

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования «Амурский государственный университет»**

Кафедра Информационных и управляющих систем
(наименование кафедры)

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Моделирование в САПР
(наименование дисциплины)

Основной образовательной программы по направлению подготовки:

230100.68 «Информатика и вычислительная техника» по магистерской программе
«Компьютерное моделирование»
(код и наименование направления)

Благовещенск 2011

УМКД разработан _____
(степень, звание, фамилия, имя, отчество разработчиков)

Рассмотрен и рекомендован на заседании кафедры

Протокол заседания кафедры от « ____ » _____ 201__ г. № ____

Зав. кафедрой _____ / _____ /
(подпись) (И.О. Фамилия)

УТВЕРЖДЕН

Протокол заседания УМСС _____
(указывается название специальности (направления подготовки))

от « ____ » _____ 201__ г. № ____

Председатель УМСС _____ / _____ /
(подпись) (И.О.Фамилия)

1. Рабочая программа учебной дисциплины

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Амурский государственный университет»

УТВЕРЖДАЮ
Проректор по учебной работе
_____ В.В. Проказин
« ___ » _____ 20__ г.

РАБОЧАЯ ПРОГРАММА

МОДЕЛИРОВАНИЕ В САПР

Направление подготовки 230100.68 «Информатика и вычислительная техника»
по магистерской программе «Компьютерное моделирование»
Квалификация (степень) выпускника – магистр
Специальное звание – магистр-инженер

Курс – 1

Лекции – 18 (час.)

Практические (семинарские) занятия – нет

Лабораторные занятия – 54 (час.)

Самостоятельная работа – 108 (час.)

Общая трудоемкость дисциплины – 180 (час.), 5.0 (з.е.)

Курсовая работа (проект) – нет

Составитель – А.В. Бушманов, доцент, канд. техн. наук

Факультет математики и информатики

Кафедра информационных и управляющих систем

Семестр – 1, 2

Экзамен – нет

Зачет – 1, 2

1 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины: изучение базовых функций современных систем автоматизированного проектирования, конструирования, производства и управления жизненным циклом изделия; знакомство с математическими методами и алгоритмами используемыми в системах автоматизированного проектирования.

Задачи дисциплины:

- дать анализ современных методов автоматизированного проектирования в АСУ;
- подготовить магистрантов к участию в разработке автоматизированных систем управления.

2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Дисциплина относится к циклу профессиональных дисциплин вариативной части (М2.В.ОД.2) Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 230100 «Информатика и вычислительная техника» (квалификация (степень) «магистр»).

Для успешного освоения данной дисциплины необходимы знания, умения и навыки, приобретенные в результате освоения дисциплин базовой части общенаучного цикла, математического и естественнонаучного цикла (М.1 и М.2) Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 230100 «Информатика и вычислительная техника» (квалификация (степень) «бакалавр»): математический анализ; методы оптимизации, автоматизированные комплексы информационных систем, имитационное моделирование; проектирование автоматизированных систем обработки информации и управления.

Знания, умения и навыки, приобретенные в результате освоения данной дисциплины найдут применение в научно-исследовательской работе магистранта (НИРМ.Ф.1) Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 230100 «Информатика и вычислительная техника» (квалификация (степень) «магистр»), а также прохождения научно-исследовательской практики и выполнения научно-исследовательской работы (М.3).

3 КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования (ПК-1, ПК-5):

1) Знать: базовую функциональность автоматизированных систем проектирования; основы 3D моделирования, вариационного моделирования на основе конструктивных элементов; методики решения задач в системах автоматизированного проектирования.

2) Уметь: проводить анализ используемых алгоритмов и методов.

3) Владеть: методами компьютерного моделирования в системах автоматизированного проектирования.

4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 5 зачетные единицы, 180 час.

№ п/ п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость в часах				Формы текущего контроля успеваемости Форма промежуточной аттестации
				Лек	Пр	Лаб	Сам	
1	Общая характеристика систем автоматизированного проектирования.	1	1-2	2	0	4	12	Защита лаб. работы
2	Геометрическое моделирование.	1	3-4	2	0	6	12	Защита лаб. работы
			5-6	2	0	6	12	Защита лаб. работы
3	Вариационное моделирование.	2	7-8	2	0	6	12	Защита лаб. работы
			9-10	2	0	6	12	Защита лаб. работы
4	Методы поиска и оптимизации решения.	2	11-12	2	0	6	12	Защита лаб. работы
			13-14	2	0	6	12	Защита лаб. работы
5	Инженерный анализ методом конечных элементов.	2	15-16	2	0	6	12	Защита лаб. работы
			17-18	2	0	6	12	Защита лаб. работы
6	Всего по разделам	2	1-18	18	0	54	108	Зачет

5 СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

5.1 Лекции

5.1.1 Раздел 1. Общая характеристика систем автоматизированного проектирования. Исторический обзор развития систем автоматизации проектирования; современные системы геометрического моделирования и их классификация; системы инженерного анализа; системы технологической подготовки производства; системы автоматизации производства; системы управления данными об изделии.

5.1.2 Раздел 2. Геометрическое моделирование.

Автоматизация чертежных работ и геометрическое моделирование функции твердотельного моделирования; декомпозиционные и конструктивные модели; корректность граничных моделей; пакеты геометрического моделирования.

5.1.3 Раздел 3. Вариационное моделирование.

Параметры, ограничения и вариационные модели; вариационный геометрический решатель; способы алгебраического моделирования геометрической задачи; метрический тензор геометрической задачи.

5.1.4 Раздел 4. Методы поиска и оптимизации решения.

Задачи удовлетворения ограничениям и оптимизации в общей постановке; классификация методов поиска и оптимизации решения.

5.1.5 Раздел 5. Инженерный анализ методом конечных элементов.

Введение в метод конечных элементов; тензор деформаций; тензор напряжений; матрицы жесткости и упругости; уравнения равновесия; общая схема конечно-элементного анализа.

5.2 Лабораторные занятия

5.2.1 Лабораторная работа 1. Анализ систем автоматизации проектирования.

5.2.2 Лабораторная работа 2. Геометрическое моделирование с использованием CAD-систем.

5.2.3 Лабораторная работа 3. Моделирование конструкций с использованием CAD-систем.

5.2.4 Лабораторная работа 4. Программирование на языке Auto LISP, построение моделей.

5.2.5 Лабораторная работа 5. Использование стандарта обмена геометрическими данными.

5.2.6 Лабораторная работа 6. Моделирование в системе ABAQUS (уч.версия).

5.2.7 Лабораторная работа 7. Моделирование в системе ANSYS (уч.версия).

5.2.8 Лабораторная работа 8. Разработка программы для анализа деформации детали в среде MathLab.

5.2.9 Лабораторная работа 9. Интеграция CAD и CAE систем.

6 САМОСТОЯТЕЛЬНАЯ РАБОТА

№ п/п	Раздел дисциплины	Форма (вид) самостоятельной работы	Трудоемкость в часах
1	Общая характеристика систем автоматизированного проектирования.	Выполнение лабораторной работы, оформление отчета.	6
2	Геометрическое моделирование.	Выполнение двух лабораторных работ, оформление отчетов.	12
3	Вариационное моделирование.	Выполнение двух лабораторных работ, оформление отчетов.	12
4	Методы поиска и оптимизации решения.	Выполнение двух лабораторных работ, оформление отчетов.	12
5	Инженерный анализ методом конечных элементов.	Выполнение двух лабораторных работ, оформление отчетов, подготовка к сдаче зачета	26

7 МАТРИЦА КОМПЕТЕНЦИЙ

№ п/ п	Раздел дисциплины	Общее кол-во компетенций		
		ПК-1	ПК-5	
1	Вариационное моделирование.	+		1
2	Методы поиска и оптимизации решения.		+	1
3	Инженерный анализ методом конечных элементов.	+	+	2

8 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Образовательный процесс по дисциплине строится на основе комбинации следующих образовательных технологий.

Интегральную модель образовательного процесса по дисциплине формируют технологии методологического уровня: модульно-рейтинговое обучение, технология поэтапного формирования умственных действий, технология развивающего обучения, элементы технологии развития критического мышления.

Реализация данной модели предполагает использование следующих технологий стратегического уровня (задающих организационные формы взаимодействия субъектов образовательного процесса), осуществляемых с использованием определенных тактических процедур:

- лекционные (вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция);
- лабораторные (углубление знаний, полученных на теоретических занятиях, решение задач);
- тренинговые (формирование определенных умений и навыков, формирование алгоритмического мышления);
- активизации познавательной деятельности (приемы технологии развития критического мышления через чтение и письмо, работа с литературой, подготовка презентаций по темам домашних работ);
- самоуправления (самостоятельная работа студентов, самостоятельное изучение материала).

Рекомендуется использование информационных технологий при организации коммуникации со студентами для представления информации, выдачи рекомендаций и консультирования по оперативным вопросам (электронная почта), при проведении лекционных и практических занятий использование мультимедиа-средств.

9 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

9.1 Оценочные средства для текущего контроля успеваемости

- 9.1.1 Контрольные вопросы допуска к выполнению лабораторных работ
- 9.1.2 Отчеты о выполнении индивидуальных вариантов заданий лабораторных работ
- 9.2 Оценочные средства для промежуточной аттестации
- Вопросы к зачету:
- 9.2.1. Опишите программное обеспечение, относящееся к классу САПР. Какова его доля на современном рынке ПО.
- 9.2.2. Укажите ключевые этапы в истории развития систем автоматизированного проектирования.
- 9.2.3. Опишите концепцию параметрического проектирования на основе конструктивных элементов.
- 9.2.4. Для чего предназначены системы автоматизации производства.
- 9.2.5. Какие задачи решают системы управления данными об изделии.
- 9.2.6. Опишите разницу между автоматизацией черчения и геометрическим моделированием.
- 9.2.7. Назовите и опишите виды геометрического моделирования.
- 9.2.8. Каковы основные функции твердотельного моделирования.
- 9.2.9. Опишите три вида декомпозиционных моделей.
- 9.2.10. В чем разница между геометрией и топологией граничной модели.
- 9.2.11. Какова базовая функциональность пакетов геометрического моделирования.
- 9.2.12. Назовите основные классы трансформаций в трехмерном аффинном пространстве. Какими геометрическими параметрами они характеризуются.
- 9.2.13. Приведите алгоритмы вычисления матричного представления трехмерной трансформации по ее геометрическим параметрам.
- 9.2.14. Как задаются однородные B-сплайновые кривые и поверхности.
- 9.2.15. Опишите области приложения вариационного моделирования.
- 9.2.16. Что такое вариационный геометрический решатель.
- 9.2.17. Что такое декомпозиция задачи удовлетворения геометрическим ограничениям. Для чего она применяется.
- 9.2.18. Дайте общее определение задач условной и безусловной оптимизации.
- 9.2.19. Что такое параметрическая оптимизация в системах автоматизированного проектирования.
- 9.2.20. Что такое конечно-элементный анализ. На каком математическом аппарате он основан. Каковы области его применения.
- 9.2.21. Что такое обобщенная матрица жесткости. Опишите применение метода конечных элементов для различных классов физических задач.
- 9.2.22. Охарактеризуйте преимущества внедрения PLM на предприятии.
- 9.3 Учебно-методическое обеспечение самостоятельной работы
- 9.3.1 Карточки с заданиями и методическими указаниями по выполнению лабораторных работ
- 9.3.2 СТО СМК 4.2.3.05-2011. Стандарт ФГБОУВПО «АмГУ». Оформление выпускных квалификационных и курсовых работ (проектов), 2011. – 95 с.

10 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

- 10.1 Кондаков А.И. САПР технологических процессов: учеб.: рек. Мин. обр. РФ/ А. И. Кондаков. -М.: Академия, 2007.-269 с.
- 10.2 Гольдберг О.Д. Инженерное проектирование и САПР электрических машин : учеб.: рек. УМО/ О. Д. Гольдберг, И. С. Свириденко; под ред. О. Д. Гольдберга. -М.: Академия, 2008.-560 с.

10.3 Бушманов А.В. Математическое и компьютерное моделирование фиксирующих устройств в травматологии: моногр. /А.В.Бушманов; АмГУ, ФМиИ, -Благовещенск, изд-во Амур.гос.ун-та, 2007, -151 с.

б) дополнительная литература:

10.4 Автоматизированное проектирование узлов и блоков РЭС средствами современных САПР: Учеб. пособие для вузов: рек. Мин. обр. РФ/ Мироненко И.Г., Суходольский В.Ю., Холуянов К.К.. -М.: Высш. шк., 2002.-392 с.

10.5 Чуприн А.И. AutoCAD 2006 : лекции и упражнения/ А. И. Чуприн, В. А. Чуприн. - СПб.: Питер, 2006.-1200 с.

10.6 Рыжиков Ю.И. Имитационное моделирование: Теория и технологии/ Ю.Рыжиков, 2004, -381 с.

10.7 Лоу А.М. Имитационное моделирование: Учеб.пособ./А.М.Лоу, В.Д.Кельтон, 2004. -847 с.

10.8. Квартани Т. Визуальное моделирование с помощью Rational Rose 2002 и UML: Пер. с англ./Т.Квартани, 2003, -187 с.

10.9 Кривошеин И.А. Методы и средства системной разработки сложных объектов на основе имитационного сетевого моделирования и технологии МетаСАПР (Framework)/И.А.Кривошеин, и др., 2005. -32 с.

в) периодические издания:

10.10 САПР и графика

10.11 Российский журнал биомеханики

10.12 Мехатроника, автоматизация, управление

10.13 Вычислительные технологии

г) программное обеспечение и Интернет-ресурсы:

Свободно распространяемое программное обеспечение

№ п/п	Наименование ресурса	Характеристика
1	see http://www.iqlib.ru	Интернет библиотека образовательных изданий, в которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам, отдельным темам и отраслям знаний.
2	http://www.intuit.ru	Интернет-университет информационных технологий, в котором вобраны электронные и видео-курсы по отраслям знаний
3	http://amursu.ru	Сайт АмГУ, Библиотека – электронная библиотека АмГУ
4	http://www.biblioclub.ru	Электронная библиотечная система «Университетская библиотека – online»: специализируется на учебных материалах для ВУЗов по научно-гуманитарной тематике, а так же содержит материалы по точным и естественным наукам

11 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

11.1 Лекционная аудитория, оборудованная мультимедийными средствами

11.2 Лаборатории, оборудованные рабочими местами пользователей ЭВМ

12 РЕЙТИНГОВАЯ ОЦЕНКА ЗНАНИЙ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

Семестровый модуль дисциплины						
№ п/ п	Раздел дисциплины	Виды контроля	выполнения (недели)	Максимальное кол-во баллов	посещение, активность на занятиях	Максимальное кол-во баллов за модуль
1	Общая характеристика систем автоматизированного проектирования.	ЛР № 1	1-2	5	1	6
2	Геометрическое моделирование.	ЛР № 2	3-4	5	1	12
		ЛР № 3	5-6	5	1	
3	Вариационное моделирование.	ЛР № 4	7-8	5	1	12
		ЛР № 5	9-10	5	1	
4	Методы поиска и оптимизации решения.	ЛР № 6	11-12	5	1	12
		ЛР № 7	13-14	5	1	
5	Инженерный анализ методом конечных элементов.	ЛР № 8	15-16	5	1	12
		ЛР № 9	17-18	5	1	
6	Промежуточная аттестация	зачет	18	6	0	6
Итого						60

2 Краткое изложение программного материала

Раздел 1. Общая характеристика систем автоматизированного проектирования.

САПР является человеко-машинным комплексом взаимосвязанных программных и технических средств, которые объединены в единый технологический процесс проектирования, которое начинается из выбора и объяснение элементарной базы проектированного изделия и заканчивается созданием технической документации на его изготовление.

Использование единой информационной модели объекту на всех этапах процесса проектирования составляет главный принцип построения САПР. При этом отличительной особенностью технологии проектирования с помощью САПР есть активное применение процедур математического моделирования, которое составляет суть второго принципа – принципа

принятия проектных решений на основе математического эксперимента с моделью проектируемого объекту.

Из предназначенных для пользователя позиций САПР есть сложной аппаратно-программной системой, способной к расширению за счет присоединения к нее новых методов и процедур решения задач проектирования в конкретной проблемной области, а также путем агрегатирования новых проблемных областей.

Из позиции системного подхода проблемы автоматизации проектирования можно условно разделить на функциональные и инструментальные. К функциональным относятся проблемы поиска алгоритмов решения отдельных задач проектирования и логической увязки этих задач в единый технологический цикл проектирования, то есть методического (математического и лингвистического) обеспечение САПР.

Инструментальные – проблемы создания информационного, программного и технического обеспечения САПР. Автоматизация проектирования сложных систем основана на комплексном подходе к исследованию, разработки и внедрения аппаратных и программных средств, предназначенных для решения двух главных проблем: бездефектного проектирования изделий и сокращения сроков их разработки.

Теория САПР включает следующие основные разделы:

- формализация объекту, целей и критериев проектирования и системы, которое проектирует;
- алгоритмизация задач проектирования;
- технология реализации САПР.

САПР, и особенно интегрированные САПР, являются сложными объектами. Их создание требует больших материальных и высоко квалифицированных людских ресурсов. Поэтому важно при создании САПР выдерживать следующие принципы: включение, системного единства, развития, комплектности, информационного единства, совместимости, инвариантности.

Принцип *включения* предусматривает согласование параметров и возможностей конкретной САПР с более сложной САПР.

Принцип *системного единства* обеспечивается взаимными связями всех подсистем САПР.

Принцип *развития* предусматривает наращивание и усовершенствование компонентов САПР и связей между ними.

Принцип *комплектности* обеспечивает связанность проектирования отдельных элементов и всего объекту в целом на всех стадиях проектирования.

Принцип *информационного единства* требует использования в подсистемах САПР установленных соответствующими нормативными документами языков программ, образов представления информации, сроков, символов.

Принцип *совместимости* обеспечивает совместное функционирование всех подсистем САПР.

Принцип *инвариантности* требует, чтобы все подсистемы и компоненты САПР были по возможности универсальными или типичными, то есть инвариантными к проектируемым объектам.

Основным методом проектирования является *блоково-иерархический* метод, то есть метод декомпозиции сложного объекту на подсистемы (блоки, узлы, детали).

САПР должна удовлетворять основным требованиям:

1. Иметь универсальную структуру, разработанную по блоково-иерархическому принципу;
2. Иметь инвариантную часть, в которую можно включить средства организации диалога и введение-вывода графической информации, систему управления базами данных, библиотеки стандартных программ;
3. Создаваться как человеко-машинная система, которая развивается. Должна иметь средства генерации трансляторов из новых языков, описания классов объектов проектирования и про-

- цесса проектирования, средства автоматизированного соединения новых прикладных программ, а также модификации тех, что существуют и описание новых баз данных;
4. Обеспечивать весь цикл проектирования сложного устройства;
 5. Предусматривать адаптацию к изменениям технологических процессов производства изделий;
 6. Владеть универсальностью, которая снимает ручную подготовку данных при переходе от одного этапа проектирования к другому;
 7. Осуществлять проектирование в реальном масштабе времени (режим "ОН-ЛАЙН");
 8. Базироваться на некотором типичном комплекте технических средств проектирования.

Ключевые вопросы:

1. Исторический обзор развития систем автоматизации проектирования;
2. Современные системы геометрического моделирования и их классификация;
3. Системы инженерного анализа;
4. Системы технологической подготовки производства;
5. Системы автоматизации производства;
6. Системы управления данными об изделии.

Литература:

- 1 Кондаков А.И. САПР технологических процессов: учеб.: рек. Мин. обр. РФ/ А. И. Кондаков. -М.: Академия, 2007.-269 с.
- 2 Гольдберг О.Д. Инженерное проектирование и САПР электрических машин : учеб.: рек. УМО/ О. Д. Гольдберг, И. С. Свириденко; под ред. О. Д. Гольдберга. -М.: Академия, 2008.-560 с.
- 3 Бушманов А.В. Математическое и компьютерное моделирование фиксирующих устройств в травматологии: моногр. /А.В.Бушманов; АмГУ, ФМиИ, -Благовещенск, изд-во Амур.гос.ун-та, 2007, -151 с.

5.1.2 Раздел 2. Геометрическое моделирование.

В соответствии с ГОСТ 22487-77, проектирование технического объекта – это процесс составления описания, необходимого для создания еще несуществующего объекта, который осуществляется преобразованием первичного описания (технического задания), оптимизацией характеристик объекта и алгоритма его функционирования, устранением некорректности первичного описания и последовательным представлением описаний детализируемого объекта для различных этапов проектирования.

При работе с двухмерными графическими пакетами (типа КОМПАС или AutoCAD) промежуточные и окончательные описания объекта проектирования строятся на основе его проекций на координатные плоскости.

Ограниченность такого подхода вынуждала широко применять физическое макетирование при проектировании сложных изделий, которое позволяло разработчикам оперировать физическими моделями в трехмерном пространстве.

Системы трехмерного 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. 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.

При создании трехмерной модели твердого тела конструктор может использовать различные инструменты (функции), входящие в арсенал того или иного программного пакета.

Количество подобных инструментов может достигать от нескольких десятков (для простых систем) до нескольких сотен (для сложных систем). Однако все это множество функций может быть разбито на несколько основных групп.

К первой группе относятся функции создания объемных тел путем перемещения плоской фигуры.

Ко второй группе относятся функции моделирования, позволяющие оперировать с уже существующими геометрическими объектами – гранями, ребрами, вершинами. Примером таких функций могут служить функции создания скругления граней, придания граням уклона или кривизны и т.д.

К третьей группе можно отнести оставшиеся многочисленные функции, в том числе и реализующие принципы объектно-ориентированного моделирования (например, создание отверстий различного типа).

Параметрическое моделирование (parametric modelling) заключается в том, что конструктор определяет форму заданием геометрических взаимосвязей и некоторых размерных параметров. Геометрические взаимосвязи описывают отношения геометрических элементов. Примерами взаимосвязей являются: параллельность двух граней, компланарность двух ребер, касательность криволинейного ребра к соседнему прямому и т. д. К размерным данным относятся не только заданные размеры формы, но и соотношения между размерами. Соотношения записываются конструктором, функция замещения с вычитанием объема в виде математических уравнений. Таким образом, параметрическое моделирование заключается в построении формы путем решения уравнений, выражающих геометрические взаимосвязи, и уравнений, описывающих заданные размеры и соотношения между ними.

Ключевые вопросы:

1. Автоматизация чертежных работ и геометрическое моделирование функции твердотельного моделирования;
2. Декомпозиционные и конструктивные модели;
3. Декомпозиционные и конструктивные модели;
4. Корректность граничных моделей;
5. Пакеты геометрического моделирования.

Литература:

- 1 Кондаков А.И. САПР технологических процессов: учеб.: рек. Мин. обр. РФ/ А. И. Кондаков. -М.: Академия, 2007.-269 с.
- 2 Гольдберг О.Д. Инженерное проектирование и САПР электрических машин : учеб.: рек. УМО/ О. Д. Гольдберг, И. С. Свириденко; под ред. О. Д. Гольдберга. -М.: Академия, 2008.-560 с.
- 3 Бушманов А.В. Математическое и компьютерное моделирование фиксирующих устройств в травматологии: моногр. /А.В.Бушманов; АмГУ, ФМиИ, -Благовещенск, изд-во Амур.гос.ун-та, 2007, -151 с.

5.1.3 Раздел 3. Вариационное моделирование.

Значительный вес в автоматизированном проектировании изделий имеет разработка геометрических моделей (ГМ). Геометрические модели используются на протяжении практически всего жизненного цикла изделия. Одной из наиболее мощных технологий разработки ГМ в машиностроительных САПР является параметрическое моделирование, подразумевающее использование параметрических геометрических моделей (ПГМ), управляемых параметрами. Особенно продуктивно вариационное параметрическое моделирование, в котором модель описывается сугубо декларативно, а геометрический решатель САПР сам находит путь построения экземпляра модели.

Применение ПГМ позволяет значительно снизить трудоемкость разработки ГМ благодаря автоматической генерации различных вариантов конструкции по заданным параметрам; широкому повторному использованию разработанных моделей и организации библиотек типовых параметрических конструктивных элементов; автоматическому решению большого

числа геометрических задач, возникающих в процессе проектирования. Наибольший эффект использование ПГМ дает в типовом конструировании, эскизном черчении, моделировании кинематики механизмов.

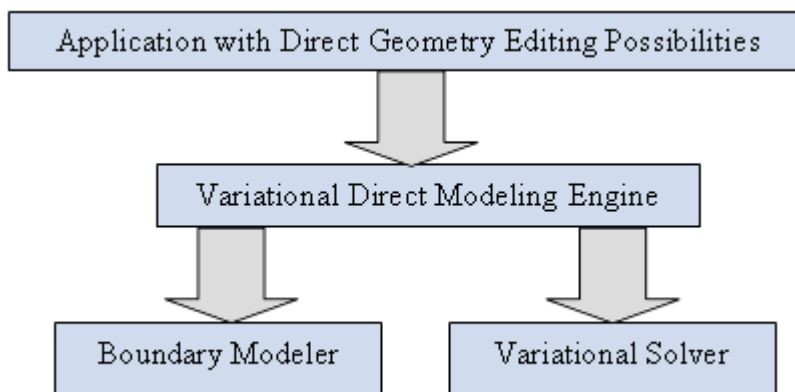
Появление и развитие теории параметризации в начертательной геометрии и технических чертежах связано с трудами отечественных ученых: Четверухина Н.Ф., Котова И.И., Рыжова Н.Н., Полозова В.С., Роткова С.И. и их учеников. Разработке и совершенствованию методов расчета вариационных ПГМ в САПР посвящены работы Голованова Н.Н., Ершова А.Г., Копорушкина П.А., Кучуганова В.Н., Лячека Ю.Т., Чижова А.В. и др. Из западных ученых значительный вклад в развитие методов расчета ПГМ внесли I.E.Sutherland, С.М.Hoffmann, J.Owen, В.N.Freeman-Benson, A.Borning, В.А.Myers и др.

Однако использование даже вариационных ПГМ требует от пользователя ряда дополнительных усилий. Кроме самой геометрии, от него требуется явно задавать геометрические, размерные и другие отношения, которые должны выполняться при расчете экземпляра модели. В сложных геометрических моделях это ведет к значительному повышению трудоемкости подготовки ПГМ. В таких случаях бывает эффективнее разрабатывать ГМ без использования параметризации. Не случайно во всех современных САПР параметризация является необязательной. Даже компания Parametric Technology Company (PTC), первоначально сделавшая ставку на полную параметризацию ГМ, в последних версиях САПР Pro/Engineer отказалась от обязательной параметризации эскизов.

Кроме того, существующие решатели вариационных ПГМ в ряде случаев проявляют свои недостатки: они либо неустойчивы, либо не всегда выдают нужное пользователю решение, либо медлительны, либо способны решать очень ограниченный круг задач и не поддерживают алгебраических отношений, либо поддерживают только функциональные, не зацикленные отношения. В основном это определяется ограничениями используемых методик расчета вариационных ПГМ.

Современное развитие методов искусственного интеллекта и, в частности, баз знаний позволяет продолжить совершенствование технологии вариационного геометрического моделирования в САПР.

Вышеприведенная таблица указывает источники информации, которая содержится в модели (с историей построения или с «немой» геометрией), и которую можно облечь в форму геометрических ограничений и размеров. Комбинация геометрического BRep-моделера с вариационным геометрическим решателем позволяет реализовать этот подход в рамках любой системы прямого моделирования. Компания ЛЕДАС, следуя анонсированному курсу на построение готовых к интеграции наборов инструментальных средств, служащих основой для конечно-пользовательских приложений с вариационной функциональностью, планирует в ближайшее время выпустить набор для прямого моделирования геометрии на основе своего вариационного геометрического решателя LGS 3D. Открытая архитектура этого инструментального набора (рис. 4) позволяет интегрировать его в любую систему геометрического моделирования на основе граничного представления (будь то полигональные сетки или BRep).



Архитектура вариационного прямого моделера

Чтобы показать выразительные возможности данного инструментального набора, компания ЛЕДАС разработала add-on приложение для одной из самых популярных систем прямого моделирования Google SketchUp.

Ключевые вопросы:

1. Параметры, ограничения и вариационные модели;
2. Вариационный геометрический решатель;
3. Способы алгебраического моделирования геометрической задачи;
4. Метрический тензор геометрической задачи.

Литература:

- 1 Кондаков А.И. САПР технологических процессов: учеб.: рек. Мин. обр. РФ/ А. И. Кондаков. -М.: Академия, 2007.-269 с.
- 2 Гольдберг О.Д. Инженерное проектирование и САПР электрических машин : учеб.: рек. УМО/ О. Д. Гольдберг, И. С. Свириденко; под ред. О. Д. Гольдберга. -М.: Академия, 2008.-560 с.
- 3 Бушманов А.В. Математическое и компьютерное моделирование фиксирующих устройств в травматологии: моногр. /А.В.Бушманов; АмГУ, ФМиИ, -Благовещенск, изд-во Амур.гос.ун-та, 2007, -151 с.

5.1.4 Раздел 4. Методы поиска и оптимизации решения.

При проведении оптимизационных расчётов можно использовать ряд различных методов в зависимости от вида модели, её свойств и структуры. Непосредственная оптимизация с помощью подходящего метода НЛП применима во всех случаях, однако для некоторых задач полезно воспользоваться другими приёмами, как, например, методом последовательной оптимизации, когда решается ряд подзадач, или двухэтапным методом, в котором используются промежуточные приближенные модели. В тех случаях, когда предполагается существование множества локальных минимальных решений, следует использовать такой метод, который приводит к глобальному минимуму.

С помощью аналитических моделей, а также моделей поверхности отклика решения получаются либо непосредственно, либо методом последовательной минимизации. При непосредственной оптимизации выясняют, подходит ли структура задачи для специальных оптимизационных методов, или же следует пользоваться общими алгоритмами НЛП. Специальные методы предпочтительнее, особенно если задачу приходится решать много раз. Если же задача решается только один раз, применение общего метода НЛП может оказаться предпочтительнее с точки зрения общей экономии рабочего времени.

Метод последовательной оптимизации заключается в том, что решение задачи получается в результате решения последовательных подзадач с ограничениями. Основная идея метода состоит в том, чтобы найти решение сложной задачи путём разделения переменных на две группы. В одну группу объединяются переменные, значения которых трудно определить, а в другую - переменные, значения которых сравнительно легко вычислить. Обе подзадачи решаются раздельно, при этом проводятся координирующие вычисления для их связи.

Оптимизация имитационных моделей проводится непосредственно или с помощью различных двухэтапных методов. При непосредственной оптимизации имитационная модель используется как программа для расчётов выпуска продукции и вычисления значений ограничений. Если выполняется условие, что выходные параметры имитационной модели непрерывно дифференцируемы по входным параметрам, то применим любой градиентный алгоритм безусловной и условной оптимизации. В противном случае нужно использовать прямые методы, такие как метод комплексов или метод случайного поиска. При использовании прямых методов оптимизации в имитационных моделях часто встречаются три случая, которые могут затруднить проведение вычислений и привести к повторению итераций:

- 1) наличие неявных ограничений для зависимых (внутренних) переменных;
- 2) наличие подразумеваемых ограничений, которые приняты при построении модели;
- 3) наличие вычислительных процедур, которые используются при имитации.

Если в результате чего-либо из вышеперечисленного оптимизационная задача в окончательном виде оказывается слишком сложной для прямых методов оптимизации, применяют различные виды двухэтапных методов. При этом с помощью имитационной модели получают модель поверхности отклика в независимых переменных, для которой используется подходящий оптимизационный метод. Процесс решения повторяется, причём каждый раз используется поверхность отклика, модифицированная в соответствии с полученным предшествующим оптимизационным решением, до тех пор, пока разность между двумя последовательными решениями не станет достаточно малой. Двухэтапные методы отличаются прежде всего по виду используемых аппроксимирующих функций, по уровню детализации создаваемой модели поверхности отклика и по применяемым оптимизационным методам.

Для надёжной оптимизации моделей, которые могут иметь несколько локальных минимумов, следует воспользоваться несколькими методами решения задачи, чтобы найти глобальный минимум. Известные методы поиска глобального минимума делятся на детерминированные и стохастические, которые в свою очередь могут быть эвристическими или строго обоснованными. Простейший метод состоит в проведении ряда оптимизационных расчётов при различных начальных условиях. Иногда этот метод называется методом с несколькими начальными точками. В нём начальные точки выбираются из определённой решётки или же генерируются случайным образом. Оба этих метода эвристические и не дают полной уверенности в результате. Теоретически обоснованные методы глобальной оптимизации разработаны только для задач со специальной структурой.

Ключевые вопросы:

1. Задачи удовлетворения ограничениям и оптимизации в общей постановке;
2. Классификация методов поиска и оптимизации решения.

Литература:

- 1 Кондаков А.И. САПР технологических процессов: учеб.: рек. Мин. обр. РФ/ А. И. Кондаков. -М.: Академия, 2007.-269 с.
- 2 Гольдберг О.Д. Инженерное проектирование и САПР электрических машин : учеб.: рек. УМО/ О. Д. Гольдберг, И. С. Свириденко; под ред. О. Д. Гольдберга. -М.: Академия, 2008.-560 с.
- 3 Бушманов А.В. Математическое и компьютерное моделирование фиксирующих устройств в травматологии: моногр. /А.В.Бушманов; АмГУ, ФМиИ, -Благовещенск, изд-во Амур.гос.ун-та, 2007, -151 с.

5.1.5 Раздел 5. Инженерный анализ методом конечных элементов.

При описании метода конечных элементов авторы публикаций, к сожалению, совершенно не уделяют внимания анализу тех теоретических положений, которые были приняты исходными для метода и на которых базируется алгоритм его осуществления. Но возможности метода, которые могут быть получены в результате его применения, определяются именно закладываемыми в его основу теоретическими положениями и тем, каким образом они реализуются. Поэтому анализ теоретических положений метода в первую очередь представляет интерес при исследовании достоверности результатов его применения.

Первоначально метод конечных элементов использовался для решения задач по определению напряженно-деформированного состояния конструкций, и до сих пор это направление его применения остается преобладающим. Поэтому анализ будет основываться на существующих теоретических положениях данной области инженерных знаний. Но это совершенно не означает, что полученные результаты будут касаться лишь напряженно-деформированного состояния конструкции. Анализ имеет общий характер, и результаты его могут быть отнесены и к другим областям инженерной деятельности.

В основе метода конечных элементов лежит вариационный интегральный принцип Лагранжа. Суть принципа Лагранжа заключается в утверждении, что в системах, стесненных идеальными стационарными внутренними связями и находящимися под действием потенциальных сил, не зависящих явно от времени, из множества кинематически допустимых пере-

мещений, соответствующих заданным условиям, те, которые удовлетворяют условиям равновесия, придают потенциальной энергии системы стационарное значение. Если начальное и конечное положение системы достаточно близки, то действие по Лагранжу имеет минимум для действительного движения. В связи с этим принцип Лагранжа называется также принципом наименьшего действия в форме Лагранжа, то есть одним из фундаментальных принципов механики. Согласно принципу Лагранжа, в состоянии устойчивого равновесия значение потенциальной энергии системы минимально. Из вариационного принципа Лагранжа следует, что в состоянии равновесия системы каждый из ее элементов также находится в равновесии, которое соответствует равновесному состоянию всей системы. Принцип Лагранжа устанавливает также пропорциональную связь между силами и перемещениями элементов системы. Как видно, из достаточно специализированного описания сущности принципа Лагранжа следуют простые, практически используемые положения.

Выяснив основные физические исходные положения, на которых базируется метод, перейдем к рассмотрению основных этапов его реализации.

Рассчитываемая система расчленяется на некоторое число отдельных элементов конечных размеров, неразрывно связанных между собой в узловых точках, для которых должен быть известен характер соотношения между перемещениями и реакциями в узлах. Таким образом, генерируется сетка из конечных элементов простой геометрической формы, которая с достаточной степенью точности аппроксимирует конкретную конструкцию. В узловых точках элементов сетки на основании принципа Лагранжа связи между действующими силами и перемещениями записываются выражением вида:

$$\{F\}_r = [K] * \{q\}_r \quad (1)$$

где $\{F\}_r$ — матрица реакций в узлах, то есть вектор узловых сил конечного элемента;

$\{q\}_r$ — матрица перемещений в узлах, то есть вектор узловых перемещений;

$[K]$ — матрица жесткости конечного элемента.

Разрешающие уравнения (1) являются, по сути, уравнениями равновесия конечных элементов и включают в себя систему неоднородных линейных алгебраических уравнений относительно вектора узловых перемещений. Матрица жесткости представляет собой математическую запись физической связи между реакциями в узлах элемента и узловыми перемещениями. Матрица жесткости является интегральной характеристикой, включающей как физические свойства материала рассчитываемой системы, так и геометрические свойства конечного элемента и сгенерированной сетки. Именно правильное построение матрицы жесткости определяет на первом этапе степень точности разрешающего уравнения, степень соответствия узловых сил и узловых перемещений состоянию системы. В общем случае матрица жесткости уникальна для каждого элемента и имеет важнейшее значение при реализации метода конечных элементов. Сетка конечных элементов может считаться построенной лишь после того, как построена матрица жесткости. Точные значения коэффициентов жесткости одномерных конечных элементов стержневых систем с линейными характеристиками (для которых вначале и использовался метод конечных элементов) приводят к достаточно точным решениям задач о напряженно-деформированном состоянии такой системы. Но инженерные конструкции редко состоят из таких простых элементов, как стержень с одной степенью свободы в узле. Расширение использования метода для решения двумерных и трехмерных задач увеличивает количество степеней свободы конечного элемента. Это в значительной степени усложняет построение матрицы жесткости. В этих случаях возникает необходимость применения специальных математических методов, обычно оперирующих множеством матриц, которые в результате дают возможность получить лишь приближенное решение. В принципе, при любой матрице жесткости будет получено соответствующее ей решение, при котором в узловых точках элементов сохранится состояние равновесия. Вопрос состоит лишь в том, насколько точно данная матрица описывает физические свойства рассчитываемой системы и насколько полученные результаты ей соответствуют. Проверить это можно лишь после выполнения расчета с помощью анализа полученного в результате расчета напряженного состояния всей системы.

При правильном построении матрицы жесткости решение разрешающих уравнений с заданной степенью невязки узловых сил или узловых перемещений также представляет собой непростую математическую задачу. Для решения таких задач применяются специальные математические методы, использование которых неминуемо вводит свойственные им допущения и ограничения. Это приводит к сужению области применения данной аналитической программы, которое должно быть оговорено в документации. Кроме того, возникает опасность получения формально (математически) правильного, но не соответствующего физике задачи результата. Однако лишь применение специальных математических методов позволяют с определенной степенью точности за конечное число итераций определить узловые силы и узловые перемещения.

Попытаемся оценить объем вычислений, необходимый для выполнения лишь одной итерации. Размерность матрицы жесткости определяется числом степеней свободы конечного элемента. Для выполнения одной итерации необходимо произвести определенное количество простых математических действий. Оценим это число на примере простейшей задачи линейного математического программирования — выбор по такому одному элементу в каждой из n строк и в каждом из n столбцов квадратной таблицы чисел, который соответствовал бы заданной целевой функции. При этом положим, что вычисление целевой функции требует выполнения лишь одной простой математической операции с каждым элементом таблицы. Область определения целевой функции в задачах линейного программирования представляет собой выпуклое многомерное множество в n -мерном пространстве переменных. Число вершин соответствующего многогранного множества равно $n!$ При мерности таблицы чисел равной 20 число вершин равно $20! = 2,43 \times 10^{18}$. Компьютеру, выполняющему 100 млн. операций в секунду, потребуется более 770 лет, чтобы перебрать все вершины многогранника, определяемого условиями задачи. Совершенно очевидно, что даже такая простая задача оказывается практически невыполнимой без использования не только вычислительной техники, но и специальных методов, позволяющих значительно уменьшить объем и, следовательно, время выполнения вычислений.

Однако основная задача определения напряженно-деформированного состояния системы вводит новые условия. При каждой итерации узлы сетки элементов получают определенные перемещения — система деформируется. Изменяются и значения узловых сил. При деформировании упругой системы характер связей между элементами изменяется. Таким образом, в процессе решения задачи конечный элемент изменяет свое информационное состояние. Это ведет к необходимости соответствующей коррекции матрицы жесткости и переводит задачу в более сложную область — область динамического программирования, где также используются соответствующие допущения, ограничения и упрощения. Все это не только значительно усложняет решение, но и ведет к накоплению погрешности результата.

Вот почему сообщения о решении задач определения напряженно-деформированного состояния конструкции за время, исчисляемое минутами или даже секундами, вызывает истинное восхищение искусством специалистов, создавших конкретную программу. Но, в то же время такие известия вызывают определенные опасения в достоверности результатов расчета.

Однако допустим, что описанные сложности метода успешно преодолены, и для всех элементов сетки определены узловые силы и узловые перемещения, точно соответствующие условию равновесия каждого элемента. Можно ли считать задачу определения напряженно-деформированного состояния системы решенной? Вопрос этот связан с тем, как метод конечных элементов использует положение принципа Лагранжа о равновесии всей системы и ее элементов. Совершенно очевидно, что состояние равновесия системы является причиной, следствие которой — равновесие элементов системы. Менять местами причину и следствие недопустимо.

Любая система может находиться в бесконечном множестве состояний равновесия, при котором все ее элементы также находятся в состоянии равновесия. Каждому из состояний равновесия всей системы соответствует совершенно определенное распределение внутрен-

них напряжений и деформаций, отличное от других распределений, то есть совершенно определенные состояния равновесия элементов, отличающиеся от других возможных состояний равновесия. Найдя с помощью метода конечных элементов узловые силы и соответствующие им узловые перемещения, при которых элемент находится в равновесии, нельзя гарантировать, что найденное состояние равновесия соответствует именно тому распределению, которое устанавливается в конкретной системе в результате воздействия заданного внешнего нагружения. Иначе говоря, предполагаемое положение о равновесии всей системы в случае, если все ее элементы находятся в состоянии равновесия, является ошибочным. Состояние равновесия системы при заданных условиях единственно возможное, отличное от всех других состояний, и распределение внутренних напряжений соответствует только этим условиям. Состояния равновесия же отдельных элементов, определенные с помощью метода конечных элементов, могут соответствовать разным распределениям внутренних напряжений и в сумме не будут обеспечивать общего состояния равновесия системы. Таким образом, в итоге может быть получен результат, который не будет соответствовать исходной задаче, что чаще всего и случается при расчете сложных систем. Безусловно, исходя из положения о том, что в любом сечении уравновешенной системы действующие силы также должны быть уравновешены, можно выявить ошибочность расчетных результатов. Но для исключения при суммировании расчетных значений сил в сечении случайного результата, удовлетворяющего условию равновесия суммарных значений, но не соответствующего их распределению по сечению, таких проверок необходимо делать несколько для разных сечений, отличающихся друг от друга. Описанные обстоятельства значительно усложняют применение программных средств, использующих метод конечных элементов, для выполнения инженерного анализа.

Описанная особенность метода конечных элементов была проверена на большом количестве расчетов напряженного состояния оболочечных конструкций различными программами, использующими метод конечных элементов. Новая технология расчета оболочек позволяет с большой точностью получить распределение внутренних сил в зависимости от геометрической формы оболочки и схемы ее внешнего нагружения. Сопоставление аналитических результатов с результатами расчета тех же конструкций с помощью программ инженерного анализа показали, что при автоматизированном расчете распределение внутренних сил не соответствует распределению, при котором сохраняется вся оболочка. При этом равновесие узловых сил каждого элемента автоматизированного расчета сохранялось.

Может возникнуть естественный вопрос о целесообразности использования метода конечных элементов в автоматизированных программных продуктах инженерного анализа. Ответ в этом случае будет однозначен: применение метода конечных элементов безусловно оправданно в соответствующих условиях. Но при этом в документацию программ в обязательном порядке должна быть введена информация, в которой совершенно конкретно и однозначно описана область и указан класс задач, для которых данная программа предназначена. Должны быть указаны и условия, при которых программа не обеспечивает получение достоверных результатов. Кроме того, разработчик программы обязан нести ответственность за достоверность результатов решения оговоренных в сопроводительной документации задач. Безусловно, что неквалифицированное использование аналитических программ не может служить причиной иска. Но ничем не ограничиваемое, а часто и провоцируемое рекламой, неоправданное расширение областей применения программных средств при одновременном значительном снижении требуемого уровня специальной подготовки пользователей, неминуемо ведет к совершенно очевидным последствиям.

Чего же можно ожидать в ближайшем будущем от автоматизированных средств инженерного анализа? По-видимому, существующий сейчас алгоритм применения метода конечных элементов в ближайшее время претерпит коренные изменения. Представляется, что вероятным направлением развития программных средств инженерного анализа станет интеграция теоретических решений в конкретных областях инженерных расчетов с возможностями численных математических методов. Достоинством теоретических решений является их точность. Для простых задач могут быть составлены простые и легко решаемые уравнения. Од-

нако в практике проектирования «простые» задачи практически не встречаются. С усложнением задачи усложняется и ее математическое описание, и теоретическое решение. В результате для большинства практических задач обычно получаем либо весьма сложные уравнения, либо вообще неразрешимые. Но практическая задача может решаться по алгоритму поэтапного уточнения. На первом этапе исходная задача упрощается таким образом, чтобы, сохраняя основные характеристики, стало возможным описать ее математически и решить аналитически. Затем в задачу вводятся не вошедшие в математическое описание элементы, влияние которых оценивается с помощью численных математических методов с использованием результатов аналитического решения, в частности с помощью метода конечных элементов.

Ключевые вопросы:

1. Матрицы жесткости и упругости;
2. Уравнения равновесия;
3. Общая схема конечно-элементного анализа.

Литература:

1. Гвоздева Т.В. Проектирование информационных систем: учеб. пособие: рек. УМО/ Т.В. Гвоздева, Б.А. Баллод. -Ростов н/Д: Феникс, 2009.-509 с.
2. Советов Б.Я. Информационные технологии: учеб.: доп. Мин. обр. РФ/ Б.Я. Советов, В.В. Цехановский. -4-е изд., стер.. -М.: Высш. шк., 2008.-264 с.
3. Емельянова Н.З. Основы построения автоматизированных информационных систем : учеб. пособие: рек. Мин. обр. РФ/ Н. З. Емельянова, Т. Л. Партыка, И. И. Попов. -М.: ФОРУМ: ИНФРА-М, 2007.-416 с.

2. Методические материалы к выполнению лабораторных работ

Лабораторная работа 1. Анализ систем автоматизации проектирования.

Программное обеспечение (ПО) САПР представляет собой совокупность всех программ и эксплуатационной документации к ним, необходимых для выполнения автоматизированного проектирования. Программное обеспечение делится на общесистемное и специальное (прикладное).

Общесистемное ПО предназначено для организации функционирования технических средств, т. е. для планирования и управления вычислительным процессом, распределения имеющихся ресурсов, и представлено операционными системами ЭВМ и вычислительных комплексов (ВК). Общесистемное ПО обычно создается для многих приложений и специфику САПР не отражает.

В специальном (прикладном) ПО реализуется математическое обеспечение для непосредственного выполнения проектных процедур. Прикладное ПО обычно имеет форму пакетов прикладных программ (ППП), каждый из которых обслуживает определенный этап процесса проектирования или группу однотипных задач внутри различных этапов.

Рассмотрим принципиальные особенности ПО, влияющие на организацию и эффективность создания и использования САПР. С развитием и совершенствованием ЭВМ все большее значение приобретает такой компонент общесистемного ПО, как операционные системы (ОС). Возможности, предоставляемые пользователям современными ОС, в большей степени определяются их операционными системами, чем техническими устройствами. ОС организует одновременное решение различных задач на ЭВМ, динамическое распределение каналов передачи данных и внешних устройств между задачами, планирование потоков задач и последовательность их решения с учетом установленных критериев, динамическое распределение памяти вычислительного комплекса. Однако ОС требует для своей работы определенных ресурсов: процессора, внешней и основной памяти. Чем большими возможностями обладает ОС, тем больше требуется для нее ресурсов.

Операционные системы можно генерировать применительно к определенным конфигурациям технических средств вычислительного комплекса и кругу решаемых задач. Но при этом параметры и состав технических средств ограничивают возможности ОС.

Важным компонентом общесистемного ПО является базовое ПО. Базовое ПО не является объектом разработки при создании программного обеспечения САПР. Примером может служить базовое ПО для обработки геометрической и графической информации, для формирования и использования баз данных (БД).

Использование АРМ, в состав которых включено подобное базовое ПО, реализующее стандартные проектные процедуры, существенно снизит трудоемкость создания программного обеспечения САПР. Однако во всех случаях за создателями САПР останется разработка прикладного ПО. С расширением области применения вычислительной техники и усложнением задач автоматизации процессов проектирования возрастают сложность и трудоемкость программирования.

В последнее время, особенно в связи с широким внедрением в инженерную практику персональных компьютеров, начинают использоваться функциональные и интегрированные пакеты программ.

Функциональные пакеты программ (ФПП) — это комплекс программных средств, ориентированных на выполнение определенной функции, более или менее безотносительно к конкретному предметному содержанию (обработка текстов — текстовые редакторы, обработка таблиц, графики).

Интегрированные пакеты программ (ИПП) — это сочетание разных пакетов программ в единой технологической системе.

Интеграция может быть реализована соединением основных функциональных пакетов в целостную монолитную систему, представленную единым программным модулем, или путем создания набора вспомогательных средств интерфейсного характера для обеспечения взаимодействия пакетов, представленных независимыми модулями.

Прикладное программное обеспечение должно удовлетворять следующим основным требованиям:

Правильность — это функционирование в соответствии с моделируемым объектом и полным совпадением с выбранным алгоритмом решения.

Точность — результаты расчета имеют допустимые отклонения от реальных.

Совместимость — возможность работы не только в автономном режиме, но и в составе интегрированных систем, САПР и др.

Надежность — при всех условиях обеспечивает повторяемость результата.

Универсальность — работа при любых допустимых исходных данных.

Защищенность — сохранение работоспособности при возникновении сбоев.

Полезность — практическая ценность решаемых задач.

Эффективность — требуемые ресурсы (память, время) невелики.

Проверяемость — возможность демонстрации качества на практике.

Адаптируемость — возможность быстрой модификации с целью приспособления к изменяющимся условиям функционирования.

Цели и задачи: Изучить функции и проектные процедуры наиболее распространенных САД, САМ и САЕ-систем.

После занятия студент должен:

1. Знать: Основные функции и проектные процедуры наиболее распространенных систем верхнего, среднего и нижнего уровней.
2. Уметь: Предложить и обосновать перечень программных средств для решения задачи проектирования и подготовки производства изделия средней сложности.

Лабораторная работа 2. Геометрическое моделирование с использованием САД-систем.

Работа состоит из трех частей.

Задача первой части — создать сборку по ранее разработанным 3D-моделям деталей. Проектируемая сборка должна удовлетворять ряду требований:

- 1) Первый компонент должен быть фиксирован. Несмотря на то, что фиксирование первого компонента в ANSYS не обязательно, на практике это может оказаться полезным.
- 2) Все компоненты должны быть полностью ограничены. Полное ограничение компонентов позволяет исключить возникновение нежелательных изменений.
- 3) Необходимо использовать инструменты дублирования. Повторное использование существующих компонентов значительно сокращает время разработки.

Во второй части практической работы необходимо спроектировать деталь в контексте сборки. Создаваемая деталь должна удовлетворять ряду требований:

- 1) Модель должна быть создана внутри сборки.
- 2) Эскиз базового примитива детали должен базироваться на плоской грани смежного компонента.

В третьей части необходимо провести анализ сборки. Анализ сборки должен содержать:

- 1) Анализ ограничений сборки.
- 2) Определение числа степеней свободы компонентов сборки.
- 3) Анализ ограничений компонентов при помощи инструмента разнесения сборки.

Лабораторная работа 3. Моделирование конструкций с использованием САД-систем.

Цель работы: Используя функции интерфейса прикладного программирования САПР SolidWorks создать прикладную программу, позволяющую осуществлять автоматизированную сборку конструкции ДЭМД тем самым закрепить навыки работы с интерфейсом прикладного программирования.

Задание:

1. Используя разработанный в предыдущей лабораторной работе интерфейс создать программу, взаимодействующую с САПР SolidWorks посредством API-функций.

2. Используя набор соответствующих API-функций обеспечить автоматизированное построение сборки конструкции ДЭМД.

3. Выполнить тестирование программы.

Порядок выполнения работы:

1. По нажатию кнопки "Создать модель" на форме при помощи функции `OpenDoc6` в среде SolidWorks последовательно открыть файлы деталей, необходимые для построения сборки ДЭМД.

2. В случае необходимости редактирования детали необходимо сделать соответствующий документ активным в среде *SolidWorks* при помощи функции `ActivateDoc2`.

```
part = swApp.ActivateDoc2("Индуктор.SLDPRT", False, longstatus)
```

Для того чтобы иметь возможность использовать функции классов *ActiveDoc* и *Extension* создать экземпляры этих классов.

```
swModel = swApp.ActiveDoc
```

```
swModelExt = swModel.Extension
```

```
boolstatus = swModel.SetUserPreferenceToggle(198, True)
```

3. Для создания нового файла типа «Сборка» в котором будет осуществляться построение сборки ДЭМД выполнить следующие функции:

```
part = swApp.NewDocument("C:\..\Сборка_ESKD.asmdot", 0, 0.0#, 0.0#)
```

```
part = swApp.ActivateDoc2("Сборка_ESKD.asmdot", False, longstatus).
```

В результате выполнения данных функций происходит создание нового файла сборки в среде *SolidWorks* и перевод его в активный режим. Для добавления деталей в сборку используется функция `AddComponent`. После того, как все необходимые детали добавлены в сборку, необходимо вызвать функции `CloseDoc` для закрытия каждого из файлов деталей.

4. После добавления всех необходимых деталей в сборку необходимо осуществить их корректное сопряжение между собой. Сопряжение выполняется при помощи функции `AddMate`. Ниже приведен пример программного кода, выполняющий сопряжение фронтальной плоскости индуктора ДЭМД с фронтальной плоскостью кронштейна крепления.

```
boolstatus = swModelExt.SelectByID2("Плоскость1@Индуктор-" &K& "@Сборка1",
```

```
"PLANE", 0, 0, 0, True, 1, Nothing, 0)
boolstatus = swModelExt.SelectByID2("Плоскость2@Кронштейн-" &K&
"@Сборка1", "PLANE", 0, 0, 0, True, 1, Nothing, 0)
Feature = swAssembly.AddMate2(0, 1, False, 0, 0, 0, 1, 1, 0, 0, 0, longstatus)
swModel.ClearSelection2(True)
```

5. После того, как все сопряжения в сборке установлены посредством вызова функции *Visible(True)* перевести приложение *SolidWorks* из фонового режима в активный для визуального контроля правильности сборки.

6. Подготовить отчет, в который занести сведения о ходе выполненной работы.

Лабораторная работа 4. Программирование на языке Auto LISP, построение моделей.

Цель работы

Построить заданный контур таким образом, чтобы меняя координаты точки привязки можно было перемещать контур по формату чертежа, а также задавать линейные и радиальные размеры контура.

Исходные данные

Контур показан на рисунке 1.

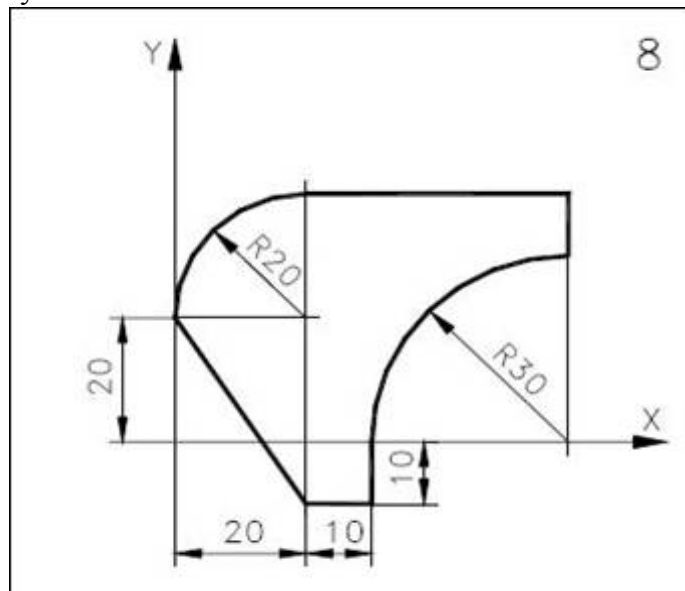


Рисунок 1 – Заданный криволинейный контур

Методика выполнения работы

Запускаем программу AutoCAD 2004, в меню Tools выбираем пункт «AutoLISP» и команду “Visual LISP Editor”. Открывается окно редактора AutoLISP и вводим описание контура.

Создание программы на языке AutoLISP.

Пишем программу для отрисовки контура с возможностью ввода параметров для точки привязки и размеров.

Рассмотрим отдельные части программы подробнее.

Для ввода угловых значений в градусах, создаем при помощи оператора “defun” новую функцию dtr, которая переводит градусы в радианы.

```
(defun dtr(a)
(* pi(/ a 180.))
)
```

Далее следует создание новой команды “kontur” и описание ее параметров.

```
(defun c:kontur()
(command "_limits" "" "210,297") – введение границ чертежа;
(command "_zoom" "w" "-10,-10" "90,90") – определение масштаба изображения на экране;
```

```
(command "_erase" "all" "") – команда очищает экран;
запрос точки привязки контура:
(princ "Укажите положение точки привязки:")
(setq pt1 (getpoint))
запрос реальных значений размеров контура:
(setq h1 (getreal "\Размер h1:"))
(setq h2 (getreal "\Размер h2:"))
(setq h3 (getreal "\Размер h3:"))
(setq h4 (getreal "\Размер h4:")) – лишний параметр! (см. замечание в конце работы).
(setq R1 (getreal "\Радиус R1:"))
(setq R2 (getreal "\Радиус R2:"))
```

Далее определяем узловые точки контура.

```
(setq pt2 (polar pt1 0. h1))
(setq pt3 (polar pt2 (dtr 90.) h2))
(setq pt4 (polar pt3 (dtr 45.) (/ R2 (cos (/ pi 4.)))))
(setq pt5 (polar pt4 (dtr 90.) (- (+ R1 h3) R2)))
(setq pt6 (polar pt5 (dtr 180.) (+ R2 h1)))
(setq pt7 (polar pt6 (dtr 225.) (/ R1 (cos (/ pi 4.)))))
(setq pt8 (polar pt7 0. R1))
(setq pt9 (polar pt3 0. R2))
(setq pt10 (polar pt9 (dtr 135.) R2))
(setq pt11 (polar pt8 (dtr 135.) R1))
```

Построение контура.

```
(command "_line" pt1 pt2 "")
(command "_line" pt2 pt3 "")
(command "_arc" pt3 pt10 pt4)
(command "_line" pt4 pt5 "")
(command "_line" pt5 pt6 "")
(command "_arc" pt6 "_e" pt7 "_r" R1)
(command "_line" pt7 pt1 "")
```

Простановка размеров.

```
(command "_dimlinear" pt1 pt2 "@ 0,-5")
(command "_dimlinear" pt2 pt3 "@ 5,0")
(command "_dimlinear" pt7 pt1 "@ 0,-5")
(command "_dimlinear" (polar pt1 (dtr 90.) h2) pt8 (polar pt7 (dtr 180.) 5.))
(command "_dimradius" pt11 pt11)
(command "_dimradius" pt10 pt10)
(command "_line" pt1 (polar pt6 (dtr 90.) 5.) "")
(command "_line" (polar pt1 (dtr 90.) h2) pt3 "")
```

Использование созданной программы.

После того, как программа готова сохраняем файл под именем Donguzov.lsp и в окне AutoCAD открываем меню Tools, пункт «AutoLISP» и запускаем команду “Load”. После чего нам предлагается выбрать файл с расширением lsp.

Выбираем необходимый файл и нажимаем кнопку “Load”, а затем “Close”. В командную строку AutoCAD вводим команду “kontur” и нажимаем ввод, после чего, если в программе нет ошибок, нам предлагается указать точку привязки контура.:

Command: Укажите положение точки привязки:

Это можно сделать путем ввода данных в командную строку, либо указать курсором «мыши» и нажать левую кнопку.

Далее предлагается ввести данные размеров контура:

Размер h1:10

Размер h2:10

Размер h3:20

Размер h4:20

Радиус R1:20

Радиус R2:30

Вводя разные значение размеров, получаем разные варианты исходного контура.

Полный текст программы:

```
(defun dtr(a)
(* pi(/ a 180.))
)
(defun c:kontur()
(command "_limits" "" "210,297")
(command "_zoom" "w" "-10,-10" "90,90")
(command "_erase" "all" "")
(princ "Укажите положение точки привязки:")
(setq pt1 (getpoint))
(setq h1 (getreal "\Размер h1:"))
(setq h2 (getreal "\Размер h2:"))
(setq h3 (getreal "\Размер h3:"))
(setq h4 (getreal "\Размер h4:"))
(setq R1 (getreal "\Радиус R1:"))
(setq R2 (getreal "\Радиус R2:"))
;-----Определение узловых точек-----
(setq pt2 (polar pt1 0. h1))
(setq pt3 (polar pt2 (dtr 90.) h2))
(setq pt4 (polar pt3 (dtr 45.) (/ R2 (cos (/ pi 4.)))))
(setq pt5 (polar pt4 (dtr 90.) (-(+ R1 h3) R2)))
(setq pt6 (polar pt5 (dtr 180.) (+ R2 h1)))
(setq pt7 (polar pt6 (dtr 225.) (/ R1 (cos (/ pi 4.)))))
(setq pt8 (polar pt7 0. R1))
(setq pt9 (polar pt3 0. R2))
(setq pt10 (polar pt9 (dtr 135.) R2))
(setq pt11 (polar pt8 (dtr 135.) R1))
;-----Отрисовка контура-----
(command "_line" pt1 pt2 "")
(command "_line" pt2 pt3 "")
(command "_arc" pt3 pt10 pt4)
(command "_line" pt4 pt5 "")
(command "_line" pt5 pt6 "")
(command "_arc" pt6 "_e" pt7 "_r" R1)
(command "_line" pt7 pt1 "")
;-----Простановка размеров-----
(command "_dimlinear" pt1 pt2 "@ 0,-5")
(command "_dimlinear" pt2 pt3 "@ 5,0")
(command "_dimlinear" pt7 pt1 "@ 0,-5")
(command "_dimlinear" (polar pt1 (dtr 90.) h2) pt8 (polar pt7 (dtr 180.) 5.))
(command "_dimradius" pt11 pt11)
(command "_dimradius" pt10 pt10)
(command "_line" pt1 (polar pt6 (dtr 90.) 5.) "")
(command "_line" (polar pt1 (dtr 90.) h2) pt3 "")
)
)
```

Обсуждение результатов

В процессе выполнения данной работы была создана программа параметрического описания контура на языке AutoLISP. Допущенные ошибки: при создании запроса реальных размеров ввел лишний параметр h_4 , который равен радиусу R_1 , и при введении значения h_4 , не совпадающего с R_1 , возникают ошибки в построении контура. Также столкнулся с рядом ошибок, которые были вызваны в основном невнимательностью (пропущены скобки или точки) и были легко устранены благодаря цветовой поддержке вводимых данных.

Вывод

В результате выполнения данной работы усвоил принципы написания программ на языке AutoLISP и закрепил на практике теоретические знания операторов данного языка. AutoLISP позволяет создавать различные параметрические модели, которые могут существенно автоматизировать работу конструктора. Работу считаю выполненной в полном объеме.

Лабораторная работа 5. Использование стандарта обмена геометрическими данными.

Рабочее проектирование изделия связано с созданием с помощью интегрированных систем автоматизированного проектирования и технологической подготовки производства (CAD/CAM/CAE) комплекта рабочей конструкторской документации, включающей согласованную и взаимоувязанную графическую и текстовую информацию об изделии. Это ограничивает возможности обмена информацией между участниками процесса проектирования изделия (и соответственно между их CAD/CAM/CAE системами) с помощью известных символьных форматов передачи графической (типа IGES, DXF, SAT) и текстовой информации (типа языков SGML, XML).

Использование различных автоматизированных систем (АС) независимых разработчиков в процессах жизненного цикла изделий порождает проблему информационной совместимости этих систем, ограничивающую возможность использования одних и тех же данных и обмена ими.

Кардинальным решением рассмотренных проблем является внедрение CALS-технологий. «CALS — это единая стратегия правительства и промышленности, концентрирующаяся на перестройке существующих бизнес-процессов в высокоавтоматизированный и интегрированный процесс управления жизненным циклом оборонных систем» [1] Основной инструмент CALS — это SDE (Shared data environment — среда совместно используемых данных). В русскоязычной литературе чаще используется термин «единое информационное пространство» (ЕИП).

Является ли CALS-технология (имеется в виду CALS, а не некоторые вольные интерпретации, известные как ИПИ) единственным возможным решением при построении ЕИП? Существует альтернатива CALS — т.н. PLM.

Эти понятия часто путают, а некоторые даже наивно полагают, что речь идет об одном и том же. В действительности же, несмотря на некоторое сходство, CALS и PLM — это два противоположных подхода к достижению единой цели, которая заключается в полном объединении всех задач, решаемых с помощью компьютера, на всех этапах жизненного цикла изделия: маркетинг, подготовка производства (проектирование, конструкторская и технологическая подготовка производства), материально-техническое снабжение, производство, контроль, упаковка и хранение, распределение, эксплуатация и утилизация (см. стандарты серии ISO 9000).

Подход PLM, суть которого отражена на рис. 1, состоит в том, чтобы обеспечить решение всех задач с помощью набора взаимоувязанных программных продуктов одного крупного разработчика программного обеспечения. На этом же рисунке просматривается и основная возникающая при этом проблема — зависимость пользователя от программных продуктов одного разработчика.

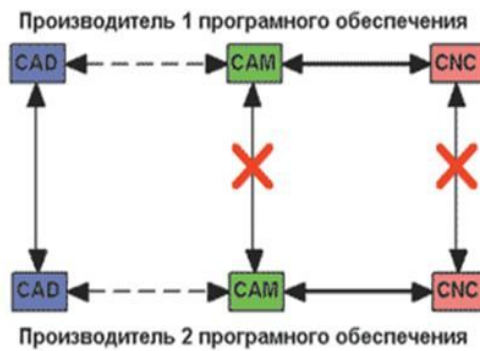


Рис. 1 PLM-решение

Подход CALS, наоборот, заключается в том, чтобы освободить пользователя от зависимости от одного разработчика. Основа подхода — это вышеупомянутое SDE, или единое информационное пространство, построенное на применении международных стандартов представления данных. Основным стандартом является ISO 10303 STEP (STandard for Exchange of Product model data — Стандарт обмена данными модели изделия). Статус международного стандарта обеспечивает два важных свойства STEP — стабильность (стандарт пересматривается примерно раз в пять лет, и новые версии не изменяют и не отменяют, а дополняют старые) и общедоступность (необходимые для практической работы материалы по стандарту либо находятся в свободном доступе в Интернете, либо могут быть куплены в официальных органах стандартизации, например во ВНИИКИ).



Рис. 2 CALS-технология на основе ISO 10303 STEP

В силу сложившихся обстоятельств на рассматриваемом предприятии используются два крупных программных комплекса, которые могут рассматриваться как PLM-решения, т.е. сложилась ситуация, изображенная на Рис. 1. Следовательно, построение ЕИП на предприятии возможно только на основе CALS-технологий.

Составление проекта внедрения CALS-технологий на предприятии представляет собой сложную техническую проблему, требующую применения специальных методик. Оснащение предприятия трансляторами AC – STEP и STEP – AC для каждой из применяемых разновидностей AC – условие необходимое, но недостаточное, т.к. в силу сложности структуры передаваемой информации, разных способов представления одних и тех же объектов (особенно геометрических и топологических) в различных AC конвертирование данных приводит к ошибкам, составляющим, как показывает отечественная и зарубежная практика, не менее 10 % от объема передаваемой информации

Лабораторная работа 6. Моделирование в системе ABAQUS (уч.версия).

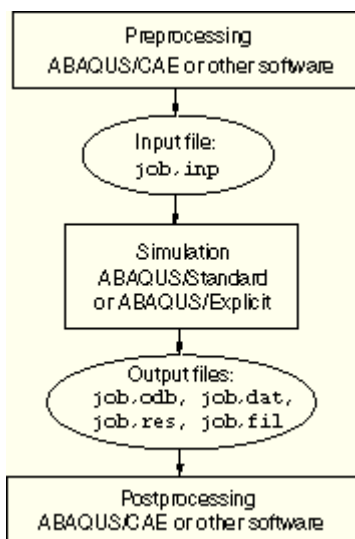
Программный комплекс ABAQUS разработан по модульному принципу. Он состоит из двух основных модулей – решателей (солверов) ABAQUS/Standard и ABAQUS/Explicit, пре-процессора ABAQUS/CAE и дополнительных модулей, учитывающих особенности специфических проблем (ABAQUS/Aqua, ABAQUS/Design, ABAQUS/Safe).

Построение конечно-элементной модели и операции с ней осуществляются в ABAQUS/CAE. ABAQUS/CAE также разделен на модули, каждый из которых содержит не-

который набор действий, близких по значению и необходимых для создания модели. В нем представлены следующие модули:

- **PART** – модуль, предназначенный для создания деталей, задания их геометрии, опорных точек и систем координат;
- **PROPERTY** – модуль, предназначенный для определения материалов и сечений, применяемых в модели, а также для задания их физических характеристик;
- **ASSEMBLY** – модуль, предназначенный для задания взаимного расположения деталей и сборки их в единую модель;
- **STEP** – модуль, предназначенный для создания шагов расчета и определения выходных данных
- **INTERACTION** – модуль, предназначенный для определения взаимодействий между деталями, контактных участков, их свойств;
- **LOAD** – модуль, предназначенный для построения сетки, фактически в нем происходит преобразование геометрической модели в конечно-элементную;
- **JOB** – модуль, предназначенный для создания файла выходных данных, проверки построенной модели, запуска вычислительного процесса и контроля над ним;
- **VASUALIZATION** – модуль, предназначенный для просмотра результатов расчета и обработки полученных данных;
- **SKETCH** – модуль, предназначенный для сохранения эскизов и чертежей полученной модели.

Полный процесс анализа в пакете ABAQUS/Standard обычно состоит из трех отдельных этапов: предварительная обработка, моделирование, постобработка. Эти три этапа соединяются вместе с помощью файлов, как показано ниже:



Предварительная обработка ABAQUS/CAE

Файл входных данных *.inp содержит определение модели физической задачи, как правило создается графически с использованием ABAQUS/CAE, либо другим препроцессором, так же входной файл ABAQUS для простых вычислений может быть создан прямо в текстовом редакторе.

Моделирование (ABAQUS/Standard)

Моделирование обычно происходит фонов процессом и является и является этапом, на котором ABAQUS/Standard решает численную задачу, определенную в файле входных дан-

ных. В процессе работы все производимы действия и промежуточные результаты отражаются в файле *.msg. По окончании работ результаты записываются в файл *.odb.

Постобработка (ABAQUS/Viewer)

Просмотр результатов происходит в интерактивном режиме с использованием ABAQUS/Viewer или другого постпроцессора. ABAQUS/Viewer считывает результаты из нейтрального файла базы данных двоичного вывода *.odb.

Лабораторная работа 7. Моделирование в системе ANSYS (уч.версия).

Исследование проведения объекта с помощью ANSYS предполагает выполнение следующих действий:

Формирование облика объекта с помощью препроцессора.

Выбор типов конечных элементов, на которые будут разбиваться области объекта.

Задание параметров материала, из которых состоит объект.

Разбиение области на конечные элементы.

Задание краевых условий и нагрузок.

Решение

Просмотр результатов решение с помощью постпроцессора.

Препроцессор ANSYS выполняет большинство своих операций на основе ключевых точек (keypoints), которые могут формироваться непосредственно пользователем как самостоятельные элементы или автоматически при формировании таких элементов как прямоугольник или окружность. При этом для окружности ключевыми точками, сформированными по умолчанию, будут ее центр и четыре точки, делящие окружность на четыре равные линии. Следует отметить, что в ANSYS нет возможности отмены последней операции, поэтому после правильно полученного результата рекомендуется запоминать его в рабочей базе данных.

Порядок выполнения работы.

В некоторых залах лабораторных работ запуск ANSYS происходит автоматически при входе пользователя в систему. Если этого не произошло, выполните следующие два пункта.

1. Запуск программы: Пуск -> Программы -> ANSYS_ED_5.4 -> Interactive
2. После этого появляется окно начального диалога, в котором необходимо указать имя работы (Initial jobname), например Lab1 и нажать кнопку RUN. В следующем появившемся окне нажать кнопку ОК.

После этого появятся основные рабочие окна ANSYS:

Utility Menu - содержит набор утилит для работы с файлами, контроля, выбора, графического контроля, задания общих параметров.

Input - окно, позволяющее задавать команды (или только ее параметры) с клавиатуры.

Main Menu - основное меню первичных ANSYS-функций.

Toolbar - окно наиболее общих функций ANSYS, из которых необходима функция SAVE_DB - сохранение базы данных и, возможно, восстановление базы данных RESUM_DB - восстановление базы данных.

Output - окно вывода сообщений, которое необходимо для контроля.

Graphics - окно формирования графических изображений пре- и постпроцессора.

Лабораторная работа 8. Разработка программы для анализа деформации детали в среде MathLab.

COMSOL Multiphysics - это мощная интерактивная среда для моделирования и расчетов большинства научных и инженерных задач основанных на дифференциальных уравнениях в частных производных (PDE) методом конечных элементов. С этим программным пакетом вы можете расширять стандартные модели использующие одно дифференциальное уравнение (прикладной режим) в мультифизические модели для расчета связанных между собой физических явлений. Расчет не требует глубокого знания математической физики и метода конечных элементов. Это возможно благодаря встроенным физическим режимам, где

коэффициенты PDE задаются в виде понятных физических свойств и условий, таких как: теплопроводность, теплоемкость, коэффициент теплоотдачи, объемная мощность и т.п. в зависимости от выбранного физического раздела. Преобразование этих параметров в коэффициенты математических уравнений происходит автоматически. Взаимодействие с программой возможно стандартным способом – через графический интерфейс пользователя (GUI), либо программированием с помощью скриптов на языке COMSOL Script или языке MATLAB. Данное пособие рассматривает основы работы с программой только через графический интерфейс.

Программа основана на системе дифференциальных уравнений в частных производных. Существует три математических способа задания таких систем:

- *Коэффициентная форма*, предназначенная для линейных и близких к линейным моделей
- *Генеральная форма*, для нелинейных моделей
- *Слабая форма (Weak form)*, для моделей с PDE на границах, ребрах или для моделей использующих условия со смешанными и производными по времени.

Используя эти способы, можно изменять типы анализа, включая:

- Стационарный и переходный анализ
- Линейный и нелинейный анализ
- Модальный анализ и анализ собственных частот

Для решения PDE, COMSOL Multiphysics использует метод конечных элементов (FEM). Программное обеспечение запускает конечноэлементный анализ вместе с сеткой учитывающей геометрическую конфигурацию тел и контролем ошибок с использованием разнообразных численных решателей. Так как многие физические законы выражаются в форме PDE, становится возможным моделировать широкий спектр научных и инженерных явлений из многих областей физики таких как: акустика, химические реакции, диффузия, электромагнетизм, гидродинамика, фильтрация, тепломассоперенос, оптика, квантовая механика, полупроводниковые устройства, сопломат и многих других.

Кроме вышеперечисленного, программа позволяет с помощью переменных связи (coupling variables) соединять модели в разных геометриях и связывать между собой модели разных размерностей.

Для создания и расчета задачи рекомендуется следующая последовательность действий.

1. Выбираем размерность модели, определяем физический раздел в Model Navigator [Навигаторе моделей] (каждому разделу соответствует определенное дифференциальное уравнение) и определяем стационарный или нестационарный анализ температурного поля.
2. Определяем рабочую область и задаем геометрию
3. Задаём исходные данные, зависимости переменных от координат и времени
4. Указываем теплофизические свойства и начальные условия
5. Указываем граничные условия
6. Задаём параметры и строим сетку
7. Определяем параметры решаемого устройства и запускаем расчет.
8. Настраиваем режим отображения
9. Получаем результаты.

Лабораторная работа 9. Интеграция CAD и CAE систем.

CAD- и CAE-ориентированные подходы требуют двойных усилий по созданию и непрерывному поддержанию двух различных моделей одного изделия. Отсутствие автоматизированных средств трансформации из одного типа модели в другой может привести к тому, что модель придется восстанавливать по документации. Это является узким местом в интеграции CAD-CAE. В дополнение, при инженерном анализе часто требуется менять степень детализации (LOD) и/или уровень абстракции (LOA) рассматриваемой модели. Как

только меняются LOD и LOA, необходимо заново проводить процесс трансформации. В качестве решения данных проблем предлагаются варианты общего модельного пространства, а также двунаправленной интеграции CAD-CAE. В данном случае система позволяет CAD-системе автоматически генерировать модели для анализа, а CAE-системе автоматически модифицировать геометрию деталей и проводить новый анализ. Процесс преобразований повторяется, пока не будет достигнут заданный критерий.

Данный метод называется CAD/CAE-интегрированным подходом, который обеспечивает унифицированное моделирование для «бесшовной» интеграции CAD/CAE. На рис. показан поток данных при этом подходе. В основе его лежат следующие технологии: проектирование с использованием фичеров, NMT, многомасштабные представления.

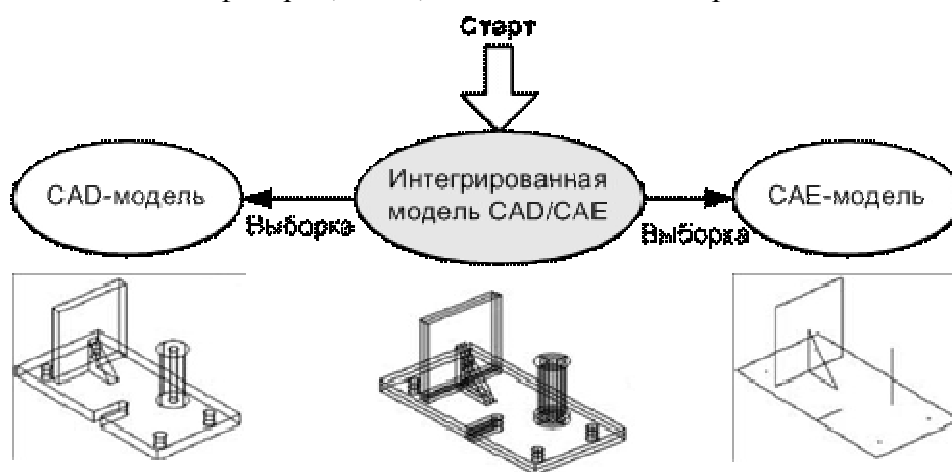


Рисунок. Интегрированный подход CAD\CAE.

При данном подходе, одновременно создаются различные типы геометрических моделей проектирования и анализа для каждой операции моделирования фичера. Все модели интегрируются в одну общую модель. Твёрдотельные модели с различными LOD легко получаются из интегрированной модели. Более того, для каждого LOD можно получить абстрактную NMT модель с различным LOA и передать её в CAE-систему.

В случае CAD/CAE-интегрированного подхода CAD и CAE модели создаются одновременно и объединяются в единую NMT модель. Из объединенной модели CAD и CAE модели получаются с помощью механизма выборки. В дополнение, этот подход поддерживает модели CAD, CAE на различных LOD и LOA. Поэтому используемые здесь технологии это проектирование на основе фичеров, алгоритмы выборки, удаления элементов и изменения размеров, многомасштабные представления.

3. Перечень используемых программных продуктов

1. Windows XP;
2. MS Office XP;
3. SolidWORKS;
4. ABAQUS;
5. ANSYS;
6. AutoCAD;
7. COMSOL.

4. Фонд тестовых и контрольных заданий

Оценочные средства для текущего контроля успеваемости

1 Контрольные вопросы допуска к выполнению лабораторных работ

2 Отчеты о выполнении индивидуальных вариантов заданий лабораторных работ

Оценочные средства для промежуточной аттестации

Вопросы к зачету:

1. Жизненный цикл изделия и место автоматизированных систем в нем.
2. Процесс проектирования и объекты проектирования.
3. Стадии проектирования. Этап технического предложения.
4. Стадии проектирования. Этап эскизного проекта.
5. Стадии проектирования. Этап технического проекта.
6. Стадии проектирования. Этап рабочей конструкторской документации.
7. Преимущества автоматизированного проектирования.
8. Системный подход к проектированию сложных изделий. Блочный-иерархический подход.
9. Иерархические уровни проектирования. Стили проектирования.
10. Описание объекта проектирования. Типы параметров объекта проектирования.
11. Типовая блок-схема процесса автоматизированного проектирования.
12. Типовые задачи проектирования: типовые задачи синтеза, типовые задачи анализа
13. Классификация САПР.
14. Основные графические примитивы системы AutoCAD.
15. Основные команды черчения системы AutoCAD.
16. Основные команды редактирования системы AutoCAD.
17. Понятие блока и работа с размерами в системе AutoCAD.
18. Каркасные и поверхностные геометрические 3D модели.
19. Твердотельные геометрические 3D модели. Грань, ребро, вершина твердого тела.
20. Основные функции создания геометрических моделей в системах твердотельного моделирования.
21. Объектно-ориентированное и параметрическое 3D моделирование.
22. Система твердотельного моделирования SolidWorks. Основные инструменты эскиза.
23. Система твердотельного моделирования SolidWorks. Геометрические взаимосвязи в эскизе.
24. Система твердотельного моделирования SolidWorks. Основные инструменты создания элементов 3D моделей.
25. Система твердотельного моделирования SolidWorks. Моделирование шлицевых валов.
26. Система твердотельного моделирования SolidWorks. Моделирование прямозубых зубчатых колес.
27. Система твердотельного моделирования SolidWorks. Работа со сборками. Виды сопряжений в сборках.
28. Быстрое прототипирование и изготовление изделий, преимущества и недостатки.
29. Процессы быстрого прототипирования и изготовления. Стереолитография.
30. Процессы быстрого прототипирования и изготовления. Отверждение на твердом основании.
31. Процессы быстрого прототипирования и изготовления. Избирательное лазерное спекание.
32. Процессы быстрого прототипирования и изготовления. Трехмерная печать.
33. Процессы быстрого прототипирования и изготовления. Ламинирование.
34. Процессы быстрого прототипирования и изготовления. Моделирование методом наплавления.
35. Применение быстрого прототипирования и изготовления.
36. Стандарты обмена данными между системами САПР.
37. Функциональное проектирование в САПР. Математические модели. Классификация математических моделей.
38. Основные требования к ММ. Адекватность, точность, универсальность, экономичность.
39. Преобразование ММ в процессе анализа.