и расширить диапазон объектов для применения своих алгоритмов. Однако лучше всего удаление реализуется посредством графического устройства, которое называется z-буфером, поэтому в настоящее время исследования на эту тему практически не ведутся.

Метод z-буфера

Метод z-буфера основан на том же принципе, что и алгоритм сортировки по глубине: на любом участке экрана оказывается проекция элемента, расположенного ближе всех прочих к наблюдателю. Здесь под элементами понимаются точки, кривые или поверхности. Данный метод требует использования области памяти, называемой z-буфером. В этом буфере для каждого пиксела хранится значение координаты Z_v того элемента, проекция которого изображается данным пикселом. Как уже говорилось, значение Z_v (то есть координата z в наблюдательской системе) есть мера удаленности объекта от наблюдателя. Объем z-буфера определяется количеством пикселов, для каждого из которых требуется сохранить вещественное число.

Грани, векторы нормалей которых направлены от наблюдателя, невидимы для него, поэтому на экран проецируются только те грани, векторы нормалей которых направлены к наблюдателю. Однако в отличие от метода сортировки по глубине, в данном случае порядок проецирования значения не имеет. Причина станет очевидной, как только вы прочитаете приведенное ниже описание алгоритма. Сначала проецируется произвольно выбранная поверхность, и в ячейки памяти z-буфера, соответствующие пикселам проекции поверхности, записываются значения координат Z_v точек поверхности, являющихся прообразами данных пикселов. Пикселы окрашиваются в цвет первой поверхности. Затем проецируется следующая поверхность, и все неокрашенные пикселы, относящиеся к ее проекции, окрашиваются в цвет второй поверхности. Если пикселы уже были окрашены, соответствующие им значения Z_v сравниваются со значениями Z_v точек текущей поверхности. Если сохраненное значение Z_v какого-то пиксела оказывается больше текущего (то есть точка на предыдущей поверхности ближе к наблюдателю, чем точка на текущей поверхности), цвет пиксела не меняется. В противном случае пиксел окрашивается в цвет текущей поверхности. Изначально z-буфер инициализируется координатами Z_v соответствующими дальней плоскости (см. рисунок 9.11), поэтому пикселы проекции первой поверхности автоматически окрашиваются в ее цвет. Повторяя эту процедуру для всех имеющихся плоскостей, мы окрасим все пикселы экрана в цвета ближайших поверхностей (рисунок 9.15).

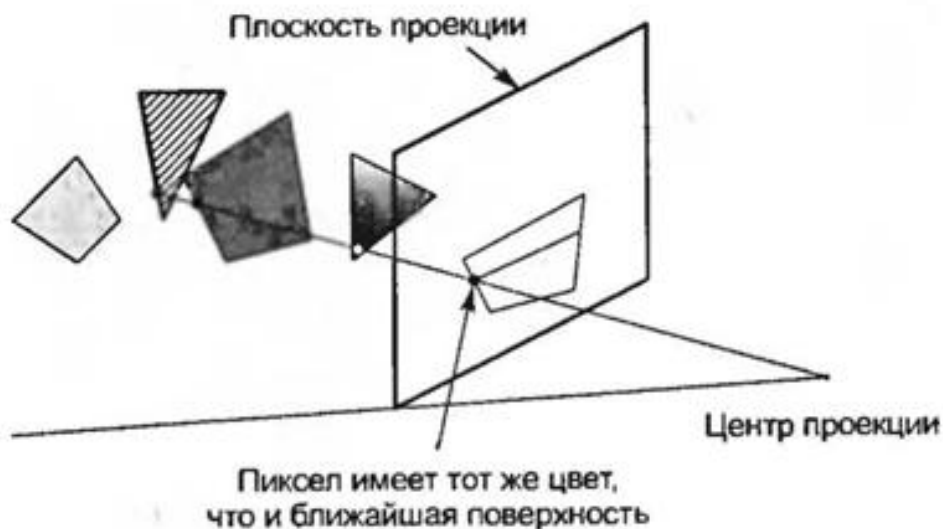


Рисунок 9.15 - Основы метода z-буфера

Как следует из предшествующего описания, метод z-буфера предназначен главным образом для удаления скрытых поверхностей. Однако метод z-буфера с небольшими изменениями позволяет реализовать и построение рисунков со скрытыми линиями. Сначала все поверхности проецируются на экран, причем пиксели окрашиваются в цвет фона (на экране ничего не появляется). При этом z-буфер заполняется правильными значениями Z_v . Затем на экран проецируются ребра поверхностей. При этом значения Z_v ребер сравниваются со значениями Z_v ближайших к наблюдателю поверхностей, уже найденными на предыдущем этапе. Окрашиваются только те пиксели, для которых новые значения Z_v больше исходных. Таким образом, участки ребер, скрытые поверхностями, на экране не появляются. Эта процедура дает правильный рисунок без скрытых линий, но некоторые граничные линии могут оказаться слишком тонкими, потому что некоторые пиксели этих линий будут относиться к поверхностям, ограниченными соответствующими ребрами. Эту проблему легко решить, пододвинув весь объект поближе в момент проецирования ребер.

Визуализация

Изображения без невидимых линий и поверхностей в большинстве случаев передают форму объекта достаточно хорошо. Однако для некоторых приложений желательно иметь возможность строить более реалистичные изображения. Особенно важную роль реалистичность играет в «виртуальной реальности» - относительно новой технологии, позволяющей имитировать на компьютере реальные ситуации. Например, вы можете занести в компьютер сведения о внутреннем и внешнем виде проектируемого здания, а затем пройти по нему и вокруг него, чтобы прочувствовать, как оно будет выглядеть на самом деле. Это средство может оказаться очень полезным для принятия решения о вложении денег в постройку здания.

Для имитации реальной сцены нужно воспроизвести эффекты, создаваемые светом, падающим на поверхности объектов. Этот процесс называется *визуализацией* или *тонированием (rendering)*. Вообще говоря, все, что мы видим, - это отражение света от поверхностей, а по этому отраженному свету мы определяем форму, текстуру и цвет объекта. В усовершенствованных графических библиотеках обычно имеются некоторые средства визуализации. Для их применения достаточно задать фасетированную модель объекта, условия освещенности и свойства поверхностей (например, блестящие или тусклые). Однако для правильного использования технологий визуализации необходимо иметь представление об их основах. Даже задать правильные значения параметров функций визуализации без достаточно хорошего знания техники визуализации непросто. Поэтому в настоящем разделе мы коротко опишем две основные технологии визуализации: *затушевывание (shading)* и *трассировку лучей (ray tracing)*. Подробное описание технологий визуализации можно найти в большинстве книг по компьютерной графике.

Затушевывание

Процедура затушевывания аналогична удалению невидимых поверхностей с той разницей, что пиксели одной поверхности окрашиваются разными цветами в соответствии с цветом и интенсивностью отраженного света в точке, проектируемой на этот пиксел. Таким

образом, основной задачей становится расчет цвета и интенсивности света, отраженного от какой-либо точки объекта. Начнем с расчета интенсивности.

Поверхность объекта может быть освещена светом, исходящим непосредственно от источников, то есть *прямым светом (direct illumination)*, а также светом, отраженным от других поверхностей, дающим *окружающее освещение (ambient illumination)*. Свет, отраженный данной точкой объекта, получается сложением отраженных лучей двух типов (рисунок 9.16).

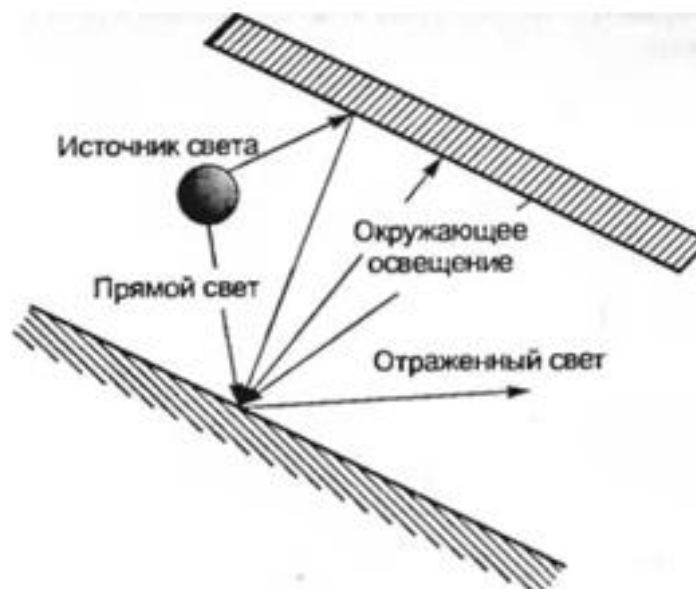


Рисунок 9.16 - Прямое и окружающее освещение

Отраженный от множества поверхностей сцены свет может считаться приходящим с бесчисленного множества направлений, поэтому отражение окружающего освещения считается равномерно распределенным во всех направлениях. Поэтому интенсивность отражения R_a записывается по формуле:

$$R_a = K_d I_a \quad (9.1)$$

Здесь I_a - интенсивность окружающего освещения, а K_d - коэффициент отражения поверхности. У белых поверхностей коэффициент отражения близок к единице, а у черных - к нулю. Наблюдатель будет воспринимать одну и ту же интенсивность отраженного окружающего освещения вне зависимости от своего положения, поскольку это освещение образуется благодаря отражению от всех поверхностей и распределяется равномерно по всем направлениям.

Для описания прямого освещения достаточно рассмотреть лишь точечные источники света, поскольку линейные и поверхностные источники могут быть рассмотрены как совокупность точечных. Отражение света от точечного источника рассматривается как комбинация двух видов отражения: диффузного и зеркального.

Диффузное отражение (diffuse reflection) состоит в том, что поверхность поглощает свет, а затем переизлучает его равномерно во всех направлениях (рисунок 9.17). Интенсивность диффузного отражения (как и отражения окружающего освещения) не зависит от положения наблюдателя. Особенности структуры поверхности, такие как текстура и цвет, становятся

видимыми именно благодаря диффузному отражению. Для грубых поверхностей отражение является преимущественно диффузным, а не зеркальным.

Зеркальное отражение (specular reflection) - это прямое отражение света поверхностью. Данный тип отражения характерен для блестящих поверхностей, таких как зеркала. Блестящие поверхности отражают почти весь падающий свет и потому обладают повышенной яркостью (рисунок 9.18). Интенсивность зеркального отражения воспринимается по-разному в зависимости от положения наблюдателя относительно сцены.

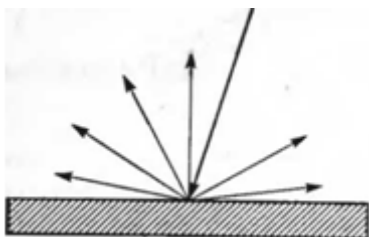


Рисунок 9.17 - Диффузное отражение

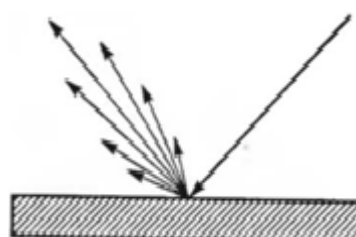


Рисунок 9.18 - Зеркальное отражение

Трассировка лучей

Метод затухивания, описанный в предыдущем разделе, может использоваться в относительно простых условиях, когда один объект освещается далеко расположенными точечными источниками света. Однако он неприменим в случае множества объектов, особенно если некоторые из них прозрачны, а другие преломляют лучи. В сложных ситуациях используется метод *трассировки лучей (ray tracing)*.

Основная идея метода заключается в следующем. Источники света испускают лучи во всех направлениях, и все эти лучи прослеживаются до тех пор, пока они не попадают в одну из точек экрана. В процессе прослеживания вычисляется изменение интенсивности и направления луча при падении его на какой-либо объект. Направление меняется по законам рассеяния, отражения и преломления. Каждая точка экрана (пиксел) окрашивается в цвет, соответствующий интенсивности пучка света в момент его падения на экран. Поскольку для заполнения экрана таким путем пришлось бы проследить бесконечно много лучей, такой подход на практике нереален. Нас интересуют цвета конечного числа пикселей экрана, поэтому рассматривать бесконечное количество лучей не обязательно. Нужно рассмотреть траектории конечного числа лучей в обратном направлении. Луч проходит из центра проекции сквозь каждый пиксел и прослеживается в обратную сторону до тех пор, пока он не выйдет из просматриваемого объема, не упадет на рассеивающую поверхность или не войдет в источник света (рисунок 9.19).

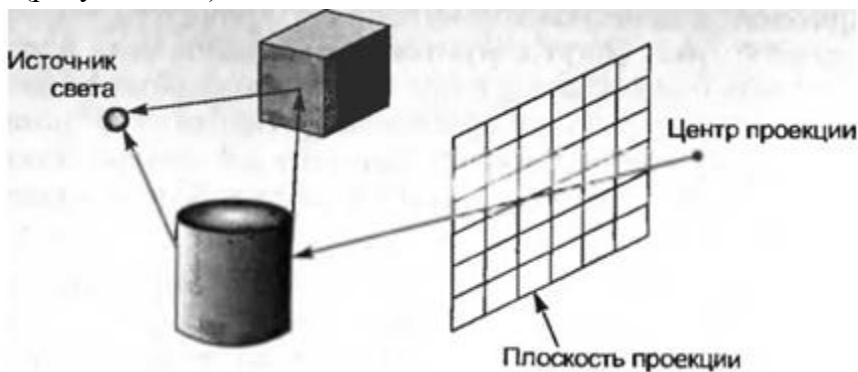


Рисунок 9.19 - Трассировка лучей

Если луч достиг источника света, это означает, что между источником и экраном отсутствуют непрозрачные препятствия, поэтому пиксел окрашивается в цвет источника. Если луч выходит из просматриваемого объема, значит, никакие объекты на данный пиксел не проецируются, и он должен быть окрашен в цвет фона. Если же луч падает на рассеивающую поверхность, проследить его траекторию дальше невозможно. В этом случае цвет пиксела окрашивается в цвет отражаемого данной поверхностью света. Следовательно, интенсивности света на рассеивающих поверхностях должны быть вычислены заранее по методу затухивания, описанному выше.

Вопросы для самоконтроля

1. Объясните, какими недостатками обладают графические программы, написанные непосредственно с использованием команд драйвера устройства.
2. Объясните, почему графическая программа, основанная на какой-либо графической библиотеке, может выполняться лишь на ограниченном числе графических устройств.
3. Какова главная причина, по которой для задания положения на графическом устройстве используется виртуальная, а не обычная система координат устройства?
4. Перечислите системы координат (4), используемых для отображения пространственной модели на плоский экран.
5. Почему для задания формы объекта используется модельная система координат этого объекта?
6. Объясните, каким образом задаются положение и ориентация каждого объекта сцены.
7. Кратко опишите процедуру преобразования координат точки объекта из модельной системы в экранную.
8. Объясните значение термина «окно» в компьютерной графике.
9. Объясните значение термина «окно просмотра».
10. Объясните различие между режимами выбора и опроса.
11. Какие операции могут выполняться с экраным файлом?
12. Перечислите матрицы преобразований координат, которые используют в компьютерной графике.
13. Объясните в чём суть метода удаления невидимых линий и поверхностей.
14. Расскажите, для чего используется метод Z-буфера и схему его реализации.
15. Объясните, с какой целью реализуется визуализация (тонирование).
16. Приведите схемы реализации технологий визуализации: затухивание и трассировка лучей.

Лекция 10

Возможности CAD модуля системы ADEM

ADEM - интегрированная CAD/CAM/CAPP/PDM система содержит средства автоматизации для различных видов инженерной деятельности, оперативное взаимодействие которых является ключом к рентабельному производству.

Бурное развитие “плоских чертилок”, как иногда называют 2D-системы, пришлось на период 1985-90 годов. В 1987 году компанией была выпущена первая версия системы CherryCAD (черри – аббревиатура слов чертить, рисовать), главным отличием которой был иконный интерфейс, устранявший необходимость длительной подготовки пользователей.

Многие уникальные черты CherryCAD были позже воплощены и в других программных продуктах. Система оказалась настолько удачной, что стала лидером по продажам в СССР среди всех отечественных и зарубежных CAD систем. В 1990 г. коллектив разработчиков был удостоен премии СМ СССР.

Если в восьмидесятых потребность в объемном (3D) моделировании в России испытывали пожалуй только авиационные и автомобилестроительные предприятия, то в девяностых в нем уже нуждались почти все.

Разработку модуля объемного моделирования вели с самого начала создания системы. Первый ADEM 3D был полностью написан на собственном ядре. Версия середины девяностых позволяла строить твердотельные модели и сборки (рисунок 10.1).

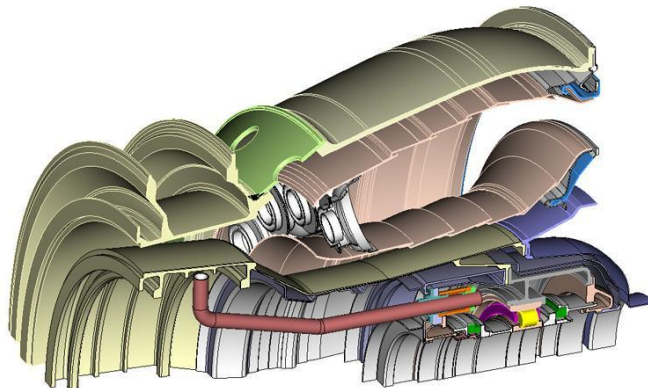


Рисунок 10.1 - Объемное моделирование на собственном ядре в ADEM v.3 1996 год

Главный эффект достигался при использовании объемного моделирования в качестве инструмента создания технологичных моделей для последующей механообработки. Эта маленькая добавка - "технологичных" и определила последующее направление разработок.

В связи с этим было принято решение взять в качестве основного ядра для 3D моделирования библиотеки ACIS (Spatial Technology), добавив к нему собственную часть для дополнительной функциональности.

При детальном изучении вопроса стало очевидным, что для реального производства одних твердотельных методов недостаточно. Нужно еще предоставить возможности ведения локального редактирования методами поверхностного моделирования. То есть должно быть реализовано гибридное объемное моделирование.

Задача сведения воедино разнородных плоских и объемных методов проектирования была решена в седьмой версии. В результате появился универсальный и довольно компактный инструмент для конструкторов и технологов с широким спектром применения.

ADEM A7 - с таким названием система вышла на отечественный и зарубежный рынки как единое 2D/3D/CAD/CAM пространство (рисунок 10.2).

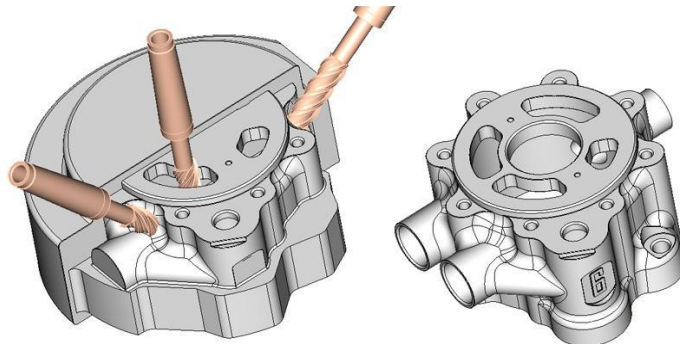


Рисунок 10.2 - Интегрированный 2D/3D/CAD/CAM ADEM V.7 2002 год

Из глубокой интеграции инструментов для конструкторов, технологов и программистов ЧПУ в системе ADEM родилась и следующая важная возможность. Теперь в процессе подготовки моделей, чертежей, техпроцессов и программ стала осуществляться генерация важнейших данных, которые нужны не только конструкторам и технологам, но и другим специалистам, например, для задач управления предприятием и производством. Используя эти данные, можно очень точно, планировать и прогнозировать финансово - экономические аспекты инженерно-технической и производственной деятельности предприятия.

На сегодняшний день в ADEM-VX интегрированы все основные средства автоматизации от простого черчения и оформления документации до глубокой технологической проработки и программирования оборудования с ЧПУ.

Дополняет функциональность системы новейшая разработка - модуль ADEM CAM Expert, который производит технологический анализ объемной модели, автоматическое распознавание конструктивных элементов и задаёт маршрут обработки детали (рисунок 10.3).

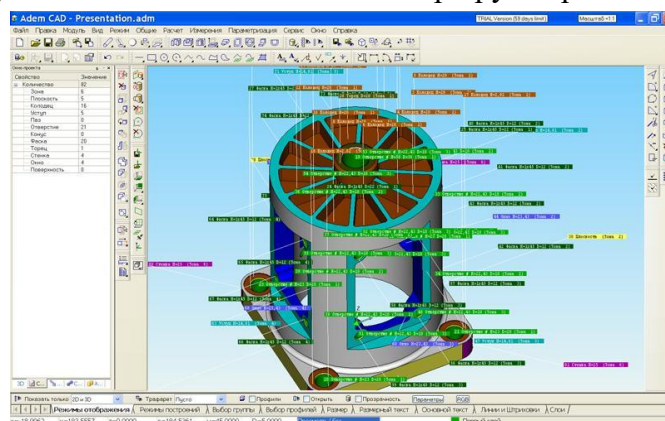


Рисунок 10.3 - Автоматическое распознавание конструктивных технологических элементов в ADEM 2013

Кроме сквозного потока информации каждый модуль системы ADEM VX имеет возможности импорта сторонних данных. Так, например CAD импортирует модели напрямую

из других систем, а также посредством стандартов SAT, STEP, IGES и т.п., чертежи через DXF, DWG, а сканированные чертежи через растровые форматы TIFF, PCX, JPEG и другие.

Функциональные возможности ADEM VX

К базовой функциональности, находящейся в ядре, теперь можно отнести следующие возможности:

- компьютерная обработка бумажных чертежей;
- плоское моделирование, черчение;**
- оформление конструкторской документации;
- оформление спецификаций;
- работа с архивами, документооборот;
- объемное твердотельное моделирование;**
- объемное поверхностное и гибридное моделирование;**
- получение чертежей от объемной модели;**
- анализ геометрии и корректности конструкции;**
- проектирование и планирование техпроцессов;
- плоское фрезерование 2х-2,5х;
- объемное фрезерование 3х-5х;
- квазиобъемное фрезерование (Z-level);
- "карандашная" обработка;
- фрезерование недоступных зон;
- зонная и комбинированная обработка;
- токарная обработка;
- электроэрозионная обработка 2х-4х;
- листоштамповка.

Рассмотрим возможности функционала системы, относящиеся к функциям геометрического моделирования.

Плоское моделирование, черчение

Черчение в системе ADEM основывается на двух схемах: с использованием элементов (отрезок, ломаная, окружность, кривая и т.п.) и методов их построения (по точкам, касательно, перпендикулярно и т.п.) и с использованием связанных контуров и булевых операций.

Первая схема традиционна для большинства современных CAD систем и в пояснениях не нуждается.

Второй вариант позволяет пользователю мыслить объектами более высокого уровня, нежели отдельные дуги и отрезки. В его распоряжении находятся, так называемые, связанные контуры, которые он может модифицировать, не разрушая их целостности и внутренних условий сопряжения (рисунок 10.4). Более того он может производить с ними операции сложения, вычитания, дополнения, создавая новые конструкции.

Объемное твердотельное, поверхностное и гибридное моделирование

Платформу для объемного моделирования в системе ADEM составляют два способа создания исходных профилей:

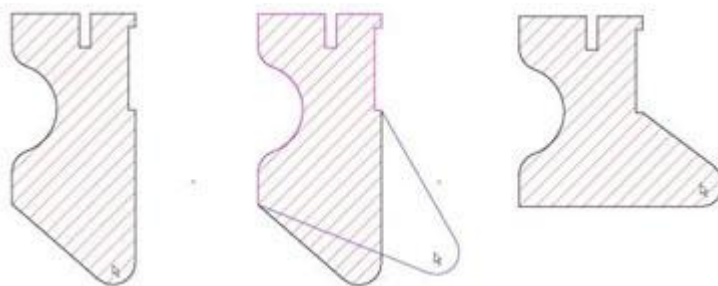


Рисунок 10.4 - Вариативное изменение конструкции

-применение плоской чертежной конструкторской части, которая функционирует в любой выбранной пространственной системе координат;

-создание пространственных проволочных объектов.

Вся функциональность модуля объемного моделирования деталей в системе ADEM может быть условно разделена на несколько групп:

-построение тел базовыми операциями (смещение, вращение, движение и т.п.);

-построение тел комплексными операциями (оболочки, гибка из листа, разъемы пресс-форм и т.п.);

-построение поверхностей (построение по сетке, слияние, затяжка и т.п.);

-конформные операции (перенос, поворот, копирование и т.п.);

-локальные (неконформные) операции (скругления, фаски, изменение положения вершин, восстановление острых углов и т.п.);

-булевы операции и их производные (объединение, дополнение, пересечение и т.п.);

-проецирование (проекции тел и поверхностей на плоскости, проекции линий на тела и поверхности);

-управление системами координат.

Все плоские элементы (линии, окружности, дуги, кривые, контуры, тексты и т.п.) могут быть расположены не только в плоскости, но и в пространстве. Позиционирование объектов может производиться как при их построении, так и в любой другой момент.

Для удобства позиционирования используется понятие "рабочая плоскость", которую можно устанавливать различными способами в динамическом режиме. Среди способов установки рабочей плоскости: основные плоскости, по трем точкам, касательно грани, через центр грани и т.п.

Рассмотрим несколько наиболее простых способов создания объемных тел.

Самой распространенной процедурой создания твердого тела является построение методом **смещения**.

Смещение (экструзия) - подъем группы профилей по заданному направлению. При этом на основе профиля строится объемная модель, любое горизонтальное сечение которой плоскостью соответствует исходному профилю (рисунок 10.5 слева). Другой особенностью является возможность задания положительных и отрицательных уклонов. При этом углы для внешних и внутренних контуров могут быть различными. Система автоматически определяет, из чего получается тело, а из чего отверстие (рисунок 10.5 справа).

Если боковые стенки учитывают изменение нормали на поверхности, от которой или к которой строится тело, то получим модель, представленную на рисунке 10.6 слева. При этом

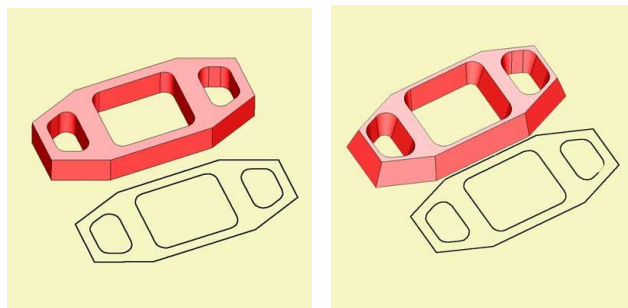


Рисунок 10.5 - Смещение и смещение с уклоном

также возможно задание уклона от нормали. Полезной может быть еще одна разновидность смещения - смещение с кручением. При этом в зависимости от высоты подъема исходного профиля производится разворот его относительно заданной оси (рисунок 10.6 справа).

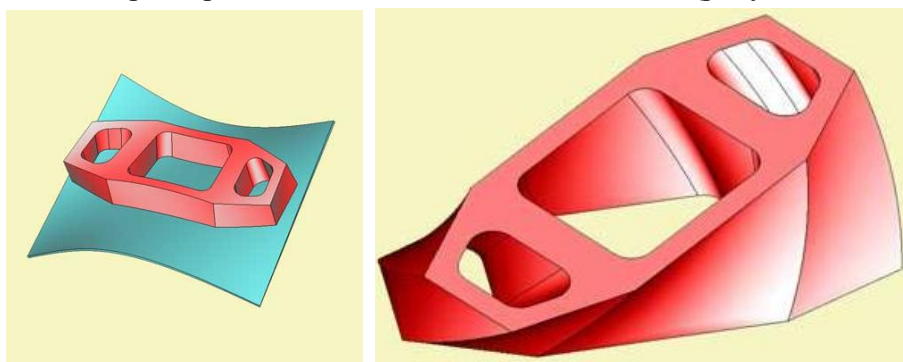


Рисунок 10.6 - Смещение с уклоном от нормали к поверхности и смещение с поворотом

Не менее часто применяется функция, которая имеет название Смещение вдоль направляющей. С помощью этой функции создаются модели движением профиля вдоль произвольных линейных, криволинейных или составных осей (рисунок 10.7). При этом положение профиля в каждой точке траектории перпендикулярно направляющей. Учет таких параметров как уклоны и нормали к поверхности в значительной степени вносят разнообразие в полученный результат. Одним из частных случаев такой функции является Смещение по спирали (рисунок 10.8).

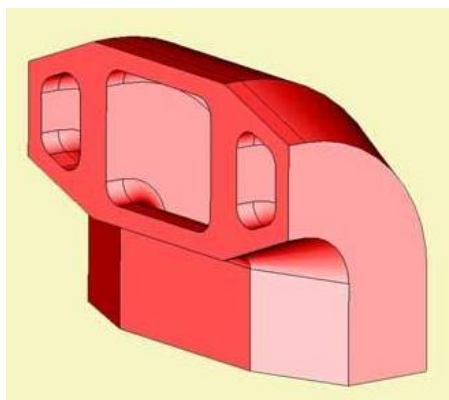


Рисунок 10.7 - Смещение вдоль направляющей

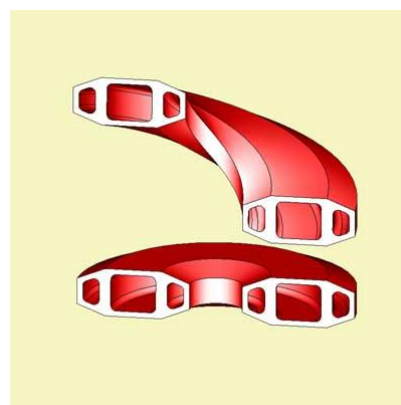


Рисунок 10.8 - Вращение и смещение по спирали

Многие модели содержат фрагменты в виде острых и сглаженных выступов, которые в системе ADEM могут быть созданы смещением типа Пирамида. В построении кроме

профилей участвуют два параметра, которые называются аспектами и влияют на характер сглаживания направляющей в начале и в конце объекта (рисунок 10.9).



Рисунок 10.9 - Смещение типа Пирамида

Комплексное смещение - когда композиционные (пересекающиеся) профили и профили с островами создают монолитные тела (рисунок 10.10а). Эта особенность экструзии в системе ADEM позволяет существенно сократить время моделирования. В комплексном смещении, как и в других процедурах, могут принимать участие фрагменты объемных тел. Это могут быть группы ребер и поверхностей (рисунок 10.10б). Направление экструзии соответствует направлению оси Z текущей рабочей плоскости, которая может и не совпадать с плоскостью профиля.

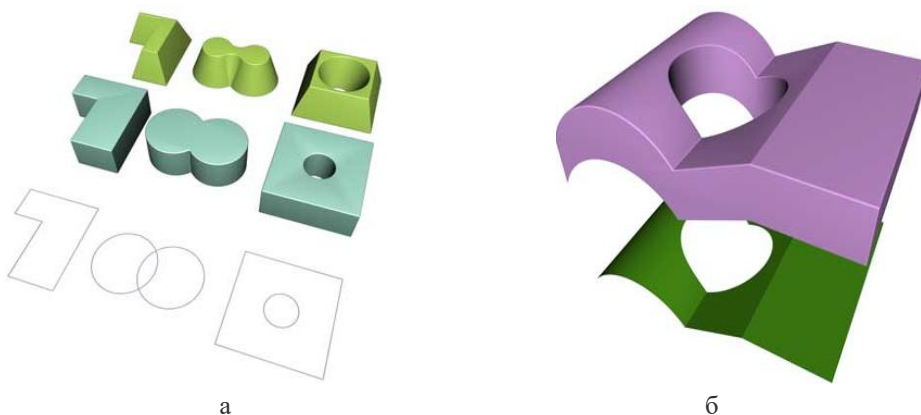


Рисунок 10.10 - Комплексное смещение профилей и поверхностей

Согласно принятому правилу, если профили являются замкнутыми контурами, то образуется твердое тело, если нет, то - поверхностная модель.

Вращение - вращение группы профилей относительно заданной оси. Точно так же, как и экструзия, вращение работает с группой профилей, учитывая пересечения и острова. Пользователь может задать необходимый угол и получить замкнутые или незамкнутые тела вращения произвольного сечения (рисунок 10.11 слева) .

Движение - движение группы профилей вдоль направляющей (рисунок 10.11 справа). Данная процедура также является комплексной. Направляющая может быть составной из отдельных элементов.



Рисунок 10.11 - Комплексные вращение и движение

Проволока и **Труба** - схожие процедуры, предназначенные для создания цилиндрических и конических тел с прямолинейной и криволинейной осью (рисунок 10.12). Труба имеет внутреннюю полость. Входные параметры: диаметр начала, диаметр конца, толщина стенки и направляющая (осевая линия). Направляющая линия может быть составной и негладкой.

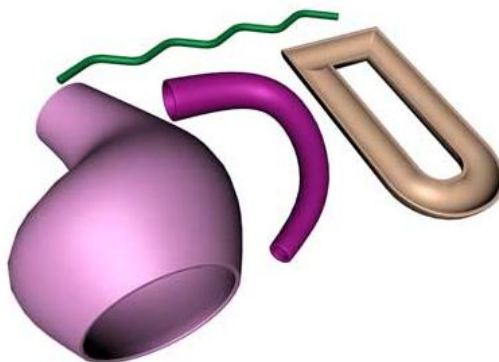


Рисунок 10.12 - Проволока и труба

Отверстие - разделка отверстий произвольной формы в одном или нескольких телах. Отверстия могут быть сквозными и глухими (рисунок 10.13 слева). В качестве параметров могут быть заданы глубина и угол конусности. Для сквозных отверстий положение профиля относительно тела не имеет значения. Для глухих отверстий глубина измеряется от плоскости профиля.

Добавление материала - создание приливов на теле (рисунок 10.13 справа). Имеются три разновидности: добавление движением профиля от тела, добавление движением профиля до тела и выдавливание части поверхности тела на заданную высоту. Все эти три метода позволяют эффективно создавать многопрофильные объекты, описывающие большой класс деталей.

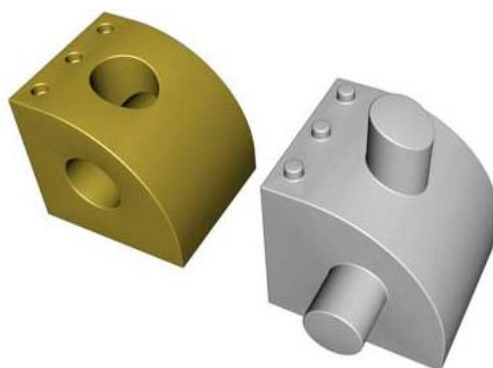


Рисунок 10.13 - Отверстия и добавление материала

Слияние - создание объектов, соединяющих тела. В зависимости от задаваемых аспектов напряженности, слияние может принимать различные формы от ломаного до плавного (рисунок 10.14). Переход от одной грани к другой учитывает направление касательных соседних граней, что позволяет создавать гладкие поверхности. Одна из разновидностей метода - радиальное слияние цилиндрических объектов. Обеспечивает сопряжение в виде гнутых труб, часто применяется для задач моделирования трубопроводов и трасс.

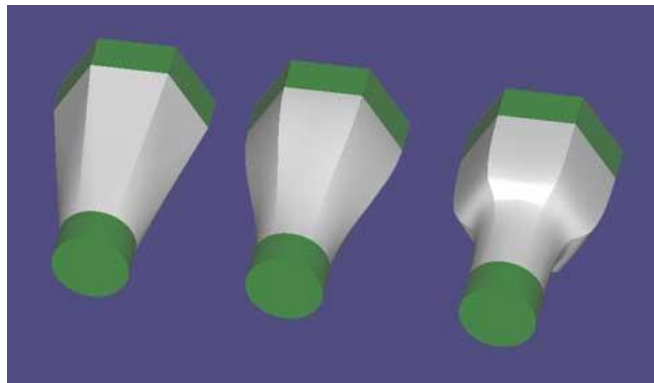


Рисунок 10.14 - Слияние

К модифицирующим операциям относятся: скругление постоянным и переменным радиусом, создание фасок, булевы операции и множество их комбинаций. Многие реализовано в виде комплексных процедур, как например, создание ярусных отверстий, построение тела по трем видам.

Скругление - важная функция для построения моделей. Основное назначение состоит в радиальном сглаживании острых углов (рисунок 10.15).



Рисунок 10.15 - Примеры скруглений

Задача построения скруглений является одной из самых сложных и самых нужных в объемном моделировании. По тому, как решена эта задача, определяют уровень системы моделирования.

Острые углы могут идти как вдоль ребра, по которому состыкованы поверхности, так и находиться в вершинах, где сходятся несколько ребер. Поэтому в системе ADEM реализовано универсальное сглаживание. Нередко требуется скругление переменным радиусом. Для этого достаточно указать характерные точки на ребре (ребрах) и задать значения радиуса.

Для скругления переменным радиусом важным является закон, по которому происходит изменение значения между контрольными точками. Не вдаваясь в подробности, скажем лишь, что ADEM формирует законы изменения оптимальным образом автоматически.

Логические операции - построение объемных тел на базе других объемных объектов. Одна из главных функций современного твердотельного моделирования. Основными процедурами являются: объединение, дополнение и пересечение.

Объединение создает одно тело из нескольких. **Дополнение**, более известное в конструкторской среде как **вычитание**, создает одно или несколько тел. Управление дополнением построено на простом условии, что из первого указанного тела вычитаются все остальные указанные тела. **Пересечение** тел создает также один или несколько объектов. При этом находятся общие части первого указанного тела с остальными.

Следует отметить, что в ADEM логические операции также применимы и к плоским фигурам и эффективно используются при черчении.

Построение по проекциям - создание тела по двум и трем проекциям. Большой класс объемных объектов может быть создан по главным проекциям (рисунок 10.16). Для этого проекции должны быть созданы по правилам начертательной геометрии и соответственно расположены на плоскости (соблюдая проекционные связи).

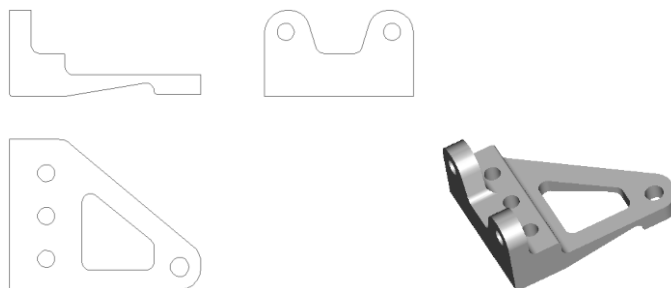


Рисунок 10.16 - Построение по проекциям

Как и все современные системы, ADEM имеет дерево построения объемной модели и возможность внесения в него изменений с последующей регенерацией модели.

Еще одна интересная особенность состоит в том, что в качестве профилей могут быть использованы проекции плоских элементов на объемные тела или поверхности. Для этого разработан обширный аппарат проецирования, куда входят прямые, нормальные и параметрические проекции, плюс еще накатка без искажения.

Там, где методы твердотельного моделирования не справляются с решением задач, там следует обращаться к локальным операциям. В ADEM включена довольно развитая функциональность, свойственная системам поверхностного моделирования. Причем переход от твердых тел к поверхностям и обратно здесь так прост и естественен, что пользователь даже не всегда осознает, какой методикой он пользуется в данный момент.

Отдельно выделенная поверхность или группа поверхностей в системе является равноправным объемным телом, с которым можно производить не только локальные: обрезка, продление, перетяжка и т.п., но и все остальные процедуры: булевы, скругления и т.д. Интересны в этом плане комплексные функции, основанные на гибридном моделировании, такие как восстановление острых углов и отсутствующих частей.

Одной из самых сложных процедур гибридного моделирования является изменение модели за счет изменения положения ее вершин. В ряде случаев эта возможность приносит большой эффект по сравнению с внесением изменений через дерево проекта. Следует отметить, что гибридное моделирование является одной из самых ярких черт системы ADEM.

Функциональность поверхностного моделирования обеспечивают в первую очередь операции построения: **по сечениям**, **по сетке**, **по треугольнику** и др. Для «вылечивания пустот» применяются процедуры **зашивки** отверстий и **затягивания** острых углов. Для расширения поверхностей предусмотрена функция **продление**.

На самом деле, в системе ADEM нет деления объектов на твердые тела и поверхности. С любым объемным объектом можно производить процедуры **обрезки**, **разделения**, **скругления** и другие, в том числе и логические.

Получение чертежей от объемной модели

При проектировании от объемной модели ADEM предоставляет возможность полуавтоматического построения чертежей на ее основе (рисунок 10.17). При этом действия пользователя эквивалентны традиционной логике проекционного черчения. Вначале конструктор осуществляет выбор необходимых ему основных видов из предлагаемого меню. Система автоматически создает соответствующие проекции с удалением или штриховым исполнением невидимых линий. Размещение проекции на бланке чертежа осуществляется простым указанием курсором на поле. Построение дополнительных видов, сечений, разрезов сводится к нанесению на поле чертежа стрелок видов и линий разрезов в обычном режиме плоского черчения. При этом система автоматически распознает пространственное положение этих объектов по отношению к модели и создает необходимые изображения, которые конструктор также помещает на чертеж, указывая их положение на поле.

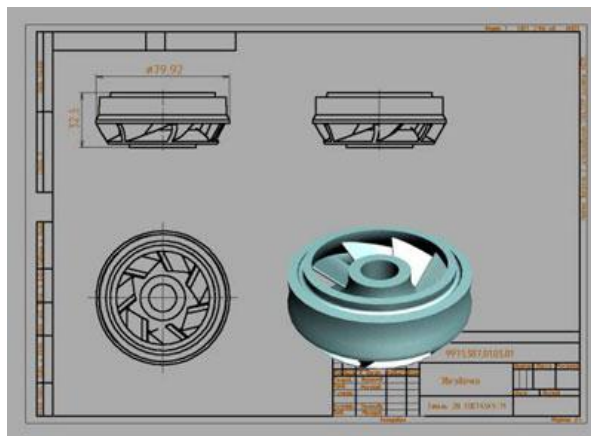


Рисунок 10.17 - Модель и чертеж в одном документе

Важно, что модель и чертеж имеют ассоциативную связь. Если модель изменяется (или заменяется), то виды, разрезы, сечения меняются автоматически. Изменение направления стрелок видов или линий сечений также приводят к изменению соответствующих видов и сечений.

Дальнейшее оформление чертежа производится тем же способом, что и при простом плоском черчении. Но при этом, система автоматически выстраивает связи элементов

оформления с моделью. Поэтому, при внесении изменений в модель, положение, например, условных обозначений, будет меняться соответственно.

Очень важным является тот факт, что работа конструктора ведется в едином пространстве модели и чертежа с единой логикой управления. Более того, возможность хранения модели и чертежей, относящихся к ней, в одном документе является серьезной базой для дальнейшей конструкторско-технологической работы над изделием.

Модификация объемных тел

Возможность модификации объемных моделей (рисунок 10.18) - одна из главных функций современного программного обеспечения для конструкторско-технологической подготовки производства.

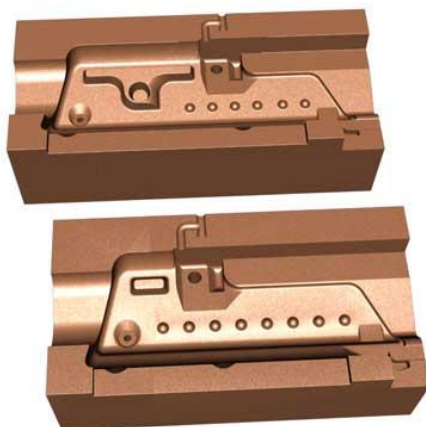


Рисунок 10.18- Редактирование моделей

Эффективность методов редактирования напрямую влияет на снижение времени и затрат на проектирование.

Серьезным шагом к оптимизации процесса редактирования стало проведение **локальных операций**. При локализации выделяется та часть модели, которая подвергается изменениям, и лишь этот фрагмент перестраивается заново, а затем присоединяется к модели (рисунок 10.19).

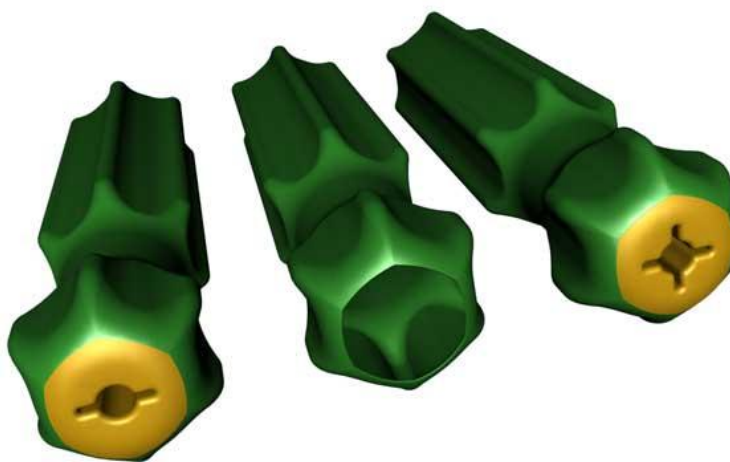


Рисунок 10.19 - Локализация внесения изменений

Для реализации метода потребовалось разработать процедуры разделения и сшивки, не нарушающие геометрию нередактируемой части. Недостатками метода локализации были и остаются:

- сложность (очень часто - невозможность) определения и выделения независимой области;
- проблемы с соблюдением граничных условий на стыке редактируемой и нередактируемой зон;
- трудоемкость процесса моделирования заново редактируемой части.

Поэтому данный метод эффективно применяется, в основном, в системах с развитыми функциями поверхностного моделирования.

Вторым серьезным шагом к автоматизации редактирования стал метод, использующий **историю создания** модели.

Вначале применялась его простая разновидность - откат до некоторого состояния, на котором создавалась исправляемая часть модели, и моделирование с учетом изменений, начиная с этого этапа (рисунок 10.20).

Это позволяло исключить повторение работы, проделанной до момента создания редактируемой части. Если же для внесения изменений необходимо вернуться далеко назад, то метод становится все более трудоемким.

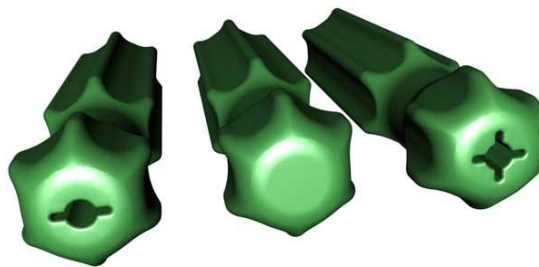


Рисунок 10.20 - Откат до этапа создания изменяемой части

Более эффективным является разновидность метода - **восстановление по измененной истории создания** - с автоматическим восстановлением геометрии модели по данным дерева истории. При этом система сама восстанавливает геометрию модели по информации, записанной в истории создания. Это позволяет исключить повторения всех действий пользователя, кроме, конечно, внесения самих изменений.

Возможность регенерации геометрии по истории сегодня является одним из самых распространенных способов автоматизации внесения изменений. Этот метод также очень часто называют параметрическим. Рассмотрим более подробно механизм его работы.

В отличие от геометрической и топологической моделей дерево истории содержит в себе не столько сами объекты, сколько ссылки на процедуры, производимые пользователем, параметры, вводимые пользователем, и ссылки на объекты, используемые при создании модели. Этих данных вполне достаточно, чтобы полностью восстановить геометрию изделия, для чего система должна лишь выполнить все действия согласно записанному протоколу, используя ссылки и параметры.

Если в истории изменить параметры, например, значения радиуса скругления или высоту смещения профиля, то в результате регенерации будет создана геометрия, отличная от

исходной. То же произойдет, если изменить геометрию профилей, на базе которых строились объемные тела (рисунок 10.21).

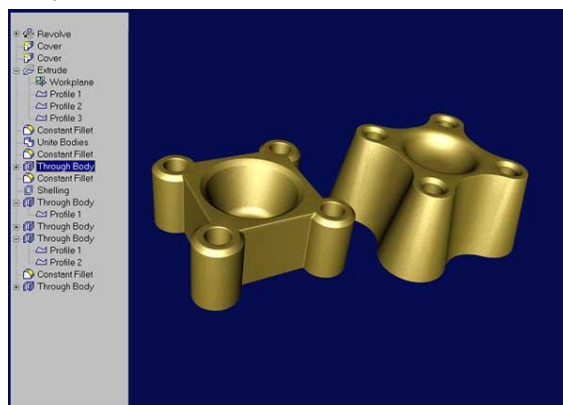


Рисунок 10.21 - Редактирование с использованием истории: изменение параметров

Для упрощения процесса выхода в нужную точку дерева истории применяются визуализаторы дерева, представляющие историю в графическом виде. В системе ADEM выйти в нужную точку можно как через визуализатор, так и просто, указав на модели элемент, который будет подвергнут изменениям.

В случае моделей высокой сложности, регенерация может занимать довольно большое время, так как система выполняет все действия заново, хотя и автоматически. Для оптимизации этого процесса в системе ADEM применен метод, схожий с методом локализации изменений, когда регенерируются только части, в которых происходит изменение. Остальные берутся в готовом геометрическом виде (рисунок 10.22).

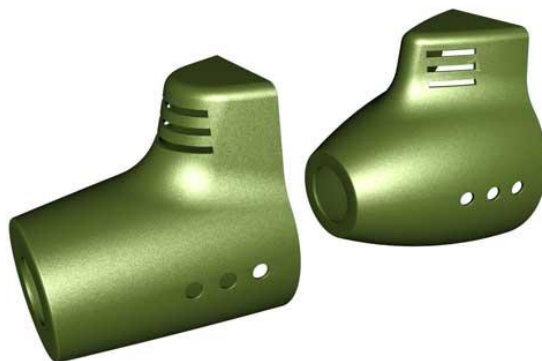


Рисунок 10.22 - Редактирование с использованием истории: изменение параметров и профайлов

Редактирование с использованием истории является эффективным методом, применяется как в системах поверхностного, так и твердотельного моделирования, но этот метод имеет два серьезных недостатка.

Первый недостаток - регенерация модели не всегда возможна после внесения изменений. Это происходит тогда, когда изменение требует отличного от записанного в историю протокола воссоздания. В этом случае система восстановит геометрию лишь частично. В некоторых случаях исправить ситуацию можно перестановкой действий в протоколе. Иногда требуется замена одних процедур на другие. Могут быть случаи, когда нельзя обойтись без введения дополнительных событий в историю.

Второй, и очень серьезный, недостаток - невозможность обмена историями между различными системами.

Возможен ли другой способ редактирования "чужих" моделей, кроме как строить их заново или кропотливо заниматься локальными изменениями?

Одним из методов, предназначенных для редактирования импортированных моделей в системе ADEM, является **восстановление по измененному каркасу**.

Геометрическая модель в простейшей интерпретации состоит из поверхностей, ребер, и вершин. Если модель твердотельная, то присутствует еще и топология, описывающая взаимосвязи между этими объектами. Идея метода проста - производить изменение положения вершин в пространстве, после чего восстанавливать по ним модель. Вносить изменения можно двумя способами: корректируя положение вершин вручную, например процедурой "перенос", или корректируя вершины параметрически, изменяя значения размеров (рисунок 10.23).

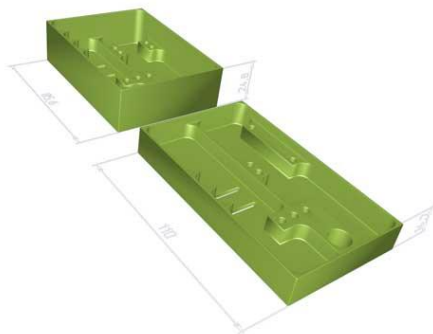


Рисунок 10.23 - Восстановление по измененному каркасу: ручное и параметрическое

Метод применим для широкого класса деталей и типов изменений, удобен и понятен для пользователей. Но, как и все в этом мире, имеет ряд ограничений. Например, его невозможно использовать для редактирования скруглений, так как после процедуры скругления теряются данные об исходном ребре и его вершинах.

Для редактирования "чужих" скруглений в системе ADEM применяют функции **восстановление каркаса**. Одна из таких процедур называется "Острый угол". Пользователь указывает поверхности, принадлежащие скруглению, и система восстанавливает угол, каким он был до скругления (рисунок 10.24). Теперь можно задать новое значение радиуса и получить измененный результат.

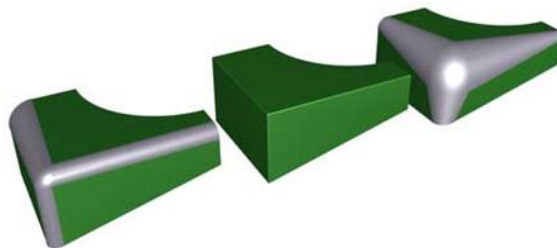


Рисунок 10.24 - Изменение скругления функцией "Острый угол"

Функция "Острый угол" довольно проста в применении, но относительно сложна в реализации, как и любая задача, связанная с восстановлением утраченной информации. Очень

часто имеется не одно, а несколько решений. Но во многих случаях функция позволяет получить хороший результат.

Еще более серьезные проблемы имеет ныне модный метод - **восстановление истории распознаванием фичерсов**. По исходной импортированной геометрии предполагается распознать историю создания и построить дерево истории. Не будем подробно останавливаться на этом методе, скажем только, что сегодня не существует ни одного продукта, который бы удовлетворял реальным задачам. На деталях, которые можно описать кубиками и цилиндрами, еще можно получить какие либо результаты, но не более.

Распознавание фичерсов имеет вероятностный характер и может не создавать рационального, с точки зрения пользователя, представления модели. Например, любой объект может быть описан системой сфер (voxселов). Но что будет делать пользователь со множеством этих шариков для редактирования модели?

Разработчики ADEM со своими партнерами работают над другим методом, который получил название **восстановление истории назначением фичерсов**. Суть метода состоит в том, что пользователь сам назначает фичерс перед редактированием. Например, если он хочет отредактировать скругление, то указывает на поверхности, входящие в это скругление и назначает им фичерс - "скругление". После этого данные поверхности редактируются как скругление заданного радиуса.

Специальные функции

Рассмотрим некоторые специальные функции объемного моделирования системы ADEM, предназначенные для проектирования изделий и инструмента.

Основу многих изделий машиностроения составляют оболочечные конструкции. Как правило, это оболочки постоянной или переменной толщины, подкрепленные ребрами и имеющие наплывы материала для посадочных мест (рисунок 10.25).

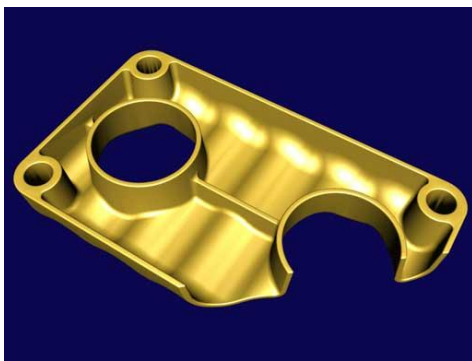


Рисунок 10.25 - Оболочка с ребрами и наплывами

Создание оболочки удобно производить на основе модели монолитного прототипа. Очевидным способом является построение эквидистант к его поверхностям (рисунок 10.26) и последующее объединение поверхностей прототипа и эквидистант в оболочку.

Кажется что все просто, но эквидистантные поверхности могут не соединяться друг с другом или пересекать друг друга. При этом необходимо решать задачи обрезки и достраивания отсутствующей геометрии. Обрезка вручную - довольно трудоемкая задача. Восстановление отсутствующей геометрии - еще более сложная проблема. Математический аппарат системы ADEM позволяет решать задачу построения оболочек автоматически.

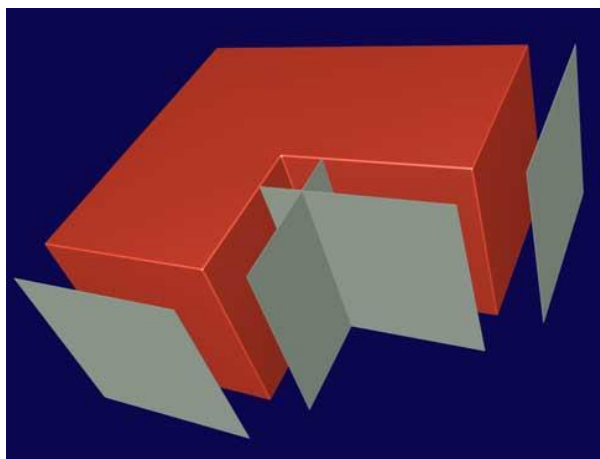


Рисунок 10.26 - Разрывы и пересечения эквидистант

Например, для построения оболочки постоянной толщины достаточно задать два параметра: высоту и глубину расположения материала относительно поверхности прототипа. При этом будет получено твердое тело оболочки (рисунок 10.27).

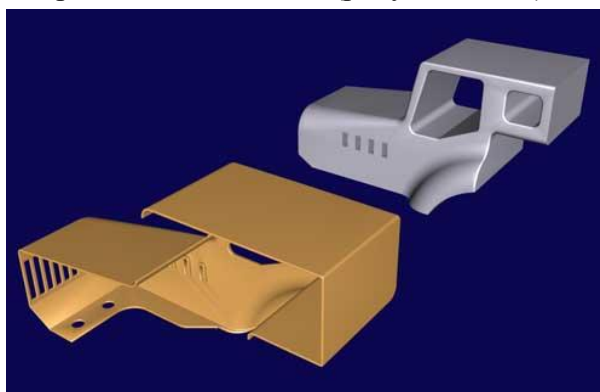


Рисунок 10.27 - Автоматическое построение оболочки по монолитному прототипу

Параметры высоты и глубины могут варьироваться в отрицательной и положительной областях. При этом оболочка будет строиться либо внутри монолита, либо вне его, либо частично внутри и снаружи. При равных параметрах высоты и глубины получается оболочка с симметричным расположением материала относительно поверхности прототипа.

Существует несколько методов построения ребер (рисунок 10.28). Наиболее простой - создание их как отдельных тел и далее удаление лишнего обрезкой. После чего производится объединение в единое тело с оболочкой и построение необходимых скруглений.

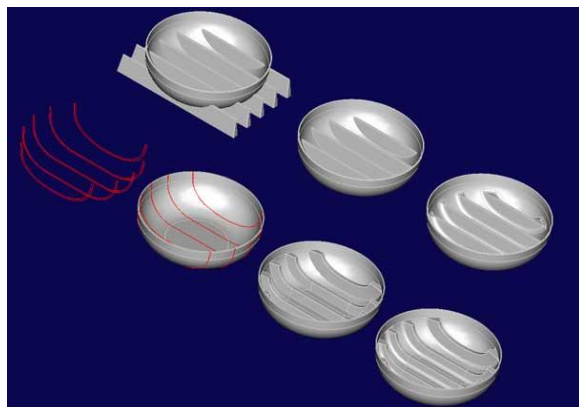


Рисунок 10.28 - Построение ребер двумя способами

Более сложный метод - использование пространственных кривых на поверхности оболочки. Ребра строятся движением профиля по этим пространственным кривым. Далее – аналогично предыдущему способу.

Построение наплывов похоже на построение ребер. Наплыв может быть выполнен в виде отдельного тела с последующей обрезкой лишнего и объединения с оболочкой. Более автоматизированный способ - экструзия профиля до оболочки.

Интересны также способы "выдавливания" материала оболочки. При этом происходит поднятие части поверхности оболочки на заданную высоту. Метод эффективен для сложных профильных приливов, например для текстовых оттисков и рисунков (рисунок 10.29).

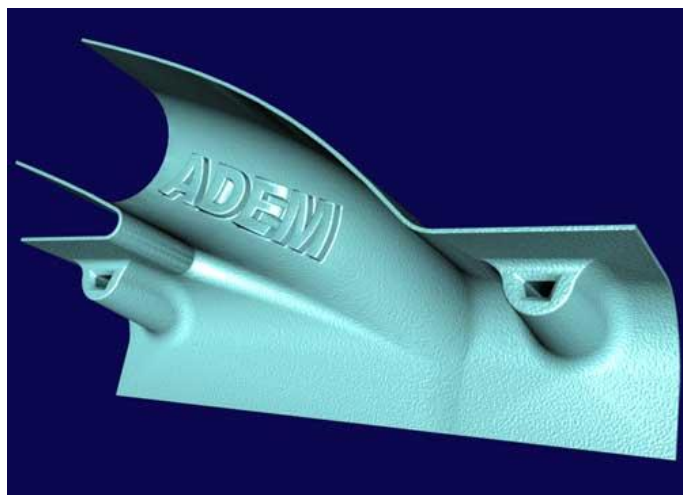


Рисунок 10.29 - Наплывы на сложной поверхности

К специальным функциям для инструментального производства можно отнести также построение активной части пресс-форм. Дело в том, что изделия, изготавливаемые объемной штамповкой, литьем под давлением и другими подобными методами, должны быть спроектированы с учетом особенностей данной технологии.

Их реальная геометрия, как и геометрия формообразующей оснастки, имеет специфические особенности. Укажем лишь одно условие, которое в основном влияет на геометрию детали и пресс-формы - деталь должна выниматься из пресс-формы.

Для более эффективного проектирования пресс-форм в ADEM разработана специальная функциональность автоматического и полуавтоматического создания поверхностей разъема и модификации моделей по заданным уклонам. Эта технология отражает последние достижения программного обеспечения для производителей оснастки и соответствует самым современным требованиям.

Как было отмечено, многие задачи объемного моделирования не могут быть решены методами твердых тел. Необходима работа на более низком уровне топологии с отдельными поверхностями или группами поверхностей.

Объемная модель в ADEM представлена таким образом, что пользователь может применять к ней оба подхода одновременно. Для этого существуют две простые функции разборки тела на отдельные поверхности и сборки группы поверхностей в объект - твердое тело (рисунок 10.30).



Рисунок 10.30 - Представление твердого тела системой поверхностей

Построение тел комплексными операциями

Эти функции основаны на создании объектов не только и не столько на основе профилей, сколько с помощью ранее созданных объемных тел. Рассмотрим несколько примеров применения комплексных операций в моделировании деталей для конструкторско-технологической подготовки производства.

В целях уменьшения массы и экономии конструкционных материалов многие изделия проектируются в виде тонкостенных оболочек. Моделировать такие конструкции удобно на основе объемных тел, представляющих внешний облик изделия (рисунок 10.31)

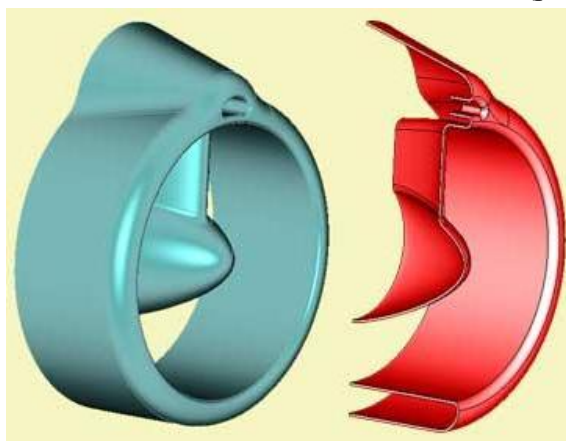


Рисунок 10.31 - Построение оболочечной конструкции функцией Оболочка

В системе ADEM оболочки на основе твердых тел строятся автоматически. При этом пользователь может численно задать толщину в виде верхней и нижней границ материала оболочки относительно поверхности исходного тела. Таким способом определяется и расположение оболочки по отношению к телу.

Нередко при построении оболочки должны учитываться радиусы скруглений в тех местах, где на исходном теле есть негладкие переломы. Для этого достаточно ввести значение радиуса, который система ADEM будет учитывать при создании оболочки.

В большинстве случаев, нужно строить оболочку не ко всему телу, а к некоторым его участкам. Для такого варианта построения достаточно указать те поверхности, которые не должны участвовать в процедуре.

Если возможностей функции Оболочка недостаточно для решения задачи автоматически, то можно работать и на более низком уровне с помощью построения эквидистант к телам и отдельным поверхностям с последующей ручной обрезкой и сшивкой.

Еще одна интересная функция связана с тонкостенными конструкциями - Формовка. Для ее выполнения необходимы два объекта - модель листа и модель инструмента. В результате модель листа будет деформирована инструментом с учетом штамповочного радиуса (рисунок 10.32).

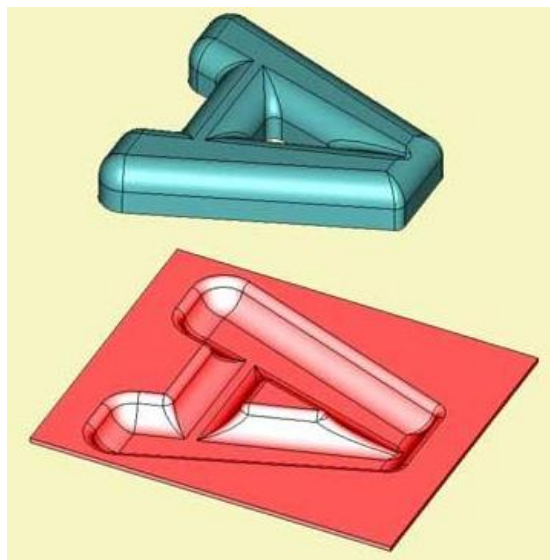


Рисунок 10.32 - Выдавливание листа

К комплексным процедурам относится также функция Отверстие. В зависимости от введенных параметров система автоматически создает составные отверстия в объемных телах. Направление оси отверстия может совпадать с нормалью к поверхности (рисунок 10.33) или в направлении, определенном пользователем.

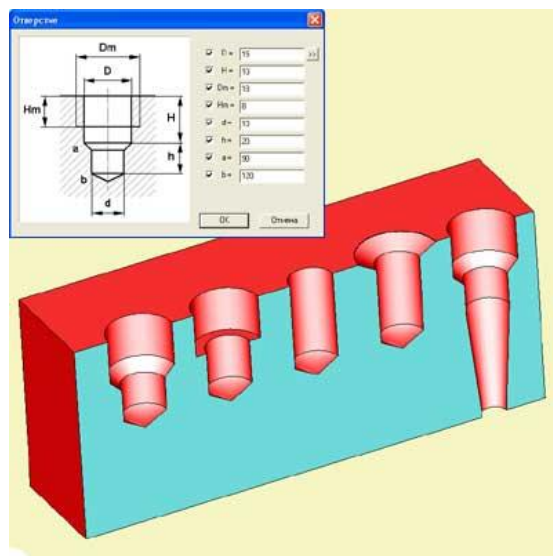


Рисунок 10.33 - Функция Отверстие

Что касается создания технологичной модели для штамповки, то очень полезным является функционал создания штамповочных (литейных) уклонов в исходном теле. В ADEM есть полностью автоматическая процедура создания уклонов (рисунок 10.34). Для ручного построения применяется процедура Построение поверхности уклона.

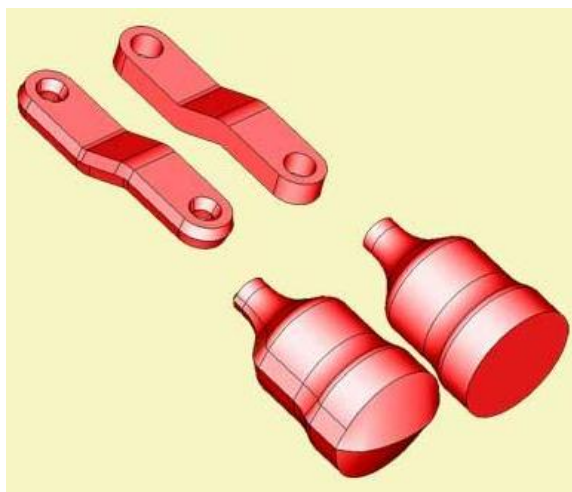


Рисунок 10.34 - Функция Сделать уклоны

Мы рассмотрели несколько примеров группы комплексных функций. Все они основаны на нескольких базовых процедурах. Так, например, функция Оболочка содержит следующие алгоритмы: построения эквидистанты к поверхности, обрезки поверхности другими поверхностями, продления поверхности, сшивки поверхностей.

Принцип адекватности воздействия на объекты

На примере функции обрезки проиллюстрируем принцип адекватности и взаимодействия твердых тел и поверхностей в системе ADEM. С помощью профилей можно разрезать или вырезать часть из твердого тела или поверхности (рисунок 10.35). Эта функция строит по контурам поверхности, которыми и обрезает и вырезает указанные объекты.

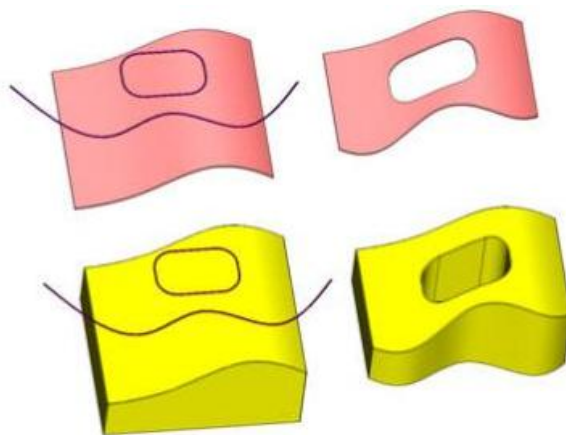


Рисунок 10.35 - Обрезка профилями

Взаимодействие между твердыми телами и поверхностями осуществляется независимо от их сути, а лишь по назначению пользователем того, что из них является объектом действия, а что инструментом (рисунок 10.36).

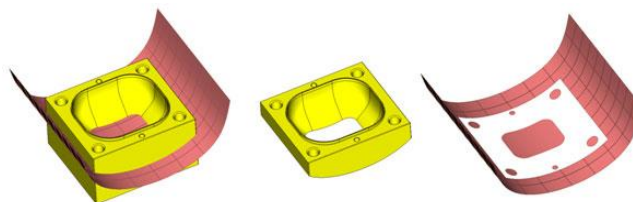


Рисунок 10.36 - Обрезка поверхностью и телом

Функция **замыкание тела** предназначена для восстановления отсутствующей геометрии в твёрдом теле. Пример восстановления показан на рисунке 10.37.

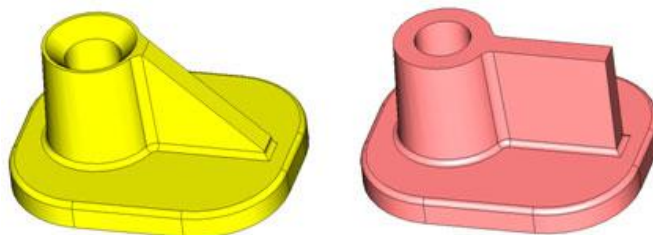


Рисунок 10.37 - Восстановление геометрии с помощью функции Замыкание Тела

Комбинации рассмотренных функций открывают широкие возможности редактирования объемных моделей различной степени сложности.

Полезным так же является отдельный класс процедур, связанных с получением проекций. Речь идет о проецировании объемных моделей на различные плоскости и проецировании профилей на объемные модели.

Первая группа предназначена в основном для получения проекций в виде плоских объектов, которые далее можно использовать для привязок, или в качестве профилей для построения объемных тел или для других целей. К примеру, можно сделать проекцию одной или нескольких граней, потом внести в эти проекции изменения или дополнения и на базе их построить тело или отверстие и т.п.

Вторая группа создает пространственные кривые на поверхности объемных тел различными способами проецирования. Сюда входит прямое, параметрическое проецирование, накатка и другие алгоритмы. Данные проекции также можно использовать в качестве профилей для создания трехмерных тел. Так, например на рисунке 10.38 показано нанесение насечки и символического клейма на модель рукоятки.

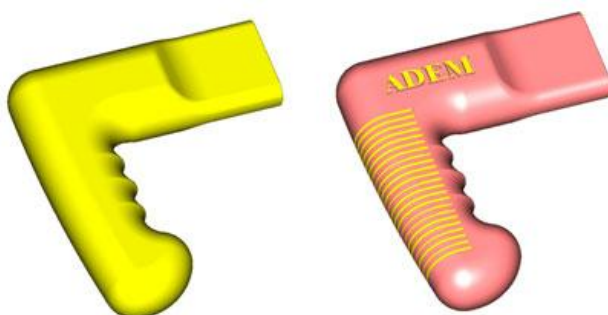


Рисунок 10.38 - Насечка и клеймо, выполненные с помощью процедуры Проецирование

Для конструкторско-технологического моделирования деталей необходим и специальный математический аппарат, который далеко не всегда присутствует в современных системах твердотельного моделирования. Покажем это на простом примере.

Допустим, есть деталь в виде куба, построенная на основе квадратного профиля (рисунок 10.39). Благодаря дереву построений, которое в твердотельных системах обеспечивает регенерацию модели по измененным параметрам, можно производить следующее редактирование: изменять высоту куба, изменять уклон, превращая куб в усеченную пирамиду (синий объект), изменять профиль и получать объекты иных форм (красный объект). Но с помощью дерева построений мы не сможем получить из куба желтый объект, хотя он отличается от исходного измененным положением всего одной вершины.

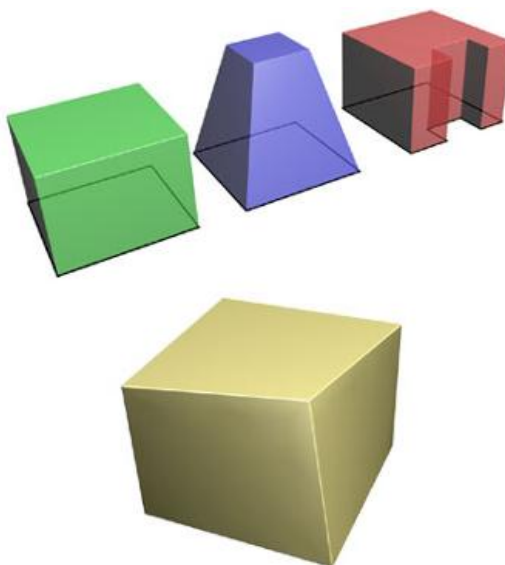


Рисунок 10.39 - Параметрическое (синий и красный) и непараметрическое (желтый) изменение модели

Чтобы произвести изменение модели (как показано на рисунке 10.40) традиционными способами, на первом шаге нужно: удалить сферическую поверхность и поверхности сопряжения, произвести затяжку образовавшегося отверстия поверхностью. В ADEM эта процедура автоматизирована и производится одним действием.

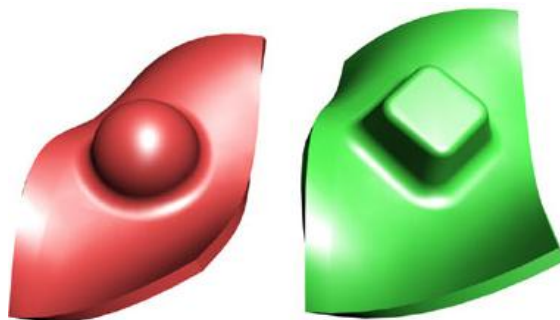


Рисунок 10.40 - Редактирование на основе автоматизированных процедур гибридного моделирования

Теперь о принципиально иных моделях, которые все чаще и чаще используются для подготовки производства. Речь идет об изделиях, описываемых так называемыми скульптурными поверхностями. В отличие от моделей, разработанных в CAD системах, эти поверхности, как правило, получают в результате объемного сканирования материальных изделий.

Современные системы объемного сканирования имеют на выходе файл в формате STL. Это не что иное, как триангулированное представление поверхности объекта. Чтобы обеспечить необходимую точность, сканирование производится с мелким шагом. Так например, поверхность (рисунок 10.41) состоит из 300 тысяч треугольных граней.

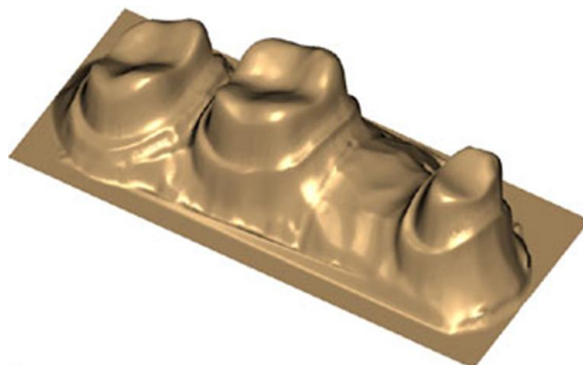


Рисунок 10.41 - Скульптурная поверхность

В более ранних версиях система ADEM воспринимала STL файл как обычное твердое тело, с которым можно было производить все возможные действия (скругления, булевы операции и т.п.). Очевидно, что с моделями, содержащими не одну тысячу граней, работать подобным образом немыслимо.

Поэтому в ADEM был добавлен новый функционал для эффективной работы с подобными объектами, который позволяет производить вариативное и параметрическое редактирование.

Прямое редактирование

В принципе любая CAD система позволяет вносить изменения в импортированный объект. Например, можно сделать скругление на ребре, можно просверлить отверстие, можно добавить материал и т.п.

Сложность возникает, когда нужно изменить радиус уже существующего скругления, убрать, передвинуть или изменить диаметр отверстия и т.д. Вот с такого рода задачами дело обстоит куда сложнее. С точки зрения истории построения модели данное редактирование относится не к добавлению новых этапов к дереву построений, а к изменению тех, что уже в нем есть. И это при отсутствии самого дерева!

Этот метод позволяет, например, выбрать сферическую область на теле и сказать, что это есть призма. При этом, вместо сферической части появится призматическая, с заданными параметрами. Подобная схема основана на замене одного фрагмента другим.

Кроме процедуры замены для редактирования твердотельных объектов нужны и другие возможности. Их перечень соответствует стандартному набору операций с самими телами: перенос, поворот, копирование, удаление и т.п. В системе ADEM эти операции, применимы и к фрагментам тел, и составляют основу прямого редактирования.

Рассмотрим ряд примеров использования прямого редактирования в ADEM (рисунки 10.42 - 10.46).

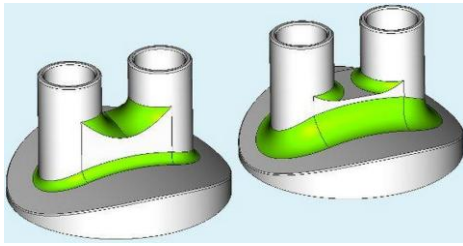


Рисунок 10.42 - Изменение радиусов скруглений

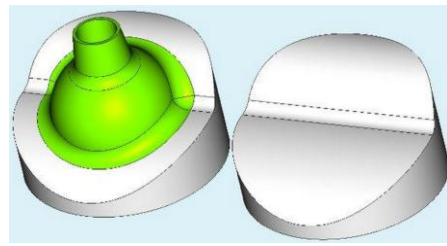


Рисунок 10.43 - Удаление фрагмента

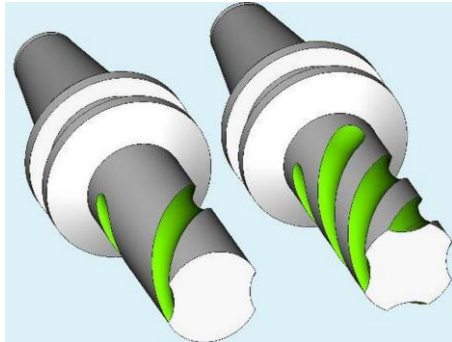


Рисунок 10.44 - Изменение числа
фрагментов

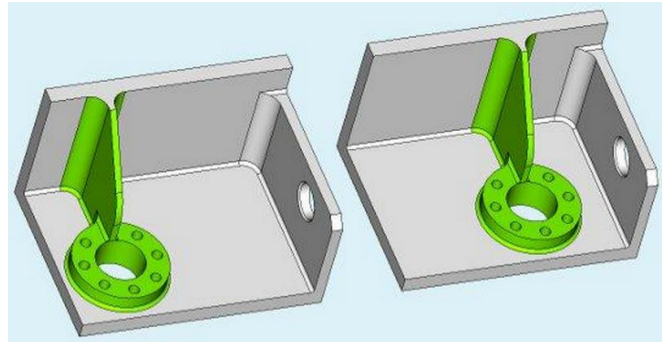


Рисунок 10.45 - Перемещение фрагмента

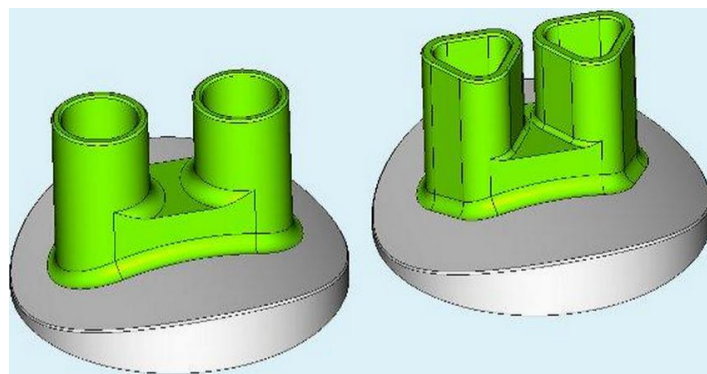


Рисунок 10.46 - Замена фрагмента

Следует заметить, что в этих примерах меняется не только геометрия тела, но и его топология. Иными словами, с помощью прямого редактирования фрагментов можно выполнять довольно глубокие изменения модели.

В системе ADEM фрагментом может быть объявлена группа поверхностей, имеющая смысловое для пользователя значение. Причем фрагменты могут иметь практически любой уровень сложности.

Существует еще один аспект, связанный с редактированием импортируемых моделей. Дело в том, что не все системы работают с твердыми телами. Точнее сказать, что не все модели могут содержать граничное представление геометрии в виде BRep-структуры (трехмерное тело представляется набором связанных друг с другом поверхностей, задающих границу между представляемым телом и остальным пространством).

В системе ADEM предусмотрен аппарат, восстанавливающий данную информацию на основе исходной геометрии. Он имеет двойную направленность, так что всегда твердое тело можно разобрать и собрать из поверхностей. Имеется в виду разборка твердого тела не только

на составляющие поверхности (см. рисунок 10.30), но и разборка тела на составляющие твердые тела (рисунок 10.47) независимо от наличия параметризации и истории построения.

Производя различные операции над такими фрагментами и варьируя их комбинации, можно добиваться кардинального изменения геометрии модели. Причем использование комбинации твердых тел в значительной мере упрощает прямое редактирование, так как не требует трудоемких работ по сшивке поверхностей и затяжке щелей.

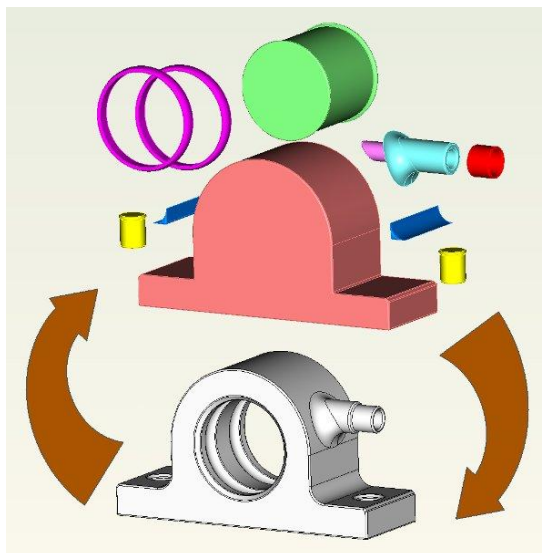


Рисунок 10.47 - Представление твердого тела системой твердых тел

Вопросы для самоконтроля

1. Приведите кратко историю развития системы CAD/CAM/CAPP ADEM.
2. Перечислите функциональные возможности (18) системы CAD/CAM/CAPP ADEM.
3. Перечислите функциональные возможности (7) CAD модуля системы ADEM.
3. Поясните, какие схемы используют при создании 2D объектов.
4. Перечислите основные возможности (8 групп) модуля объёмного моделирования ADEM.
5. Приведите примеры (схемы) реализации функций смещения, смещения с уклоном, смещение с уклоном от нормали, смещение с поворотом, смещение вдоль направляющей и вращение и смещение по спирали при моделировании твёрдых тел.
6. Объясните, как реализуется смещение типа Пирамида с учётом аспектов.
7. Приведите примеры реализации комплексного смещения профилей и поверхностей.
8. Приведите примеры (схемы) реализации функций вращения, движения, "проволоки" и "трубы".
9. Приведите примеры (схемы) реализации функций формирования отверстий и приливов.
10. Приведите примеры (схемы) реализации функций слияния и скругления.
11. Приведите примеры (схемы) реализации логических (булевых) операций.
12. Перечислите функции, используемые при моделировании поверхностей. В чём состоит отличие функций моделирования твёрдых тел и поверхностей?
13. Перечислите дополнительные функции поверхностного моделирования (6).

14. В чём проявляется реализация ассоциативности 3D и 2D объектов?

15. Приведите примеры (схемы) реализации методов редактирования (модификации) объёмных тел: проведения локальных операций, использование истории создания (дерева) модели, восстановление по изменённому каркасу, восстановление каркаса (функция "острый угол"), восстановление истории распознаванием и восстановление истории назначением фичерсов.

16. Приведите примеры (схемы) реализации специальных функций: автоматическое построение оболочек, построение сложных рёбер и наплывов, построение геометрии активной части пресс-форм.

17. Приведите примеры (схемы) реализации комплексными операциями: формовка, отверстие, формирование уклонов.

18. Объясните принцип адекватности воздействия на объекты. Приведите примеры.

19. Приведите примеры (схемы) реализации функций проецирования объёмных моделей на плоскости и поверхности.

20. Приведите примеры (схемы) реализации прямого редактирования: изменение радиусов скруглений, перемещение, поворот, удаление фрагмента, копирование (изменение числа фрагментов), замена фрагмента.

Список литературы

1. Ли, К. Основы САПР (CAD/CAM/CAE). Спб.: Питер, 2004. - 560 с.: ил.
2. Малюх, В.Н. Введение в современные САПР: Курс лекций. - М.: ДМК Пресс, 2010. - 192 с.: ил.
3. Ушаков, Дм. На ядре http://isicad.ru/ru/articles.php?article_num=14210, 18. 01.2011.
4. Технический рисунок: учебно-методическое пособие/ Писканова Е.А. – Тольятти : ТГУ, 2011. – 122 с.: ил.
4. Быков, А.В. и др. ADEM CAD/CAM/TDM. Черчение, моделирование, механообработка / А.В. Быков, В.В. Силин, В.В. Семенников, В.Ю. Феоктистов - СПб: БХВ-Петербург, 2003. - 320 с.: ил.