

Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
Владимирский государственный университет

Е.А. НОВИКОВА

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В МАШИНОСТРОЕНИИ И НАУЧНЫХ ИССЛЕДОВАНИЯХ

Учебное пособие

*Допущено Учебно-методическим объединением вузов
по образованию в области автоматизированного машиностроения
(УМО АМ) в качестве учебного пособия для студентов высших
учебных заведений, обучающихся по направлению подготовки
дипломированных специалистов «Конструкторско-технологическое
обеспечение машиностроительных производств»*

Владимир 2005

УДК 004:681.3.06

ББК 32.81

Н73

Рецензенты

Заслуженный деятель науки и техники РФ,
доктор технических наук, профессор
Тулского государственного университета
А.С. Ямников

Кандидат технических наук, доцент
зав. кафедрой информатики и защиты информации
Владимирского государственного университета
М.Ю. Монахов

Печатается по решению редакционно-издательского совета
Владимирского государственного университета

Новикова, Е. А.

Н73 Компьютерные технологии в машиностроении и научных исследованиях : учеб. пособие / Е.А. Новикова; Владим. гос. ун-т. – Владимир: Ред.-издат. комплекс ВлГУ, 2005. – 108 с. – ISBN 5-89368-559-8.

Изложены основные сведения и понятия в области применения информационных и компьютерных технологий для дисциплин «Основы научных исследований» и «Компьютерные технологии в машиностроении». Рассмотрены информационные технологии, ориентированные на работу в Интернете, программные средства публикации и презентации результатов исследований, программы и технологии автоматизации инженерной деятельности. Приведены описания возможностей, основная функциональность и приемы работы в программах *Excel, Access, PowerPoint, Mathcad*, показана методология автоматизации проектирования в *CAD/CAM/CAE*-системах.

Предназначено для студентов специальностей 120100 – технология машиностроения и 120700 – машины и технология высокоэффективных процессов обработки материалов, высших учебных заведений всех форм обучения.

Табл. 1. Библиогр.: 17 назв.

УДК 004:681.3.06

ББК 32.81

ISBN 5-89368-559-8

© Владимирский государственный университет, 2005

ОГЛАВЛЕНИЕ

| | |
|---|----|
| <i>Введение</i> . Компьютерные технологии. Основные понятия | 5 |
| <i>Часть 1</i> . Компьютерные технологии в научных исследованиях | 7 |
| 1.1. Сбор и обработка научно-технической информации | 8 |
| 1.1.1. <i>Internet</i> . Основные сведения | 10 |
| 1.1.2. Система <i>Microsoft Internet Explorer</i> | 12 |
| 1.1.3. Система оптического распознавания <i>FineReader</i> | 13 |
| 1.1.4. Автоматизированный перевод в системе <i>Prompt</i> | 15 |
| 1.2. Применение компьютерных технологий в научных исследованиях..... | 17 |
| 1.2.1. Компьютерные технологии в теоретических исследованиях | 17 |
| 1.2.2. Компьютерные технологии в экспериментальных исследованиях и моделировании..... | 19 |
| 1.2.3. Обработка результатов исследований | 21 |
| 1.3. Оформление результатов научных работ | 45 |
| 1.3.1. Оформление результатов в текстовых редакторах | 45 |
| 1.3.2. Программа подготовки презентаций <i>PowerPoint</i> | 46 |
| 1.4. Общение с коллегами по научно-исследовательской работе..... | 48 |
| <i>Часть 2</i> . Компьютерные технологии в машиностроении | 51 |
| 2.1. Создание и использование баз данных (СУБД <i>Microsoft Access</i>)..... | 51 |
| 2.1.1. Основы работы с СУБД <i>Microsoft Access</i> | 51 |
| 2.1.2. Реализация ТехноПро на базе СУБД <i>MS Access</i> | 57 |
| 2.2. Автоматизация инженерных расчетов (<i>Microsoft Excel</i> и <i>Mathcad</i>) | 60 |
| 2.2.1. Применение <i>Excel</i> для автоматизации инженерных расчетов..... | 60 |
| 2.2.2. Применение <i>Mathcad</i> для автоматизации инженерных расчетов..... | 75 |

| | |
|---|-----|
| 2.3. Инженерный анализ и автоматизация проектирования..... | 84 |
| 2.3.1. Понятие о <i>CAD/CAM/CAE</i> -системах..... | 84 |
| 2.3.2. Системы геометрического моделирования..... | 88 |
| 2.3.3. Системы инженерного анализа методом конечных элементов..... | 92 |
| 2.3.4. Системы автоматизированного производства..... | 95 |
| 2.3.5. Системы управления данными об изделии..... | 101 |
| 2.3.6. Сетевая работа над проектом..... | 103 |
| Рекомендательный библиографический список..... | 106 |
| <i>Образовательные ресурсы Internet</i> | 107 |

ВВЕДЕНИЕ

В любой области деятельности человека технология – это совокупность знаний о способах и средствах проведения производственных процессов, под которыми следует обобщенно понимать выполняемую работу. В производственных процессах важнейшим ресурсом является информация. В этой связи под термином «информационная технология» понимают современные виды информационного обслуживания, основанные на использовании средств вычислительной техники, связи, множительных средств и оргтехники.

Компьютерные технологии являются частью информационных и обеспечивают сбор, обработку, хранение и передачу информации с помощью ЭВМ. Основу современных компьютерных технологий составляют три достижения: возможность хранения информации на машинных носителях, развитие средств связи и автоматизация обработки информации с помощью компьютера.

Практически компьютерные технологии реализуются применением программно-технических комплексов, состоящих из персональных компьютеров или рабочих станций с необходимым набором периферийных устройств, включенных в локальные и глобальные вычислительные сети и обеспеченных необходимыми программными средствами.

Использование названных элементов увеличивает степень автоматизации как научных исследований, так и решения задач в машиностроении. Компьютерные технологии повышают уровень эффективности проводимых работ за счет следующих факторов:

1. Упрощение и ускорение процессов обработки, передачи, представления и хранения информации.
2. Увеличение объема полезной информации с накопителем типовых решений и обобщением опыта научных разработок.
3. Обеспечение глубины, точности и качества решаемых задач. Возможность реализации задач, ранее не решаемых. Постановка исследований и получение результатов, недостижимых другими средствами.
4. Возможность анализа большого числа вариантов синтеза объектов и принятия решений.
5. Сокращение сроков разработки, трудоемкости и стоимости НИР при улучшении условий работы специалистов.

Информационные технологии в настоящее время используются практически во всех сферах деятельности человека. Задача курсов «Компьютерные технологии в машиностроении» и «Основы научных исследований» обобщить знания по компьютерным технологиям применительно к науке и машиностроению.

Предлагаемое вниманию читателей учебное пособие посвящено краткому описанию возможностей современных компьютерных технологий, призванных облегчить и автоматизировать труд инженеров, научных работников. Данное пособие не является систематическим руководством пользователя, а скорее представляет собой путеводитель, призванный помочь студенту или начинающему инженеру сориентироваться в необъятном мире современного программного обеспечения. Поэтому автор ставил перед собой задачу показать не весь спектр существующих программ и информационных технологий, а лишь самые необходимые, наиболее популярные и зарекомендовавшие себя приложения, которые просто обязан знать современный инженер.

Пособие разделено на две части в соответствии со сферами применения компьютерных технологий. В первой части представлен обзор компьютерных технологий, наиболее широко востребованных при проведении и организации научных исследований. Вторая часть ориентирована на технологии автоматизации инженерной деятельности. Следует иметь в виду, что это деление весьма условно, поскольку, с одной стороны, научная деятельность в области технических наук немыслима без моделирования и расчетов в математических пакетах или *CAD/CAM/CAE*-системах, а, с другой стороны, инженер должен владеть основами поиска, упорядочения и представления научно-технической информации. Тем не менее такое разделение полезно по методическим соображениям, в соответствии с программами дисциплин «Основы научных исследований» и «Компьютерные технологии в машиностроении», читаемых на кафедре «Технология машиностроения» ВлГУ.

При составлении учебного пособия были использованы пособия, руководства и справочная литература [1 – 17], а также информационные и образовательные ресурсы *Internet* (адреса приведены в конце книги), которые рекомендуются читателям для самостоятельного изучения.

ЧАСТЬ 1

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В НАУЧНЫХ ИССЛЕДОВАНИЯХ

Известно, что наука – это сфера деятельности, направленная на получение новых знаний, которая реализуется с помощью научных исследований. Целью научных исследований являются изучение определенных свойств объекта (процесса, явления) и на этой основе разработка теории или получение необходимых для практики обобщенных выводов.

По целевому назначению научные исследования делят на фундаментальные, прикладные и разработки.

Фундаментальные связаны с изучением новых явлений и законов природы, с созданием новых принципов исследований (физика, математика, биология, химия и т.д.).

Прикладные исследования – это нахождение способов использования законов природы и научных знаний, полученных в фундаментальных научных исследованиях и в практической деятельности человека.

Разработки – это процесс создания новой техники, систем, материалов и технологий, включающий подготовку документов для внедрения в практику результатов прикладных научных исследований.

Реализация целей научных исследований выполняется на основе методов. Метод – это способ достижения цели, программа построения и применения теории. Методы научных исследований делят на следующие группы: эмпирические, экспериментальные и теоретические. Особую группу составляют методы научно-технического творчества.

Эмпирические исследования выполняются с целью накопления систематической информации о процессе. При этом используются методы – наблюдение, регистрация, измерение, анкетный опрос, тесты, экспертный анализ.

Экспериментальный уровень научных исследований – это изучение свойств объекта по определенной программе.

Теоретические исследования проводятся с целью разработки новых методов решения научно-технических задач, обобщения и объяснения эмпирических и экспериментальных данных, выявления общих закономерностей и их формализации.

На двух последних уровнях используются методы моделирования, анализа и синтеза, логические построения (предположения, умозаключения), аналогии, идеализации.

Рациональная организация научно-исследовательских работ строится с использованием принципов системного подхода и может содержать следующие этапы:

1. Сбор и обработка научно-технической информации.
2. Проведение теоретических (включая моделирование) и экспериментальных исследований.
3. Обработка результатов исследования.
4. Представление и оформление результатов.

Исходя из задач исследования и порядка их реализации, можно определить следующие основные направления рационального применения компьютерных технологий в научных исследованиях:

1. Сбор, хранение, поиск и выдача научно-технической информации.
2. Подготовка программ научных исследований, подбор оборудования и экспериментальных устройств.
3. Математические расчеты.
4. Решение интеллектуально-логических задач.
5. Моделирование объектов и процессов.
6. Управление экспериментальными установками.
7. Регистрация и ввод в ЭВМ экспериментальных данных.
8. Обработка одномерных и многомерных (изображений) сигналов.
9. Обобщение и оценка результатов исследований.
10. Оформление и представление итогов научных исследований.
11. Управление научно-исследовательскими работами.

Наиболее эффективно, когда эти задачи реализуются в рамках автоматизированных систем научных исследований (АСНИ).

1.1. Сбор и обработка научно-технической информации

При системном подходе научные исследования начинаются со сбора и предварительной обработки информации по теме исследования. Эта информация может включать сведения о достижениях в исследуемой области, об оригинальных идеях, открытых эффектах, научных разработках, технических решениях и т.д.

Целью данного этапа является получение ответов на следующие вопросы:

1. Какие авторы или научные группы занимаются аналогичной темой?
2. Каковы известные решения по исследуемой теме?
3. Какими известными методами и средствами решаются исследуемые проблемы?
4. Каковы недостатки известных решений и какими путями их пытаются преодолеть?

Углубленное изучение информации по предмету исследования позволяет исключить риск ненужных затрат времени на уже решенную проблему, детально изучить весь круг вопросов по исследуемой теме и найти научно-техническое решение, отвечающее высокому уровню.

Основным источником информации являются научные документы, которые по способу представления могут быть текстовыми, графическими, аудиовизуальными.

Научные документы подразделяются на первичные и вторичные, опубликованные и неопубликованные. Первичные документы – это книги, брошюры, периодические издания (журналы, труды), научно-технические документы (стандарты, методические указания). Важное значение здесь имеет также патентная документация, под которой подразумевают издания, содержащие сведения об открытиях, изобретениях и т.п. К неопубликованным первичным документам относят: научные отчеты, диссертации, депонированные рукописи и т.п. Они содержатся в фонде ВНИИЦентра. Вторичные документы содержат краткую обобщенную информацию из одного или нескольких первичных документов: справочники, реферативные издания, библиографические указатели и т.п.

Сбор и обработка научно-технической информации может быть выполнена следующими способами: анкетирование, собеседование, экспертный опрос и т.д., но основой является работа с научно-техническими документами, которая включает поиск, ознакомление, проработку документов и систематизацию информации. Поиск выполняется по каталогам, реферативным и библиографическим изданиям. Автоматизация этой процедуры обеспечивается использованием специализированных информационно-поисковых систем библиотек и научно-исследовательских институтов, электронных каталогов, поиском в машиночитаемых базах данных, а также с помощью программ поиска в сетях *Internet*.

Компьютерные технологии широко используются при работе с научной литературой на бумажных носителях. Сканирование текстов позволяет создавать электронные документы, базы данных и книги. Специальные программы-переводчики позволяют осуществлять перевод любого текста с основных европейских языков на русский и обратно.

Учитывая стремительное возрастание объемов научно-технической информации в *Internet*, обратим внимание на основы работы с данной глобальной системой, рассмотрим систему оптического распознавания *Fine-Reader* и приемы автоматизированного перевода в системе *Stylus*.

1.1.1. *Internet*. Основные сведения

Internet – это всемирная корпоративно управляемая совокупность компьютерных сетей, обменивающихся информацией и связанных телефонными, оптоволоконными и другими кабелями, а также радиоканалами, в том числе спутниковыми. *Internet* предоставляет в распоряжение пользователей огромное количество информационных ресурсов. Одних только *Web*-документов в сети *Internet* десятки миллиардов, кроме этого в нем размещаются программы, изображения, звуковые файлы и т.д.

Работа в сети может выполняться в различных режимах. Наиболее часто используется протокол обмена данными *TCP/IP*, который обеспечивает передачу сообщений фрагментами, сборку и проверку переданного документа; *IP* обеспечивает доставку информации конкретному адресату. Таким образом, каждый компьютер в сети имеет свой уникальный *IP*-адрес в виде цифровой последовательности XXX.XXX.XXX.XXX.

Пользователь сети (организация или частное лицо) в *Internet* идентифицируется именем пользователя и именем домена, разделенных знаком @: XXXXXX.XXX@XXXXX.XXX.

Internet предоставляет следующие возможности:

1. Поиск и просмотр информационных документов.
2. Доступ к БД (*WAIS*).
3. Связь и работу с другими ПК (*Telnet*).
4. Группы новостей, электронные бюллетени и т.д.
5. Средства, в которых пользователи могут направлять свои сообщения и знакомиться с имеющимися.
6. Электронная почта (*Internet Mail*).

7. Почтовые списки – средства рассылки информации. Разговор текстом на экране в реальном времени (программа *IRC – Internet Relay Chat*).

9. Передача файлов, в том числе программных (система *FTP*).

В основном возможности *Internet* реализуются при наличии ОС: *WS 95,98, WS NT, UNIX, Linux, Solaris*. Большинство информационных ресурсов находятся на серверах узлов *Internet*, имеющих универсальные *URL*-адреса, которые в зависимости от системы размещения начинаются с записи: *http://* – для *Web*-серверов и *ftp://, news://* – для серверов *FTP* и групп новостей соответственно.

Для поиска *URL*-адресов можно пользоваться справочником «Желтые страницы *Internet*». Его можно найти в *Internet* по адресу: *http://www.jellow.com*. Имеется этот справочник и на компакт-диске.

Наиболее эффективной службой для поиска информации в *Internet* является *World Wide Web* (всемирная паутина) или просто *Web*, использующая гипертекстовое представление информации (протокол *HTTP*).

Гипертекст – это представление документа в виде узлов и связей. Если в узлах помимо текста, таблиц, графики имеются аудио- и видеофайлы, то *Web* превращается в мультимедийную систему.

Указанная структура документа с помощью гиперссылок позволяет более эффективно выполнять поиск необходимой информации.

В *Web* используются понятия:

Web-страница – отдельный файл;

Web-сайт – группа документов, объединенных по смыслу;

HTML – язык разметки гипертекста;

VRML – язык моделирования трехмерной и подвижной информации;

Web-браузер – программа поиска и просмотра *HTML*-документов и других информационных ресурсов *Internet*. Наиболее распространены: *MS Internet Explorer* и *Netscape Navigator*.

Кроме указанного следует назвать *JAVA* – объектно-ориентированный язык для создания распределенных прикладных *Web*-систем, позволяющий выполнять программы непосредственно на ПК клиента. *CGI стандарт* запускает внешние прикладные программы на *Web*-серверах.

В среде *Web* для повышения эффективности поиска целесообразно кроме браузеров применять поисковые серверы и каталоги *Web*, использующие ключевые слова.

Наиболее известные российские поисковые системы: <http://www.rambler.ru>, <http://www.yandex.ru>, <http://www.aport.ru>. Эти системы обеспечивают поиск актуальной, уникальной и специальной информации. Они дают большой охват ресурсов *WWW*, но при этом качество ссылок, предоставляемых по результатам поиска, достаточно низкое. Качество полученного результата зависит от предмета поиска и точности формулировки запроса.

В настоящее время все высшие образовательные учреждения имеют свои сайты, на которых размещена информация о проводимых научно-исследовательских работах, конференциях. Большое количество печатных научных журналов и газет имеют свои электронные версии в *Internet*. Перестало быть новшеством проведение научных конференций в *Internet* с последующим электронным опубликованием материалов во «всемирной паутине». Ряд ученых и преподавателей имеют свои персональные сайты, где в режиме свободного доступа можно познакомиться с их научными работами. Растет количество электронных библиотек, однако объем литературы по техническому направлению в них мал.

1.1.2. Система *Microsoft Internet Explorer*

Для начала поиска информации в *Internet* можно использовать *Internet Explorer*. Окно этого браузера имеет стандартный для *Windows*-приложений вид с меню и инструментальными панелями.

Перед началом работы необходимо ввести в строку **Адрес URL** – адрес сервера или выбрать его из списка.

В пункте **Переход** необходимо задать вид просмотра: Страница *Web*, **Почта** или **Новости**.

После завершения поиска (может быть до нескольких минут) в рабочей части экрана появляется начальная страница информации, где выделены цветом и подчеркнуты гиперсвязи, что можно использовать для быстрого знакомства с документом и поиска нужной информации. Для исключения повторов при просмотре цвет использованных связей меняется.

Поиск по нужному слову в длинных документах можно выполнить с помощью пункта **Правка/Найти**. Возвращение назад, прекращение поиска выполняются кнопками инструментальной панели. При поиске можно открыть несколько документов, которые будут храниться в кэш-памяти на винчестере. Их имена находятся внизу раскрывающегося окна. Сохранение

необходимой информации выполняется отметкой необходимого фрагмента документа, копированием или записью с использованием пункта **Файл/Сохранить как....**

При сохранении нужно учитывать тип информации (текст, таблица, рисунок и т.д., так как браузер помещает скопированные файлы в соответствующие приложения (текстовый редактор, электронные таблицы и т.п.). Для неинсталлированных приложений необходима установка дополнительных программ вьюверов-просмотрщиков.

1.1.3. Система оптического распознавания *FineReader*

Программа *FineReader* предназначена для автоматизации ввода в ПК типографских документов. Работает в среде *Windows*. Обеспечивает распознавание до 99,6 % символов.

Программа способна выполнять сканирование и распознавание текстов на разных языках, в том числе и смешанных двуязычных текстов. С ее помощью можно выполнять пакетную обработку многостраничных документов, а также настраивать режим распознавания для улучшения соответствия электронного документа бумажному оригиналу при плохом качестве последнего или использования в нем шрифтов, далеких от стандартных.

Главное окно системы (как и все *Windows*-приложения) включает управляющее меню, инструментальные панели и рабочее поле.

Работа в *FR* осуществляется в три этапа: сканирование, распознавание, редактирование. Сканирование выполняется при физически и программно установленном сканере, регистрация которого в системе происходит через пункт **Сервис/ Опции/ Сканер**.

Для качества распознавания перед началом работы в диалоговом окне установки сканера (пункт **Scan&Read**) задаются необходимые параметры: тип изображения, формат, ориентация, разрешение и яркость.

Выполнение сканирования производится по команде: пункт **Scan&Read/ Сканирование**.

После завершения процесса в рабочем окне появляются окна: **Изображение**, **Крупный план** и **Текст**. Полученный файл для дальнейшей обработки необходимо сохранить в формате *FRF* (пункт **Файл/Сохранить как...**).

Распознавание, т.е. формирование системой истинного образа документа, включает настройку системы на документ, разбиение документа на

блоки и распознавание блоков. Настройка заключается в задании параметров распознавания (язык текста, тип и расположение) в диалоговых окнах **Сервис/Опции/Распознавание и Сервис/Опции/Страница**.

Разбиение документа на блоки (текстовые, табличные и т.д.), т.е. зоны, ограниченные рамками с соответствующей нумерацией выполняется автоматически или вручную командами пункта **Редактор/Выделить блоки**. Распознавание разделенного на блоки документа выполняется командой **Scan & Read/Распознавание**. Во время распознавания обработанная часть выделяется цветом в окне «Изображение», а после окончания появляется окно «Текст» с содержанием документа.

Редактирование документа включает корректировку, орфографический контроль и сохранение текста. Корректировка выполняется для первичного уточнения текста командами пункта **Редактор**, т.е. средствами встроенного текстового редактора.

Проверка орфографии производится с помощью встроенной в **FR** системы **Lingvo Corrector**, которая позволяет находить ошибки и неуверенно распознанные слова, корректировать ненужные пробелы и т.п. Эта операция выполняется через пункт **Сервис/Орфография**.

Сохранение документа (пункт **Файл/Сохранить как...**) может быть выполнено:

- для текстовых блоков – в форматах **TXT, RTF**;
- для таблиц и форм – в форматах **CSV, DBF, XLS**.

К дополнительным возможностям **FR** можно отнести:

1. Обучение системы распознаванию «плохих» текстов.

При хорошем полиграфическом качестве документа используется режим омнифонт (см. **Сервис/Опции/Распознавание**), в котором каждый символ автоматически сравнивается с имеющимися в базе данных образцами.

При плохом качестве сканируемого документа используется распознавание с обучением (включается соответствующий флаг в окне **Сервис/Опции/Распознавание**). При этом пользователь, просматривая документ, выделяет плохо различимые символы и объявляет их названия системе, которые используются при дальнейшей работе.

2. Режим пакетной обработки. Применяется при вводе больших объемов однотипных документов практически без участия пользователя. Реализуется со сканером, имеющим механизм автоподачи страниц. Включает-

ся кнопкой *Scan&Read* на ИП. Этот режим очень удобен для подготовки табличных файлов перед вводом их в БД.

3. Распознавание форм. Обеспечивает ввод переменной информации с однотипных документов. Здесь блоки делятся на реперные (нераспознаваемые) и распознаваемые, затем формируется шаблон.

4. **FR** может быть встроен в интегрированный пакет *Stylus Lingvo Office*, реализующий законченную технологию обработки иностранных текстов: распознавание, перевод, проверка орфографии, что обеспечивает получение готового документа на необходимом языке.

1.1.4. Автоматизированный перевод в системе *Prompt*

К средствам автоматизации перевода можно отнести два вида программ: электронные словари и программы перевода.

Электронные словари представляют собой средства для перевода отдельных слов, отображаемых на экране или имеющих в документе. Удобство их использования состоит в возможности немедленно получить перевод неизвестного слова без поиска его в отдельном толстом томе.

Программы перевода автоматизируют перевод текста.

Технический текст в отличие от художественного использует ограниченное число языковых конструкций и более ориентирован на однозначную интерпретацию. Среди используемых лексических единиц встречается большое число технических терминов, имеющих совершенно определенный смысл в рамках изучаемых научного и технического направлений.

Программы автоматизированного перевода имеет смысл использовать для перевода технических текстов в следующих случаях:

- при абсолютном незнании иностранного языка;
- при необходимости получить перевод быстро, даже ценой снижения его качества (например перевод *Web*-документов);
- для перевода на иностранный язык (умения читать иноязычные тексты недостаточно, чтобы научиться объясняться на иностранном языке);
- для быстрого создания первоначального черновика («подстрочника»), используемого в ходе подготовки полноценного перевода.

Система *Prompt* обеспечивает перевод с основных европейских языков на русский и обратно.

Окно *Prompt* имеет обычный для приложений *Windows* вид. Главное меню помимо известных пунктов включает команду Перевод.

Автоматизированный перевод обычно включает этапы:

1. Первоначальный перевод.
2. Повышение качества перевода.
3. Постредактирование.

Первоначальный перевод начинается с загрузки исходного текста или его непосредственного набора. В пункте **Вид** устанавливается горизонтальное или вертикальное представление исходного и переведенного текстов.

Перевод может выполняться по абзацам, выделенным фрагментам, всего текста и т.д. Способ задается пунктом **Перевод**. Можно перевести отдельное слово или выделенную фразу при установке на них указателя мыши.

Повышение качества перевода достигается за счет: резервирования слов, не требующих перевода (названия, специальные термины, сокращения). Здесь иногда удобно также применять транслитерацию слов: запись буквами нужного языка (Иванов – *Ivanov*); подключения специализированных словарей по определенной тематике (например по информатике) и пользовательских словарей, обеспечивающих настройку на конкретный текст; пополнения словарей.

Для работы со словарями используются вкладки списков словарей, незнакомых и зарезервированных слов, расположенных в нижней части экрана. Данная информационная панель включается через пункт **Вид**. При этом можно переносить слова в указанном списке из текста или из списка в список.

Для качества перевода его целесообразно выполнить дважды, при этом в первом варианте перевода выясняют:

1. Какие слова не переведены (выделяются другим цветом)?
2. Какие слова и словосочетания переведены неудачно?
3. Какие слова следует оставить без перевода?

Такие слова включают в пользовательский словарь, резервируют или исправляют средствами редактора *Stylus*. После этого перевод повторяют.

Постредактирование заключается в смысловом редактировании переведенного текста. Здесь используются возможности внутреннего редактора системы с использованием команд пункта **Правка**. Для облегчения поиска абзацев между переводом и оригиналом существует постоянная связь.

Отредактированный текст может быть дополнительно проверен с помощью справочных словарей или систем проверки орфографии. Получен-

ный перевод записывается в файл или распечатывается непосредственно из *Stylus* (можно сохранить и оригинал).

Система *Stylus* может взаимодействовать с другими приложениями *Windows* следующими способами: переводом текста в буфере обмена; установлением связи с приложениями; встраиванием перевода внутрь приложений.

1.2. Применение компьютерных технологий в научных исследованиях

1.2.1. Компьютерные технологии в теоретических исследованиях

Основной задачей теоретических исследований является создание теории по исследуемой проблеме, включающей объяснение явлений с использованием математического аппарата или качественных правил.

Объем исследований зависит от специфики и сложности проблемы. В общем случае может включать этапы:

1. Постановка задачи, где определяются цели исследования, наиболее эффективные пути реализации. Иногда формируется гипотеза, предварительно объясняющая явление.

2. Разработка модели процесса функционирования изучаемого объекта. В теоретических исследованиях обычно используются математические, информационные или логические модели явления.

3. Выбор методов построения модели и их проверка.

4. Разработка алгоритмов и программных средств реализации моделей.

5. Выполнение математических расчетов или обработка информационных алгоритмов.

6. Анализ полученных результатов с помощью логических рассуждений и выводов, формулирование результатов исследований.

Эффективность теоретических исследований в значительной степени зависит от используемых исследователем методов. При этом обычно применяются:

1. Известные общенаучные методы: абстрагирование, идеализация, формализация, анализ и синтез, обобщения и т.п.

2. Математические методы: аналитические, численные, оптимизационные, вероятностно-статистические.

3. Эвристические приемы и методы: инверсия, универсальность, самообслуживание, ассоциации, аналогии и т.д.

4. Логические методы и правила, к числу которых можно отнести: правила вывода сложных понятий из простых, установление истинности, выявление непротиворечивости и т.п.

Компьютерные технологии находят широкое применение в реализации научных задач. Наиболее часто их используют в проведении математических расчетов. Программное обеспечение для данного направления условно делится на следующие категории:

1. Библиотеки программ для численного анализа, которые также делятся на библиотеки общего назначения (пакеты *SSP*, *NAG*) и узкоспециализированные пакеты, ориентированные на решение определенного класса задач (*Micro Way* – матрицы, преобразование Фурье).

2. Специализированные системы для математических расчетов и графического манипулирования данными и представления результатов (*Phaser* – дифференциальные уравнения, *Statgraf* – статистический анализ), *Eureka*, *Statistica*.

3. Диалоговые системы математических вычислений с декларативными языками, позволяющими формулировать задачи естественным образом (*Mu-Math*, *Reduce*, *MathCad*, *Matlab*, *Mathematica*).

4. Электронные таблицы, которые позволяют выполнять различные расчеты с данными, представленными в табличной форме (*Supercalc*, *Quattro Pro*, *Lotus*, *Excel*).

В реализации эвристических методов рациональное применение ВТ связано с использованием методов морфологического анализа (таблиц), ассоциативных методов (каталога, гирлянд случайностей и ассоциаций), с помощью которых генерируется большое число вариантов решения задачи, а затем производится их оценка и выбор рационального.

Теоретические исследования технических проблем в некоторых случаях целесообразно проводить с использованием автоматизированной системы решения изобретательских задач – *АРИЗ*, которая охватывает все этапы технического творчества от анализа технических систем до поиска вариантов решения.

Наиболее сложной является компьютеризация логико-лингвистических методов научных исследований. Решение проблемы в этом направлении обеспечивается средствами, включающими элементы искусственного ин-

теллекта. Это системы автоматизированного перевода (**СОКРАТ**, **PROMT**, **Stylus**), интеллектуальные прикладные программные продукты, расчетно-логические системы, системы поддержки принятия решения и различные экспертные системы.

Интеллектуальные программные средства дают возможность решать задачи по описанию процесса с помощью непроцедурного языка без программирования алгоритма. При этом система сама формирует математическую модель исследования и определяет необходимые программные модули для ее реализации (**ПРИЗ**, **СПОРА**, **МАВР**).

Расчетно-логические системы предназначены для коллективного решения общих исследовательских задач при выполнении локальных задач на отдельных рабочих местах за счет координируемого взаимодействия по каналам связи (Система комплексного планирования **ГРАНИТ**).

Экспертные системы – это программные комплексы, использующие знания в предметной области и способные на их основе с помощью логических (рассуждений) правил формулировать выводы о состоянии системы, основанные на анализе модели представления экспертов о закономерностях ее функционирования. Экспертные системы обычно включают: подсистему общения, базу данных с подсистемой накопления знаний, решающий блок, подсистему объяснения. Данные системы наиболее эффективны для анализа процессов и явлений, которые сложно представить математической моделью.

1.2.2. Компьютерные технологии в экспериментальных исследованиях и моделировании

Основными задачами экспериментальных исследований могут быть:

1. Целенаправленное наблюдение за функционированием объекта для углубленного изучения его свойств.
2. Проверка справедливости рабочих гипотез для разработки на этой основе теории явлений.
3. Установление зависимости различных факторов, характеризующих явление, для последующего использования найденных зависимостей в проектировании или управлении исследуемыми объектами.

Выделяют этапы подготовки эксперимента, проведения исследований и обработки результатов. На подготовительном этапе определяются цели и задачи экспериментального исследования, разрабатываются методика и

программа его выполнения. Этот этап включает также подбор необходимого оборудования и средств измерений.

При разработке программы исследования стремятся к меньшему объему и трудоемкости работ, упрощению эксперимента без потери точности и достоверности результатов. В этой связи данный этап требует решения задачи определения минимального числа опытов (измерений), наиболее эффективно охватывающего область возможного взаимодействия влияющих факторов и обеспечивающего получения их достоверной зависимости.

Данная задача решается средствами раздела математической статистики – планирование эксперимента, который представляет необходимые методы для рациональной организации измерений, подверженных случайным ошибкам.

Этап проведения собственно исследований определяется спецификой изучаемого объекта. По характеру взаимодействия средств эксперимента с объектом различают обычные и модельные экспериментальные исследования. В первом случае взаимодействие оказывается непосредственно на объект, во втором – на заменяющую его модель.

Метод моделирования объектов и процессов является основным в научном эксперименте. Различают физическое и математическое моделирование.

Физическое моделирование выполняется на специальных установках. При этом вычислительная техника используется для управления процессом эксперимента, сбора регистрационных данных и их обработки.

Математическое моделирование, в широком смысле, включает исследования не только с помощью чисто математических моделей. Здесь используются также информационные, логические, имитационные и другие модели и их комбинации. В этом случае математическая модель представляет собой алгоритм, включающий определение зависимости между характеристиками, параметрами и критериями расчета, условия протекания процесса функционирования системы и т.д. Данная структура может стать моделью явления, если она с достаточной степенью отражает его физическую сущность, правильно описывает соотношение свойств и подтверждается результатами проверки. Применением математических моделей и вычислительной техники реализуется один из наиболее эффективных методов научных исследований – вычислительный эксперимент, позволяющий изучать поведение сложных систем, которые трудно физически смоделировать. Часто это связано с большой сложностью и стоимостью

объектов, а в некоторых случаях невозможностью воспроизвести эксперимент в реальных условиях.

Для математического моделирования целесообразно использовать программные средства известных фирм, разработанные высококвалифицированными специалистами с использованием последних достижений прикладной математики и программирования. Возможности современных программных средств в части машинной графики, включая параметризацию, использование методов конечных элементов и способов «фрактала» и «морфинга», цветовой динамики, мультипликации и т.п., обеспечивают достаточную наглядность результатов.

Наиболее широкое применение находят компьютерные технологии для: логического, функционального и структурного моделирования электронных схем; моделирования и синтеза систем автоматического управления (САУ); моделирования механических и тепловых режимов конструкций, механики газов и жидкостей. При этом используют как сотни функционально ориентированных программных средств (*MICRO-Logic, PC-LOGS* из *P-CAD, ANSYS, DesignLAB*), так и системы универсального применения (электронные таблицы *Excel, QuattroPro*, система *Mathcad*).

1.2.3. Обработка результатов исследований

Выполнение теоретических или экспериментальных исследований предусматривает получение и регистрацию больших объемов информации, которая может быть представлена в виде:

- а) массивов числовых данных как результатов дискретных измерений;
- б) комплексов одномерных или многомерных сигналов.

Обработка числовых данных в зависимости от характера исследований может включать:

1. Выявление грубых измерений. Здесь используются:
 - а) правило трех сигм $X_i > X \pm 3 \delta$ (где X – среднеарифметическое значение, δ – среднеквадратичное отклонение);
 - б) величина малой вероятности результата;
 - в) эмпирические критерии ошибок.
2. Анализ систематических и случайных погрешностей. Систематические ошибки обусловлены определенными постоянными факторами и определяются по таблицам, графикам для каждого прибора. Учет случайных

погрешностей проводится с использованием теорий вероятности и случайных ошибок.

3. Графическую обработку результатов измерений, которая выполняется после исключения погрешностей числовых данных и позволяет наглядно выявлять функциональные зависимости исследуемых факторов.

4. Вывод эмпирических зависимостей, т.е. зависимостей между взаимодействующими величинами в виде алгебраических или других типов выражений, соответствующих экспериментальным кривым. В данном случае используются методы средних и наименьших квадратов, различные методы аппроксимации и интерполяции на основе полиномов, рядов, сплайн-функций и т.п., корреляционный и регрессионный анализы.

Обработка сигналов выполняется с целью выделения из них интересующей исследователя информации. При этом для одномерных сигналов характерны следующие операции:

1. Визуализация результатов измерений, т.е. графическое представление сигналов с использованием различных систем координат и масштабированием.

2. Измерение параметров сигнала (периоды колебаний, амплитуды и т.п.).

3. Обработка сигнала заключается в исключении содержащихся в нем случайных помех. Здесь используются методы сглаживания данных и фильтрации. Исследования свойств сигнала во многих случаях проводятся с применением методов спектрального анализа. При этом определяются частотные составляющие, скрытые периодичности и т.п. Классическим средством спектрального анализа является программная реализация преобразований Фурье. В обработку сигналов входят также процедуры оценки передаточных функций (например, каналов связи или САУ).

4. Классификация и идентификация сигналов. Эти процедуры дают информацию для различных систем контроля и диагностики.

Обработка многомерных сигналов связана с анализом изображений (рентгеновских, ультразвуковых, оптических и т.п.). Многие задачи здесь близки анализу одномерных сигналов. Типичное математическое обеспечение этого процесса включает решение следующих функций:

1. Ввод, сжатие и запись в виде файлов.

2. Визуализация изображения с возможностью его контрастирования и использования цветовой гаммы.

3. Измерения на изображении (вычисление размеров, площадей, периметров и других характеристик объектов).

4. Фильтрация изображения выполняется для подавления в нем случайных составляющих.

5. Статистический анализ изображения по гистограммам яркости, что позволяет определить степень его искажения.

6. Классификация изображения основывается на измерении характеристик объектов, что позволяет их идентифицировать и распознать.

Необходимо отметить, что на этапе обработки результатов исследований наибольшее применение находят программные средства, обеспечивающие выполнение математических расчетов с использованием теории вероятности, теории ошибок, математической статистики и т.п., а также программные средства векторного и растрового анализов изображений.

Табличный процессор Excel

Информация в научных исследованиях часто представляется в табличной форме. Обработка такой информации эффективно выполняется с использованием табличных процессоров или электронных таблиц. Электронные таблицы применяются на всех этапах выполнения научных исследований, но наиболее целесообразно их использование при выполнении математических расчетов, математическом моделировании, численном эксперименте и отработке данных.

Применение электронных таблиц упрощает работу с данными и позволяет получать результаты без проведения расчетов вручную или специального программирования. В научно-технических задачах электронные таблицы эффективно используются для проведения однотипных расчетов над большими наборами данных; автоматизации итоговых вычислений; решения задач путем подбора значений параметров, табулирования формул; обработки результатов экспериментов; проведения поиска оптимальных значений параметров; подготовки табличных документов; построения диаграмм и графиков по имеющимся данным.

Программа *Excel* предназначена для работы с таблицами данных, преимущественно числовых.

Выполнение математических расчетов в электронных таблицах основано на возможности связывания числовых значений клеток с помощью математических операторов и встроенных функций. Табличный процессор *Excel* предоставляет возможность работы с математическими, статистиче-

скими, логическими, информационными и другими категориям функций. Выбор необходимой функции выполняется с помощью **Мастера** функций командой **Вставка/Функция** или с использованием стандартной панели инструментов. При этом происходит помещение функции в «активную клетку».

В части расчетов *Excel* позволяет выполнять:

1. Реализацию численных методов решения дифференциальных уравнений, алгебраических уравнений и их систем.
2. Обработку векторных и матричных массивов информации.
3. Оптимизационные расчеты, включая методы математического программирования (линейное и т.д.).
4. Операции с комплексными числами.

При этом расчеты сводятся к вычислению промежуточных результатов в соответствующих колонках таблиц.

Моделирование и численный эксперимент в электронных таблицах основаны на возможности автоматического пересчета результатов и их связанном графическом отображении. Для наиболее простых случаев используется анализ по способу “что-если”, когда поочередно меняются значения переменных функций.

Вариантом названного анализа является метод подбора параметра. Требуемые значения функции при этом находятся за счет варьирования переменными, от которых она зависит. Метод реализуется командой **Сервис/Подбор параметра** через соответствующее диалоговое окно. При этом может быть выполнено несколько операций с заданием величины числа. Эта операция может быть реализована графически с выделением отображения переменной (*Ctrl+ЩЛ*) и его изменением БМ.

Более сложный анализ для нахождения рационального численного решения при большем числе условий и ограничений выполняется методом «поиск решения». Эта задача решается диалогом в пункте **Сервис/Поиск решения**. (Режим должен быть предварительно включен пунктом **Сервис/Настройка**).

При обработке данных, полученных по результатам исследований, *Excel* может быть использован:

1. Для расчета среднеарифметического и среднеквадратичного отклонений наборов данных при выявлении грубых ошибок измерений. Здесь применяются функции **СРЗНАЧ**, **КВАДРОТКЛ** и т.п.

2. Статистического анализа данных. При этом может быть выполнено:

- определение минимального (максимального) значения (функции МИНИ, МАКС) ряда данных,
- стандартное отклонение (СТАНДОТКЛОН);
- корреляционный, дисперсионный анализы, анализ Фурье и т.п. через команду **Сервис/Анализ данных**, включаемую через диалог **Сервис/Надстройки**.

3. Графического отображения результатов измерений с использованием прямоугольных и логарифмических шкал осей. Последние могут быть установлены через диалоговое окно “Форматирование оси”, открываемое двойным щелчком по соответствующей оси.

Для удобства представления результатов на график может быть нанесена сетка – пункт **Вставка/Сетка** и включены планки погрешностей – пункт **Вставка/Планки погрешностей**.

4. Определения коэффициентов эмпирических линейных зависимостей (функция **ТЕНДЕНЦИЯ**), построение регрессионных зависимостей с различными видами аппроксимации. Эта операция выполняется после выделения необходимых точек диаграммы и использования диалога **Диаграмма/Добавить линию тренда**, где могут быть выбраны линейное, степенное и другие виды приближений.

Система Mathcad

Эффективность, а часто и возможность решения задачи определяются теми методами, которыми она решается. Если задача сформулирована математически, то методы ее решения чаще всего численные, кроме случаев, когда ее можно решить аналитически (в общем виде). Инженер-исследователь должен иметь представление о различных численных методах решения поставленной задачи. В современных универсальных математических программах предлагается несколько различных методов решения некоторых классов задач, и пользователь должен обоснованно осуществить свой выбор.

Одна из особенностей научной деятельности состоит в сочетании знаний в специализированной области со знаниями в области информационных технологий. Программа *Mathcad* является средством автоматизированного решения как инженерных, так и исследовательских задач, и может быть использована в качестве универсального инструмента при подготовке инженеров на всем периоде обучения.

Характеристика программы *Mathcad*

Mathcad – программное средство для выполнения на компьютере разнообразных математических и технических расчетов, предоставляющая пользователю инструменты для работы с формулами, числами, графиками и текстами, снабженная простым в освоении графическим интерфейсом.

Графический интерфейс – это совокупность средств и способов взаимодействия пользователя с программой с помощью пиктограмм, диалоговых окон, меню и других элементов, располагаемых на экране монитора.

В основу программы *Mathcad* положена идеология *рабочего документа*, представляющего собой и описание решаемой задачи, и одновременно протокол ее решения. Любые изменения в рабочем документе приводят к пересчету решения задачи, что позволяет легко организовать *многовариантный анализ* и проведение так называемых *численных экспериментов*.

В *Mathcad* встроено большое число операторов и функций, предназначенных для численного и символьного решений инженерно-технических проблем различной сложности. *Mathcad* – мощное гибкое и простое средство проведения инженерных расчетов.

На базе программы *Mathcad* разработано большое число так называемых *электронных учебников (книг)* по различным направлениям науки и техники, которые могут использоваться для обучения. Их фрагменты также могут быть вставлены в рабочие документы *Mathcad* для решения конкретных задач.

Записав с помощью средств *Mathcad* в привычной форме математическое выражение, можно затем выполнить с ним различные символьные или численные математические операции: вычислить значение, выполнить алгебраические преобразования, решить уравнение, продифференцировать, и пр. Можно снабдить вычисления текстовыми комментариями, иллюстрациями, в том числе построенными в других приложениях *MS Windows*.

Для использования пакетов программ решения инженерных задач, к которым относится программа *Mathcad*, задача должна быть сформулирована математически, т.е. представлена формулами, уравнениями, неравенствами и т.п. Далее под математической понимается любая задача, алгоритм которой может быть описан в терминах того или иного раздела математики.

Описание задачи составляется на обычном математическом языке, с учетом встроенных в *Mathcad* возможностей.

Рабочий документ программы *Mathcad*

После постановки задачи и ее описания на языке математики главной проблемой является ввод ее в рабочий документ в рамках программы *Mathcad*. Знание основных правил ввода данных, включая клавишные команды, позволяет достаточно быстро заполнить документ *Mathcad* и получить решение задачи.

Mathcad, как и любое приложение *MS Windows*, обрабатывает рабочий документ, окно которого открывается внутри окна приложения.

Строка заголовка содержит стандартные управляющие кнопки, название программы и имя рабочего документа (файла).

Главное меню (вторая строка окна) позволяет использовать все возможности программы *Mathcad*, вызывая необходимые команды или открывая соответствующие диалоговые окна. Многие из наиболее употребительных команд меню дублируются пиктограммами (кнопками) на *панели инструментов* и/или клавишными командами.

Ввод данных обеспечивается и существенно облегчается использованием *наборных панелей (палитр)*, которые вызываются нажатием на кнопки вывода палитр (*Palette*) или с помощью главного меню (команда *Просмотр – Панели*). Можно вывести все палитры одновременно. Положение палитр на экране произвольное; они могут перемещаться по экрану обычным перетаскиванием после зацепления курсором мыши верхней строки палитры.

В новейших версиях *Mathcad* палитры заменили или дополнили многие команды главного меню, кнопки панелей инструментов и клавишные команды.

Наборные поля (палитры) позволяют ввести в документ арифметические операторы и некоторые функции, операторы отношения, операторы графики, матричные операторы, шаблоны математических объектов (производная, интеграл, сумма и пр.), операторы программ, буквы греческого алфавита. Для ввода широко используются шаблоны с *полями ввода*, заполнение которых позволяет описать задачу в привычном (естественном) математическом представлении.

Ввод информации (описание задачи) производится в любом месте рабочего документа, которое указывается щелчком мыши. Место ввода в рабочем документе отмечается курсором (визиром), который имеет вид красного крестика.

В результате ввода информации пространство рабочего документа разделяется на *прямоугольные области* вокруг каждого объекта (каждой отдельной конструкции) – формулы, графика, текста и других, которые могут перекрываться.

Mathcad читает рабочий документ *сверху вниз* и *слева направо*. Это накладывает жесткие ограничения на расположение областей в документе.

Просмотр документов выполняется с помощью полос прокрутки либо с помощью клавишных команд.

Создание нового документа производится при запуске программы *Mathcad* или командой *Файл (File) – Создать (New)*. Новому документу по умолчанию присваивается имя *Untitled* с номером.

Сохранение документа в виде файла на диске выполняется как обычно с помощью команд *Файл (File) – Сохранить (Save)* (под присвоенным ранее именем) или *Файл (File) – Сохранить как (Save as...)*, что позволяет присвоить документу новое имя и выбрать папку, в которую помещается документ.

Открытие ранее созданного документа выполняется с помощью стандартного диалогового окна *Открыть (Open)* после вызова команды *Файл (File) – Открыть (Open...)*.

При открытых одновременно нескольких документах переход от видимого окна к невидимым осуществляется с помощью команды *Окна (Window)* главного меню. Окна открытых документов могут быть видимыми одновременно после выполнения команд *Каскадом (Cascade)*; *По горизонтали – По вертикали (The Horizontal – The Vertical)* из меню «Окна» (*Window*).

Ввод и редактирование формул

Формулы представляют собой операнды, соединенные между собой знаками операторов, в том числе знаками арифметических операций. В качестве операндов в формулах могут выступать константы, переменные, переменные с индексами, встроенные функции и функции пользователя и более сложные конструкции, такие как сумма, произведение, интеграл, производная, матрица.

Из формул, связанных знаками равенства и неравенства, образуются уравнения и неравенства.

Формулы или текст вводятся в любое место рабочего документа. Каждая введенная математическая конструкция, текст, диаграмма являются

областью. Место ввода указывается щелчком мыши вне существующих областей рабочего документа *Mathcad*.

Знаки операций (операторы) вводятся либо с помощью клавишных команд, либо с помощью кнопок соответствующей палитры.

При вызове шаблонов математических конструкций, например с помощью палитр, на экране появляются *поля ввода*. Поле ввода в виде маленького черного квадрата представляет собой место для ввода соответствующих операндов. Поле ввода активизируется щелчком мыши.

Mathcad не переносит на новую строку «длинную» формулу при достижении правой границы страницы.

При вводе и редактировании формул важно убедиться в правильной последовательности действий при вычислениях. *Mathcad* во многом помогает пользователю, предоставляя ему при вводе различные шаблоны.

Основным моментом при вводе и редактировании формул является выделение всего выражения или его части выделяющей рамкой в виде *углового маркера (маркера ввода)*. То, что заключено в выделяющую рамку, становится операндом следующего вводимого оператора.

При вводе формул новые символы вставляются слева или справа от позиции формульного курсора. Направление ввода определяется либо срезом на выделяющей рамке (в предыдущих версиях *Mathcad*), либо углом маркера ввода: новые символы вводятся в направлении угла.

Изменение направления угла маркера, перемещение его по формуле для последующего редактирования формулы, изменение размеров обрамляющей рамки производится с помощью клавиш со стрелками или клавиши пробела. Чтобы выйти за пределы напечатанного операнда, нужно нажать клавишу пробела. Управление выделением фрагментов выражений становится интуитивно понятным уже после небольшого практического опыта. За положением и направлением углового маркера при вводе нужно внимательно следить.

Вводимые символы распознаются как составляющие одного операнда (в операции деления, при вводе подкоренного выражения, показателя степени, индексов), пока не нажата клавиша пробела.

Греческие буквы (например символ « π ») вводятся с помощью кнопок соответствующей палитры (*Greek Symbol Palette*).

При вводе *чисел* для отделения дробной части от целой используется *точка*. Для ввода числа в экспоненциальной форме (с плавающей точкой)

используется знак операции возведения в степень числа 10, например $3 \cdot 10^8$. Следует отметить, что такая запись воспринимается как число, но не формула.

При редактировании формул часто бывает необходимо заключить часть выражения в круглые скобки. Для этого при выделенном фрагменте формулы можно нажать на клавишу с одиночной кавычкой (апостроф <>). Естественно, можно вставлять открывающую или закрывающую скобки клавишами по одной в нужных местах, отмеченных маркером ввода. Маркер ввода перемещается по выражению клавишами со стрелками или помещается в нужное место выражения щелчком мыши.

Для удаления, копирования или перемещения фрагментов выражения можно использовать обычное выделение с помощью мыши (при нажатой левой клавише) и буфер обмена *Windows*.

Вычисление выражений

Для вычисления значения выражения (формулы) вводится само выражение, содержащее константы (числа), и ранее определенные переменные и функции, в том числе встроенные в *Mathcad* функции, и затем вводится знак равенства $\langle = \rangle$ (команда *Вычислить*). *Mathcad* вычисляет значение выражения и показывает результат после знака $\langle = \rangle$.

Использование переменных

Переменная представляет собой *имя переменной*, которой может быть присвоено какое-либо значение в соответствии с правилами *Mathcad*. При выборе имен переменных в *Mathcad* действуют следующие правила (ограничения):

- Имена переменных и функций не могут содержать пробелы.
- Имя не может начинаться с цифры, знака подчеркивания, символа процента.
- Все символы в имени должны быть введены *шрифтом одной гарнитурой*, размера и начертания; иначе говоря, одинаковые имена, введенные различными шрифтами, определяют различные переменные.
- Имена переменных и функций не должны совпадать.
- Некоторые имена уже используются *Mathcad* для встроенных констант, единиц измерения и встроенных функций; использование совпадающих имен приводит к путанице.
- *Mathcad* различает в именах символы верхнего и нижнего регистра.

Любые символы в имени, напечатанные после нажатия клавиши <точки>, будут представлены как *нижний буквенный индекс*, который является *частью имени переменной*.

Греческие буквы вводятся с помощью соответствующей палитры. Символ «л» можно ввести комбинацией клавиш <Ctrl>+<P>.

Нижние индексы в описаниях *переменных с индексами* (элементов массивов) создаются клавишей левой прямоугольной скобки <[>, если не используются соответствующие шаблоны. Выход из режима описания нижних индексов выполняется клавишей пробела.

Верхние индексы (для указания столбцов массивов) вводятся с помощью команды <Ctrl>+<6>.

Для *локального определения* переменной необходимо ввести ее имя, напечатать символ <:> (двоеточие), который вводит в документ *знак определения* <:=>, и ввести значение переменной в виде выражения (формулы). Частным случаем выражения является число. При первом описании переменной можно использовать просто знак равенства <=>.

Слева от знака определения <:=> могут стоять:

- имя простой переменной;
- имя переменной с нижними индексами;
- имя переменной с верхними индексами;
- имя функции со списком аргументов, состоящим из имен простых переменных.

Определение локальной переменной задает значение ее *всюду ниже* места расположения определения.

Значение переменной может быть переопределено несколько раз в рабочем документе. В вычислениях используется *последнее значение переменной*, если просматривать документ сверху вниз и слева направо.

Глобальные определения вычисляются прежде любых локальных определений и доступны всем локальным определениям в рабочем документе, независимо от того, где расположены глобальные определения. Глобальное определение задается именем переменной, знаком «тильда» <~>, выводимым в документе как символ <≡>, и выражением. Если в глобальном определении справа от знака <≡> используется переменная, то эта переменная должна быть определена выше (ранее в документе) и тоже с помощью знака <≡>.

Функции пользователя

Определение функции (локальное) в рабочем документе *Mathcad* аналогично определению переменной. **Функция пользователя** определяется так: вводится выбранное имя функции, в круглых скобках – список аргументов (формальных параметров) через запятую, знак определения $\langle : \rangle$ и выражение (формула), определяющее функциональную зависимость.

Выбор *имени функции* подчиняется тем же правилам, что и выбор имени переменной. В частности, имена функций чувствительны к шрифту: функции с одинаковыми именами, описанные различными шрифтами, являются различными функциями. В то же время встроенные в *Mathcad* функции одинаково определены для всех шрифтов (кроме шрифта *Symbol*).

Переменные в списке аргументов в определении функции являются *формальными параметрами* (формальными переменными). Поэтому не имеет значения, были ли ранее определены или использованы в рабочем документе имена из списка аргументов. Выражение справа от знака определения содержит имена из списка аргументов, а также, возможно, другие ранее определенные переменные и функции.

При использовании описанной в документе функции пользователя формальные переменные заменяются фактическими, которые могут представлять собой выражения. *Фактическими аргументами* функции могут быть скаляры, векторы, матрицы.

Если в описании функции используется переменная, которой нет в списке аргументов (формальных параметров), то она должна быть определена ранее в документе. Важно иметь в виду, что при вычислении значения функции используется *то значение переменной, которое она принимала при определении функции*.

Предопределенные переменные

Mathcad содержит переменные, значения которых определены сразу после запуска программы. Предопределенные переменные определены для всех гарнитур, размеров и начертаний символов в их именах. Значения переменных *TOL*, *ORIGIN*, *PRNPRECISION*, *PRNCOLWIDTH* могут быть установлены в режиме *Математика (Math)* – *Встроенные переменные (Built-in Variables...)* или с помощью оператора глобального присваивания, вводимого клавишей $\langle \sim \rangle$. Значения этих переменных могут быть определены в любом месте рабочего документа.

Управление вычислениями

По умолчанию *Mathcad* обновляет результаты в окне рабочего документа автоматически после любого изменения в документе (*Автоматический режим*). Автоматический режим можно отключить или включить командой *Математика (Math) – Автоматический режим (Automatic Calculation)*.

Автоматическое обновление рабочего документа производится в рамках видимой на экране части документа. Перемещение по документу приводит к обновлению других частей документа. Обновление всего документа выполняется при обращении к печати документа или при перемещении в конец документа.

Можно заставить *Mathcad* обновить весь документ (пересчитать все значения), вызвав команду *Математика (Math) – Пересчитать все (Calculate Worksheet)*. Ручное обновление *видимого окна* вызывается командой *Математика (Math) – Пересчитать (Calculate)* либо командой <F9>, либо щелчком по кнопке со знаком равенства на панели инструментов. Следует помнить, что *Mathcad* обрабатывает документ, просматривая его сверху вниз и слева направо.

Прерывание вычислений вызывается нажатием клавиши <Esc>. В открывшемся окне нужно щелкнуть по кнопке <Ок> для остановки процесса либо по кнопке <Отмена> для его продолжения. Выражение, которое обрабатывалось в момент нажатия клавиши <Esc>, отмечается соответствующим сообщением. Чтобы продолжить вычисления, нужно щелкнуть по отмеченному выражению мышью либо выбрать команду *Математика (Math) – Пересчитать (Calculate)*, либо нажать клавишу <F9>, либо щелкнуть по кнопке со знаком равенства на панели инструментов.

Создание и редактирование массивов

Скаляр – это одиночное число; вектор – столбец или строка чисел или выражений; матрица – прямоугольная таблица чисел или выражений; массив – вектор или матрица.

Массив создается путем заполнения пустых полей шаблона или использования дискретного аргумента (ранжированной переменной), или считывания данных из файлов.

Шаблон массива в нужное место документа вставляется командой <Ctrl>+<M> либо с помощью палитры *Вектора и матрицы (Vectors and Matrix Palette)* или командой *Вставка (Insert) – Матрицы (Matrix)*. При

этом заполняются поля диалогового окна (число строк и столбцов) и нажимается кнопка *<Вставить (Insert)>*.

Шаблон заполняется значениями элементов массива последовательно: после щелчка мышью по одному из полей шаблона вводят соответствующее значение. По полям шаблона можно перемещаться с помощью клавиш со стрелками.

Как только вектор или матрица созданы, их можно использовать в формулах и вычислениях как единый элемент, естественно, с соблюдением правил выполнения матричных операций.

Размер существующей матрицы можно изменить, вставляя или удаляя строки и столбцы. Щелкнув мышью по одному из элементов матрицы, нажимается кнопка *<Матрица или вектор>* в палитре *Вектора и матрицы*, и в окне «*Матрицы*» указывается число строк и/или столбцов, которое необходимо вставить или удалить. Затем нужно щелкнуть по кнопке *<Вставить (Insert)>* или *<Удалить (Delete)>*.

Строки пустых полей вставляются *ниже* выбранного элемента. Столбцы пустых полей вставляются *справа* от выбранного элемента.

Строки или столбцы *удаляются*, начиная со строки или столбца, занятого выбранным элементом. Удаляются строки *вниз*, а столбцы – *вправо* от выбранного элемента.

Удалить всю матрицу можно, выделив ее выделяющей рамкой и вводя команду ** или команду *Правка (Edit) – Вырезать (Cut)*.

Переменные могут являться массивами (векторами или матрицами). Правила выбора имен векторов и матриц обычные. Однако полезно выбирать для имен векторов и матриц какой-либо особый шрифт – заглавные (прописные) буквы, «жирный» шрифт или курсив. Не следует использовать одинаковое имя для скалярной и векторной переменных. Это приведет просто к переопределению переменной и, следовательно, к путанице.

Переменная в виде массива определяется как обычно: вводится имя переменной, знак определения (клавиша *<:>*), шаблон матрицы, который затем заполняется значениями элементов.

Элементы массива (вектора или матрицы) обычно нумеруются, *начиная с нулевой строки и нулевого столбца*. Чтобы изменить этот порядок, нужно изменить значение встроенной (предопределенной) переменной *ORIGIN* с помощью команды *Математика (Math) – Опции (Option) – Встроенные переменные (Built-in Variables...)* или описанием в любом месте рабочего документа глобального значения, например *ORIGIN = 1*.

К отдельным элементам вектора или матрицы обращаются с помощью *переменных с индексами* (нижними). К отдельному столбцу массива обращаются с помощью *переменной с верхним индексом*.

Чтобы напечатать *нижний индекс*, используется клавиша левой прямоугольной скобки <[>. В поле индекса вводится (в виде выражения) значение одного или двух (номер строки и номер столбца через запятую) индексов. Выход из поля индексов – клавиша пробела < Space >.

Чтобы вставить *верхний индекс*, используется команда <Ctrl>+<6>. Верхний индекс, определяющий номер столбца матрицы, выводится в документе в угловых скобках. Выход из поля верхнего индекса – клавиша пробела.

Значение переменной с индексами можно определять или переопределять как любую переменную. Переменная с индексами может использоваться везде наравне с обычной переменной. Переменные с индексами могут применяться для вывода значений соответствующих элементов массива.

Если ранее массив не был создан, то попытка определить переменную с индексами приводит к созданию массива с размерностью по значениям использованных индексов. Неопределенные элементы массива заполняются нулями.

Отображение векторов и матриц

Поскольку в рабочем документе *Mathcad* могут использоваться векторы и матрицы больших размерностей, для их отображения в документе автоматически, если число строк и/или столбцов больше 9, используются *таблицы вывода* с номерами строк и столбцов и с полосами прокрутки (если число строк и/или столбцов достигает 14). Размер таблицы вывода можно изменить растягивая ее контуры мышью вниз или вправо.

Можно изменить способ отображения массивов, отказавшись от использования таблиц вывода. Для этого после щелчка мышью по таблице вывода выбирается режим *Формат (Format) – Числовой формат (Number...)*, в окне которого активизируется поле *Вывести как матрицу (Display as Matrix)*.

Ограничения размеров массивов. Нельзя с помощью шаблона создать или увеличить массив, имеющий более чем 100 элементов. Массив большей размерности можно получить соединением нескольких массивов, либо используя дискретный аргумент, либо путем ввода из файла на диске.

Полный объем используемых массивов ограничивается объемом памяти компьютера, но не может превышать 8 млн элементов.

Операции с матрицами

Над матрицами выполняются операции транспонирования, сложения (вычитания), умножения, вычисления обратной матрицы.

Операция транспонирования матрицы A определяется как A^m . Знак транспонирования вводится оператором <Ctrl>+<I>.

В качестве операторов сложения, вычитания и умножения матриц используются обычные знаки арифметических операций. При этом матрицы, выступающие как операнды, должны иметь соответственные размерности.

Операция умножения двух векторов дает скалярное произведение. Векторное произведение двух векторов определяется знаком «крестик» из палитры символов или нажатием клавишей <Ctrl>+<8>.

Операция **обращения квадратной матрицы** задается как возведение этой матрицы в степень -1 . Определитель матрицы или длина вектора вычисляются, если соответствующее имя заключено в рамки символом <|>.

Дискретные аргументы

Дискретный аргумент (ранжированная переменная) принимает ряд значений, отделяемых *одинаковым шагом* (арифметическая прогрессия).

Дискретная переменная (аргумент) определяется следующим образом: вводится имя переменной, знак определения <:=>, начальное значение, через запятую – *второе значение* из набора дискретных значений, символ <.>, отображаемый в документе как многоточие <..>, и конечное значение.

Если шаг изменения дискретной переменной равен единице, то второй элемент в описании может быть опущен (не указываться).

В качестве начального и конечного значений дискретного аргумента и величины шага могут использоваться любые скалярные выражения (с вещественными значениями). Величина шага может быть как положительной, так и отрицательной.

Если дискретный элемент используется в выражении, то *Mathcad* вычисляет это выражение столько раз, сколько значений принимает аргумент, один раз для каждого значения аргумента.

Нельзя определять простую переменную через дискретный аргумент. Если дискретный элемент принимает дробные значения, то он не может использоваться в качестве нижнего индекса у переменной с индексами.

Дискретный аргумент может применяться для присваивания значений элементам вектора или матрицы. Использование двух дискретных аргументов и двойных нижних индексов позволяет обрабатывать матрицы.

С помощью дискретных аргументов могут быть организованы многократные (циклические) и рекурсивные вычислительные процедуры. В рекурсивных вычислениях определяется первый элемент массива и затем каждый следующий элемент определяется через предыдущий.

Дискретные переменные можно использовать для вывода результатов. В частности, для построения таблицы значений функции (табулирования) указывается имя функции со списком аргументов в круглых скобках и знак равенства $\langle = \rangle$, если ранее аргументы определены как дискретные переменные.

Если для вывода вектора использовать имя вектора с нижним индексом – дискретным аргументом, то будет показана таблица вывода.

Если напечатать имя вектора без нижнего индекса, то будет представлена обычная форма вектора, но если вектор имеет более девяти элементов, то появится таблица вывода с номерами строк и столбцов (при необходимости и с полосой прокрутки).

Дискретные переменные можно использовать для организации ввода данных в форме таблицы ввода. Для ввода числовых значений в таблицу можно задать дискретный аргумент, принимающий только целые значения, и затем определить переменную с индексом в виде этого дискретного аргумента. В поле ввода справа от знака определения $\langle := \rangle$ вводится значение первого элемента таблицы и затем *запятая*. В появившееся новое поле ввода вводится второй элемент матрицы и т.д. Таким же образом можно вставить значение в середину таблицы, расширить таблицу под дополнительные значения, заменить или удалить значение из таблицы.

Каждое значение в таблице ввода должно быть или числом, или выражением, которое возвращает число, или именем массива, либо выражением, возвращающим массив. Таблицы ввода присваивают значения только элементам, определяемым дискретным аргументом. Неопределенным элементам приписываются нулевые значения. Таблицы ввода не могут иметь более 50 ячеек.

Построение простых графиков

В *Mathcad* встроены эффективные средства отображения графических зависимостей.

Для построения графика создается графическая область с помощью палитры графиков (*Graph Palette*) или команды *Вставка (Insert) – График (Graph)*.

Шаблон для построения двумерного графика в прямоугольной системе координат вызывается нажатием кнопки *<X-Y Plot>* палитры или выбором соответствующей строки в меню.

В поле ввода под осью абсцисс вводится выражение, содержащее дискретную переменную. В поле ввода около оси ординат вводится функция или выражение, содержащее дискретную переменную. График отображается после щелчка мышью вне графической области. Под функцией изображается образец линии, который использован для построения графика (легенда).

Для построения нескольких графиков на одной плоскости в поле ввода после первой функции ставится *запятая* и вводится вторая функциональная зависимость и т. д.

Форматирование графиков (выбор типа линий, делений на осях, сетка, цвета, надписи) выполняется с помощью соответствующих установок в диалоговом окне, которое открывается после двойного щелчка внутри графической области.

Размер графической области изменяется путем перетаскивания темных квадратиков на границах области.

Вставка текста в рабочий документ

В рабочем документе *Mathcad* могут быть созданы текстовые области.

Ввод текста. Для этого следует, поместив курсор документа (визир) в нужное место, выполнить команду *Вставка (Insert) – Текстовая область (Text Region)* либо ввести символ *<">* (двойную кавычку). Вообще говоря, ввод текста может начинаться с любой буквы непосредственно в месте нахождения визира.

Текстовая область при вводе ограничена рамкой, которая расширяется по мере ввода и исчезает после выхода из текстовой области. Выход из текстовой области производится щелчком мыши в любом месте рабочего документа. Текстовая область не смещает в рабочем документе другие области и может их перекрывать. В текстовую область можно вставить уравнения (формулы, выражения). Для этого после помещения курсора в нужное место вызывают команду *Вставка (Insert) – Математическая область (Math Region)*. Уравнения, вставленные в текст, имеют те же свойства, что и уравнения в других областях документа. Они редактируются обычными методами и будут пересчитываться вместе с другими уравнениями.

Ввод выражений в текст может изменить интервал между строками в тексте. Добавление выражения в текст осуществляется обычными технологиями копирования и вставки.

В текст могут быть введены символы *греческого алфавита*. Для этого используют палитру греческих символов.

Корректировка текста. После щелчка мышью внутри текстовой области появляется текстовая рамка вокруг текста и текстовый курсор. Для *удаления символов* слева или справа от текстового курсора используются клавиши <Backspace> и соответственно.

Замена текста выполняется при печати нового текста *в режиме замены* после нажатия клавиши <Ins>. Для возврата в режим вставки следует повторно нажать клавишу <Ins>. Начало нового абзаца – клавиша ввода <Enter> («жесткий» перенос строки, который сохраняется при переформатировании текста). «Мягкий» переход на новую строку – команда <Shift>+<Enter>. Удаление фрагмента текста – клавиша после выделения этого фрагмента.

Для **редактирования фрагмента** текста внутри текстовой области или параграфа его необходимо выделить обычными приемами с помощью мыши. Когда фрагмент текста выделен, его можно удалить, скопировать, переместить, изменить стиль, размер, цвет шрифта, проверить орфографию.

Изменение шрифта выделенного фрагмента выполняется командой *Формат (Format) – Текст (Text...)* и соответствующими установками в диалоговом окне. Можно также использовать поле шрифтов в панели инструментов.

Изменение ширины текстовой области. Правая граница текстовой области устанавливается в процессе ввода текста в нужном месте нажатием клавиши пробела < Space > и затем <Ctrl>+<Enter>. Изменение ширины существующей текстовой области после установки правой границы указанным способом производится путем перетаскивания правой границы выделяющего текстовую область прямоугольника.

Расчет по формулам

Формула, числовое значение которой необходимо вычислить, может содержать числа и функции с числовыми (вычисляемыми) аргументами. Элементы формулы (операнды) объединяются знаками арифметических операций.

В программе *Mathcad*, если значение формулы не используется далее в рабочем документе, можно не вводить имя для вводимой формулы; результат вычислений помещается справа от формулы после ввода команды $\langle \Rightarrow \rangle$ («вычислить и вывести»).

При записи (при вводе) формулы главной проблемой является сохранение соответствия между исходной формулой и формулой, записанной по правилам той или иной программы, т.е. сохранение правильной последовательности действий при вычислениях. Управляют последовательностью действий с помощью круглых скобок, поскольку по правилу приоритета элементы формул, заключенные в скобки, вычисляются прежде всего. Типовой ошибкой при записи формул является неправильная расстановка скобок, в частности, несоответствие числа открывающихся и закрывающихся скобок (непарные скобки).

Достоинство программы *Mathcad*, которое облегчает ввод формул при соблюдении правильной последовательности вычислений, состоит в том, что отображение формулы в рабочем документе *Mathcad* соответствует обычной математической записи формул.

Расчеты по формулам являются многоступенчатыми, когда в формулах используются значения результатов расчета по предыдущим (другим) формулам. В этом случае промежуточным формулам присваиваются имена (обозначения). Если в громоздкой формуле встречаются повторяющиеся фрагменты, то каждый такой фрагмент целесообразно обозначить именем и вычислить лишь один раз. Таким образом, при многоступенчатых вычислениях в формуле могут присутствовать кроме чисел и функций еще и переменные (их имена), обозначающие соответствующие числовые значения.

В программе *Mathcad* присваивание формуле имени имеет вид:

$$\langle \text{переменная} \rangle := \langle \text{выражение} \rangle,$$

где $\langle \text{выражение} \rangle$ представляет собой формулу, которая в свою очередь, возможно, содержит какие-либо имена переменных.

Анализ функциональных зависимостей

Программа *Mathcad* позволяет легко построить таблицу и график функции. Исследуемая функция $f(x)$ описывается в рабочем документе *Mathcad* по обычным правилам. При этом формула, отображающая функцию, представляется в рабочем документе в обычном математическом виде. Значения аргумента задаются в виде описания дискретной переменной, например $x:=1,1.1..10$, что означает изменение аргумента от 1 до 10 с ша-

гом 0,1. Команда $f(x)=$ выводит таблицу значений функции. Для получения графика во вставленной в рабочий документ области графика указываются в соответствующих полях дискретный аргумент x и функция $f(x)$. Масштабы по осям выбираются автоматически. Размеры графика и его формат легко изменяются путем перетаскивания границ области графика и вызовом контекстного меню щелчком правой кнопки мыши внутри области графика.

Решение уравнений

Решить уравнение – найти такое значение аргумента, которое превращает уравнение в тождество. Уравнение чаще всего задается в виде $f(x) = 0$. Известные численные методы, реализованные в математических программах, отыскивают значение одного корня функции $f(x)$. Алгоритмы численных методов представляют собой итерационные процедуры – процедуры последовательного приближения к искомому решению, начиная от некоторого начального значения. Это *начальное приближение* в программе *Mathcad* задается явно при описании задачи.

Если функция $f(x)$ монотонна и имеет один корень, то численное решение уравнения обычно сходится к искомому корню. Если функция имеет несколько корней, то поиск нужного корня возможен только после выполнения процедуры *выделения корней*, т.е. выявления таких интервалов, на каждом из которых находится по одному корню. Для выделения корней необходим предварительный анализ функции методами, описанными выше. В программе *Mathcad* для решения уравнения используется встроенная функция **root**. После обычного описания функции $f(x)$ и задания начального приближения функция **root** ($f(x), x$) непосредственно возвращает значение корня.

Решение систем уравнений

В программе *Mathcad* для решения таких задач имеются функции **Find** и **Minerr**. Для описания задачи (системы уравнений) создается блок, который начинается служебным словом *Given* (*Дано*) и заканчивается строкой, где присутствует обращение к указанным функциям. Уравнения вводятся как обычно, однако вместо «обычного» знака равенства вводится «жирный» (логический) знак равенства. До блока *Given* вводятся начальные приближения. Функции **Find** и **Minerr** возвращают векторные значения. В качестве аргументов функций перечисляются неизвестные величины из системы уравнений (через запятую).

Решение систем линейных уравнений

Систему линейных алгебраических уравнений в программе *Mathcad*, во-первых, можно решить так же, как нелинейную систему с помощью функций *Find* или *Minerr*, создав блок *Given*. Во-вторых, решение системы выполняется просто и наглядно в матричной форме, поскольку задача в рабочем документе *Mathcad* описывается в обычной математической форме.

Кроме того, в *Mathcad* имеется встроенная функция *lsolve*, возвращающая вектор решения системы уравнений, записанной в матричной форме.

Определение корней многочленов

Задача нахождения корней многочлена встречается при анализе различных физических и технических систем. В частности, решение линейного дифференциального уравнения n -го порядка (или системы n уравнений) требует решения характеристического уравнения, которое представляет собой многочлен (полином) n -й степени, приравненный нулю. Из алгебры известно, что многочлен n -й степени имеет n корней – действительных и/или комплексных. При действительных коэффициентах полинома комплексные корни могут быть только сопряженными парами.

В программе *Mathcad* имеется соответствующая специальная функция *polyroots*. Аргументом ее является вектор, составленный из коэффициентов полинома, начиная с нулевой степени (со свободного члена).

Вычисление определенных интегралов

В *Mathcad* задание на вычисление определенного интеграла оформляется очень наглядно. Достаточно с помощью соответствующего шаблона описать интеграл в привычной для математики форме и ввести оператор «вычислить и вывести» в виде знака равенства $\langle = \rangle$:

Вычисление производных

В программе *Mathcad* задача вычисления производной в точке описывается как обычно в математике. *Mathcad* позволяет вычислять производные и в аналитическом виде.

Расчет максимальных и минимальных значений функций (задачи оптимизации)

Одной из наиболее типичных инженерных задач является задача поиска экстремальных значений функции. К такой математической задаче сводится задача поиска оптимальных значений параметров какого-либо объекта (устройства, системы) при заданном критерии оптимальности.

В программе *Mathcad* для решения подобных задач может быть использована функция *minerr*, которая минимизирует отклонение функции от заданного значения. В качестве такого значения можно взять число, заведомо меньшее (большее), чем искомое минимальное (максимальное) значение функции. Для использования функции *minerr* необходимо задать в явном виде начальное приближение к искомому значению X и создать блок *Given*:

В *Mathcad* введена специальная функция *minimize* $\{y, x\}$, которая минимизирует функцию $y(x)$ на основе алгоритма, который не требует вычисления производной, что позволяет решать такие задачи, в которых вычисление производных по каким-либо причинам невозможно.

Аппроксимация нелинейных характеристик

В программе *Mathcad* для решения задачи нелинейной аппроксимации имеется функция *genfit*. Неудобством является то, что для ее применения нужно сформировать вектор, содержащий аппроксимирующую функцию и ее частные производные по искомым параметрам. Заметим, что частные производные могут быть определены аналитически средствами самой программы *Mathcad*. Метод наименьших квадратов можно реализовать непосредственно в рабочем документе *Mathcad*.

Решение задач в аналитическом виде

В программу *Mathcad* встроен мощный *символьный процессор*, который позволяет решать многие математические задачи аналитически, не прибегая к численным методам. Аналитическое решение, когда оно возможно, не содержит погрешностей, свойственных численным методам. *Mathcad* позволяет выполнять в аналитическом (символьном) виде преобразования выражений, приведение подобных членов, разложение на множители, алгебраические и матричные преобразования, операции дифференцирования и интегрирования и пр.

Символьные вычисления в *Mathcad* осуществляются с помощью:

- команд меню;
- ключевых слов (команд) символьного процессора;
- оператора символьных преобразований (символьного вывода) $\langle \rightarrow \rangle$.

Первый способ – на основе команд меню – применим по отношению к одному предварительно выделенному выражению. При этом на результат

не оказывают влияние выражения, присутствующие выше в документе *Mathcad* (например описания каких-либо переменных). Результат выполненных символьных преобразований выводится в виде отдельной конструкции, обычно ниже исходного выражения. Например, для упрощения следующего выражения после его ввода в документ *Mathcad* его выделяют и выполняют команду *Symbolics (Символика) – Simplify (Упростить)*.

В меню *Symbolics (Символика)* программы *Mathcad 8* имеются следующие команды символьных преобразований:

Simplify (Упростить) – выполнить арифметические операции, привести подобные члены, сократить дроби, используя для упрощения основные тождества.

Expand (Развернуть, Разложить) – раскрыть скобки, перемножить и привести подобные члены.

Factor (Разложить на множители) – представить, если это возможно, выражение в виде произведения простых сомножителей.

Substitute (Подставить) – заменить в алгебраическом выражении букву или выражение другим выражением.

Convert to partial fraction – разложить рациональную дробь на простейшие дроби.

Символьные операции с помощью указанных команд меню выполняются над предварительно выделенным объектом – выражением, его частью или отдельной переменной. Если программа не может выполнить существенное упрощение выражения, то на следующей строке повторяется введенное (исходное) выражение.

Второй способ выполнения символьных преобразований и решение задач в символьном виде состоит в применении команд палитры *Symbolic (Символика)*. После ввода символьного выражения выбирается необходимая команда среди кнопок палитры. В появляющееся поле ввода нужно ввести имя переменной, относительно которой выполняются действия. В частности, для рассмотренного выше примера после ввода выражения активизируется команда *Expand (Разложить)* палитры *Символика* и в поле ввода вводится имя переменной x , относительно которой производится разложение.

Следует заметить, что при выполнении символьных вычислений вторым способом символьный процессор учитывает (в отличие от использования меню) ранее введенные в рабочий документ конструкции (объекты),

в том числе присвоенные переменным значения. В частности, это свойство позволяет вычислять значения выражений (действительные – *float* или комплексные – *complex*), в которых присутствуют ранее определенные переменные. Такой способ может оказаться более эффективным, чем использование знака равенства – оператора $\langle = \rangle$. Вычисления в режиме *float* (с плавающей точкой) выполняются с высокой точностью. Точностью вывода результатов можно управлять.

С помощью оператора символьных преобразований (символьного вывода) эффективно выполняются, например, матричные операции .

Символьный процессор является мощным инструментом инженера. Вычисления в символьном виде чрезвычайно наглядны. Во многих случаях применение символьного процессора дает более корректные результаты по сравнению с численным аппаратом.

1.3. Оформление результатов научных работ

1.3.1. Оформление результатов в текстовых редакторах

Результаты научных исследований могут быть представлены в виде отчета, доклада, статьи, презентации и т.п., в оформлении которых в настоящее время широко используются средства вычислительной техники. Обычно процесс создания научного документа включает:

1. Подготовку текстовой части, содержащей формулы и спецсимволы.
2. Формирование таблиц и их графическое отображение.
3. Подготовку иллюстраций в виде схем, рисунков, чертежей, графиков, диаграмм.
4. Грамматический и лексический контроль.
5. Импорт рисунков и графических изображений из других систем.
6. Прямой и обратный переводы.
7. Форматирование документа и печать.

Названные операции в основном поддерживаются текстовыми и табличными процессорами общего назначения, системами грамматического контроля, автоматизированного перевода, а также комплексными и интегрированными системами.

Необходимо отметить, что подготовка научных работ, насыщенных математическими, химическими формулами, имеющими несколько уровней, представляет определенные трудности. Проблема решается использо-

ванием специальных редакторов для научных документов, к которым можно отнести: *ChiWriter*, *TCube*, *WordPerfect* и др. Возможно использование для этих целей системы *Mathcad*. Подготовка научных текстов, сильно насыщенных формулами, наиболее эффективна в системе *TEX (ViTEX)*, где набор формул выполняется средствами специального языка.

Для документов с небольшим количеством формул из обычных текстовых редакторов можно использовать *MS Word*, хотя работа в нем достаточно трудоемка, так как обеспечивает только поэлементное конструирование формул. Текстовый редактор *Word* поддерживает процесс создания научных документов следующими средствами:

1. Функция **Вставка/Символ** позволяет использовать в тексте различные символы.
2. Пункт **Формат/Шрифт** обеспечивают соответственно установку верхних и нижних индексов.
3. Редактор формул дает возможность набора формул с символами.
4. Выполнение несложных схем и изображений с помощью функции панели инструментов **Рисование**.
5. Создание и редактирование таблиц.
6. Грамматический контроль: пункт **Сервис/Орфография**.
7. Замена повторяющихся слов на синонимы (пункт **Сервис/Язык/Тезаурус**).

В создании научных документов кроме редакторов научных текстов можно использовать и другие программные средства:

1. Формирование табличной информации целесообразно вести средствами табличных процессоров (*Excel*, *QuattroPro*) с использованием возможностей графического отображения.
2. Для создания сложных графических иллюстраций в научных документах удобнее применять системы деловой графики (*Corel Draw*, *Visio*) и геометрического моделирования (*AutoCAD*, КОМПАС и т.п.).
3. Для представления доклада в виде фильма или слайдов используют систему презентаций *PowerPoint*.

1.3.2. Программа подготовки презентаций *PowerPoint*

Основное назначение программ графических презентаций – это возможность наглядно иллюстрировать свой доклад графиками, диаграммами, схемами и фотографиями, делая акценты в определенных местах. Таким

образом на конференции предоставляется возможность общаться с коллегами на языке зрительных образов:

- вы самостоятельно регулируете объем и вид доклада;
- определяете, на чем следует заострить внимание;
- выбираете порядок появления элементов;
- назначаете темп показа.

В настоящее время персональные компьютеры и ноутбуки без труда выполняют разнообразные мультимедийные задачи. Возможность встроить в презентацию звуковой файл или видеоклип значительно усилит эффект от представляемой информации.

PowerPoint – это программа, входящая в состав *MS Office*. Она предназначена для создания презентаций, используемых в различных сферах профессиональной научной деятельности. Кроме того, она позволяет создавать слайды для кодоскопов, проекторов, персональных компьютеров, работающих в сети (*Internet* и/или *Intranet*) и выводить их на бумагу и пленку.

Как программный инструмент по созданию презентаций **PowerPoint** предоставляет следующие возможности:

- работа с текстом, таблицами, графикой, анимацией, звуком, видео, а также объектами, заимствованными из **Word, Excel, Internet** и т. д.;
- управление процессом презентации (отображение имеющихся слайдов в определенном порядке);
- управление переходами от слайда к слайду.

Любая презентация включает:

- набор слайдов с их параметрами, задаваемыми либо пользователем, либо с помощью мастеров автосодержания;
- набор параметров рабочей области (ее размер, структура, ориентация и т. п.).

Каждый слайд презентации имеет собственные свойства, которые влияют на его отображение во время демонстрации:

- размер;
- разметку (расположение заголовка, рисунков, таблиц, надписей и т.п.);
- шаблон оформления (цветовая схема, фон, шрифты и т. п.);
- эффект перехода, определяющий режим появления и исчезновения слайда (по нажатию кнопки мыши или автоматически через заданный временной интервал, с анимационными или звуковыми эффектами и т. п.).

Отметим, что слайды различаются по назначению (титульный, нетитульный), а также по информации и объектам, размещаемым на слайде (заголовки, подзаголовки, текст, список, таблица, диаграмма, объект, клип).

Набор текста выполняется по правилам, определенным, например для *Word*. Вставка рисунков, диаграмм, схем и т. п. выполняется по правилам, принятым в соответствии с технологией *OLE (Object Linking & Embedding – связывание и встраивание объектов)*, через буфер обмена.

К конкретному слайду (или ко всем слайдам презентации) могут быть применены различные эффекты. Выполнить необходимые действия возможно, воспользовавшись функцией *Смена слайдов*. К основным возможностям относятся: различные эффекты появления и исчезновения слайда; продвижение рисунка, графика, заголовка или текста по слайду; время демонстрации слайда; техника смены слайда либо по щелчку, либо автоматически по времени; звуковое сопровождение (слайда или презентации в целом).

PowerPoint предоставляет пользователю возможность переносить созданную и/или готовую презентацию на другой компьютер за шесть шагов, используя мастер упаковки. Эти этапы, последовательно выполняемые мастером, очевидны и не требуют дополнительных разъяснений.

1.4. Общение с коллегами по научно-исследовательской работе

Электронная почта. Почтовая программа MS Outlook Express

Полученные научные материалы (отчеты, статьи, документация) могут быть отправлены коллегам не только по обычной почте, но и по электронной (через *Internet*). Таким образом, у заинтересованных в научных разработках лиц через короткий промежуток времени есть возможность познакомиться с электронным вариантом вашей работы. Данный прием активно используется при согласовании научно-технической документации, отправке статей в научные журналы, тезисов на конференции. При этом не исключается дублирование обычной почтой наиболее важных документов и информации.

Электронная почта (Electronic Mail) – удобное и надежное средство передачи персональных сообщений с одного компьютера на другой. Для работы с электронной почтой используют специальные почтовые программы, которые соответствующим образом должны быть настроены на компьютере.

Адрес электронной почты имеет следующий формат:

Имя_пользователя@доменное_имя_почтового_сервера.

Например, адрес `mtf@vpti.vladimir.ru` означает, что на почтовом сервере `vpti.vladimir.ru` зарегистрирован пользователь с сетевым именем `mtf`. Знак `@` соответствует английскому «*at*» и на российском компьютерном сленге именуется «собака».

Точно так же, как и традиционное письмо, сообщение *E-mail* содержит:

1. Заголовок (соответствующий по смыслу надписи на конверте – адресу получателя и адресу отправителя, вместе взятым);

2. Тело письма (лист бумаги с текстом, вложенный в обычный конверт). В состав электронного письма можно включить какой-нибудь файл, например, документ *Word* или *Excel*, графический архивный файл и т. п. (это в традиционном толковании напоминает бандероль).

С точки зрения механизма доставки наиболее важен заголовок письма. Кроме адресов *E-mail* получателя и отправителя в заголовок автоматически включаются такие данные, как дата и время отправления, а также некоторая служебная информация о маршруте следования письма, формате кодирования сообщения и т. д. Каждый заголовок содержит еще пару необязательных параметров: тему письма и адреса пользователей, куда отправляются копии письма (т.е. одно и то же письмо можно легко разослать нужному числу адресатов).

Необходимо отметить, что согласно сложившейся традиции в ответных электронных письмах распространено взаимное цитирование. То есть вы, отвечая на какие-либо вопросы вашего коллеги по переписке, вначале приводите его вопросы, а свои ответы размещаете ниже.

Чтобы отличить ваш текст от текста первоначального послания, текст последнего отмечается специальными знаками в начале каждой строки (обычно угловой скобкой `>` или выделяется иным цветом). Почтовые программы позволяют вставить даты и осуществить такое выделение в автоматическом режиме.

Программа ***Outlook*** помогает читать, сортировать, эффективно отслеживать и находить сообщения электронной почты. Вид окон позволяет одновременно выводить на экран большое количество сведений даже на небольших мониторах. Автоматическая группировка электронных сообщений помогает быстро находить сообщения, в какой бы группе они не находились (входящие, исходящие). Программа представляет возможность одновременно перемещать или удалять все сообщения из группы.

Фильтр нежелательной почты помогает отсеять значительную часть ненужных сообщений, получаемых ежедневно. Последние достижения технологии *Microsoft Research* позволяют оценить, является ли письмо нежелательным, по совокупности факторов, включая время отправки и содержимое письма. Отбор производится не просто по адресу отправителя или типу письма, вместо этого выполняется анализ содержимого письма и его структуры, что повышает вероятность распознавания нежелательной почты. Все сообщения, перехваченные фильтром, перемещаются в папку нежелательной почты, откуда их можно впоследствии восстановить или просмотреть. Вы можете также составить список адресов надежных отправителей, письма от которых никогда не будут считаться нежелательными, и список заблокированных отправителей, чтобы блокировать все сообщения с определенных адресов или из определенных доменов.

Созданная в *Outlook* область чтения позволяет читать сообщения электронной почты без необходимости открывать отдельное окно для каждого сообщения. Подобно листу бумаги область чтения ориентирована по вертикали. Это удобнее для глаз и вместе с новым многострочным списком сообщений позволяет видеть почти вдвое больше сообщений на мониторе.

Outlook имеет возможность предупреждать о поступлении новых сообщений электронной почты в папку «Входящие» с помощью небольшого оповещения на рабочем столе даже во время работы с другой программой. Пользователь сам может быстро создать правила хранения сообщений на основе любого полученного сообщения; для этого достаточно выбрать сообщение и нажать кнопку «Создать правило».

2.1. Создание и использование баз данных (СУБД Microsoft Access)

2.1.1. Основы работы с СУБД Microsoft Access

Microsoft Access – это система управления реляционной базой данных, позволяющая формировать данные и решать задачи управления ими. В плане обработки информации *Access* обладает значительно большими возможностями, чем электронная таблица.

Реляционная модель данных является одним из самых простых видов представления информации. Эта модель представляет собой данные, упорядоченные в таблицы, чаще всего двухмерные. Таблицы такого вида называются *отношениями*. Структурированные таким образом данные могут храниться в ЭВМ в виде *баз данных* (БД). База данных может быть определена как совокупность предназначенных для машинной обработки и хранения данных, которые могут использоваться одним или несколькими пользователями.

Реляционная модель описывает, какие данные могут храниться в реляционных базах данных, а также способы манипулирования такими данными. Основная идея реляционной модели состоит в том, что данные должны храниться в таблицах и только в таблицах. Существует много различных систем обработки данных, оперирующих понятием «таблица», например, электронные таблицы *Excel*, таблицы текстового редактора *Word* и т.п. Ячейки электронной таблицы могут хранить разнотипные данные, например, числа, строки текста, формулы, ссылающиеся на другие ячейки. На одном листе электронной таблицы можно разместить несколько совершенно независимых таблиц, если под таблицей понимать прямоугольную область, расчерченную на клеточки и заполненную данными. Таблицы текстовых редакторов вообще могут иметь совершенно произвольную структуру. Электронные таблицы и текстовые редакторы позволяют хранить и обрабатывать данные очень гибко, но как быть, если требуется хранить информацию, например, обо всем технологическом оборудовании большого предприятия, и периодически выдавать ответы на запросы типа «представить список операций, выполняемых на данном фрезерном станке,

для которых используется определенный инструмент, позволяющих обрабатывать деталь с заданной точностью». Для получения ответов на подобные запросы и предназначены системы управления базами данных.

Система управления базами данных (СУБД) – программа, которая управляет данными, осуществляет хранение, извлечение, поиск, редактирование информации, хранимой в базе данных. СУБД также подразделяются на иерархические, сетевые и реляционные в зависимости от данных, которые они обрабатывают. Существует множество различных СУБД: *IMS, Oracle, Clipper, Fox Pro, Access*.

Классическая реляционная модель данных требует, чтобы данные хранились в так называемых плоских таблицах. Более точно, пользователи и приложения, обращающиеся к данным, должны работать с данными так, как если бы они размещались в таких таблицах. В упрощенном виде плоская таблица – это таблица, каждая ячейка которой может быть однозначно идентифицирована указанием строки и столбца таблицы. Кроме того, в одном столбце все ячейки должны содержать данные одного простого типа.

Работе с БД должно предшествовать определение количества, структуры и взаимосвязи таблиц, входящих в БД, состав каждой таблицы и документов, которые необходимо получить по информации БД. Каждая таблица состоит из столбцов, строк. Каждому столбцу присвоено однозначное имя, один элемент столбца называется *полем*, а строка называется *записью*.

Каждому полю, включаемому в запись, приписывается тип данных, определяющий вид информации, которая будет храниться в данном поле. Тип данных вносится в колонку *Data Type*, причем его можно выбрать из списка доступных типов.

Характеристики каждого поля определяются рядом параметров. Эти параметры объявляют способы обработки, сохранения и индикации данных. Параметры специфицируемого поля перечисляются в режиме проектирования в нижней части окна таблицы. Набор характеризующих поле параметров зависит от типа данных, выбранного пользователем для данного поля.

В реляционной базе данных пользователь может описать отношения между несколькими таблицами. *Access* учитывает эти отношения при поиске взаимосвязанных данных во время обработки запросов, формуляров и отчетов, базирующихся на нескольких таблицах.

Запрос можно представить себе как точку зрения на данные, включенные в таблицу. Запросы служат для селекции и фильтрации набора данных. Они позволяют выбрать из базы только необходимую информацию, т.е. ту, которая соответствует определенному критерию (условию) и нужна для решения конкретной задачи. Результат обработки программой *Access* такого запроса представляет собой таблицу, называемую *Dynaset*. В эту таблицу включены выбранные из основной таблицы (или нескольких таблиц) блоки данных, которые удовлетворяют критериям запроса. *Dynaset* – динамический, временный набор данных, поэтому при каждом выполнении запроса он строится вновь на основе «свежих» табличных данных.

Наряду с запросами выбора *Access* реализует также запросы действий, параметрические запросы и запросы кросс-таблиц. С помощью запроса действия пользователь может изменять или переносить данные таблицы, а также актуализировать, добавлять или удалять группы блоков данных, изготавливать новые таблицы из набора *Dynaset*. Различают четыре типа запросов действий: добавления, удаления, актуализации и запрос создания таблицы.

Часто используются запросы, которые представляют собой незначительно видоизмененные варианты однажды подготовленного базового запроса. Речь может идти, например о запросе, который позволяет выбирать из базы данных телефоны клиентов определенного региона, причем сам регион задается отдельно, в диалоге. Параметрические запросы видоизменяются от случая к случаю, но незначительно. Для их реализации проектируется один запрос, в котором указывается критерий (критерии), изменяющийся по заказу пользователя.

Если необходимо объединить данные в формате строк-колонок (двумерная таблица), то следует изготовить запрос *кросс-таблицы*. При проектировании запроса кросс-таблицы можно указать в качестве заголовков для колонок значения некоторых полей или выражений. Запросы кросс-таблиц позволяют более компактно, чем обычные запросы, индексировать данные, объединяя однотипную информацию.

Просмотр базы данных в виде таблицы в режиме заполнения дает пользователю возможность оценить базу как единое целое, сравнить записи и т.п. Часто, однако, возникает необходимость работы с отдельными записями базы. В этом случае присутствие на экране других записей (как это имеет место в режиме заполнения) только мешает и отвлекает. Работа с

отдельными записями посредством формуляров позволяет сосредоточиться только на относящейся к делу информации.

Формуляр представляет собой бланк, подлежащий заполнению, или маску, накладываемую на набор данных. Бланк-формуляр позволяет упростить процесс заполнения базы данных, благодаря чему появляется возможность поручить ввод информации персоналу невысокой квалификации. Маска-формуляр позволяет ограничить объем информации, доступной пользователю, обращающемуся к базе. Речь здесь также может идти о блокировании индикации служебных или засекреченных полей.

Отчет – это информация, которая оформляется в соответствии со спецификациями. Отчет позволяет извлекать и представлять данные как значимую информацию, которую можно использовать и распространять. Примерами отчетов могут служить почтовые адреса, накладные, маршрутные карты. С помощью *Access* можно спроектировать отчет, который представит информацию в желательном виде, включая самые разнообразные элементы проектирования (текст, данные, рисунки, линии, поля и графики). Отчеты также являются эффективным средством для распечатки информации регулярного использования. Создав проект отчета и сохранив его для дальнейшего использования, пользователь получает возможность распечатывать текущие данные.

Access обеспечивает несколько способов получения информации из созданной базы данных, используя запрос, формуляр или отчет. Вы можете выбрать тот метод, который наиболее подходит к решению вашей задачи.

Например:

- для просмотра всех операций определенного станка используйте запрос;
- для просмотра полной информации только об одной операции используйте формуляр;
- для распечатки операций, необходимых для изготовления детали, используйте отчет.

В готовую БД можно вносить изменения. В частности, можно изменять параметры отдельных полей, добавлять поля в запись в нужных местах и удалять лишние. Но при этом следует вносить все исправления в спецификацию до начала заполнения базы данных, так как попытка изменить параметры полей заполненной базы может повлечь за собой потерю или искажение данных.

База данных в *Access* хранится в одном файле с расширением **.mdb**, который содержит все таблицы, запросы, формуляры, отчеты и другие объекты БД. Основные операции следующие:


1. Выполняется вход в систему *Access*. Далее задается имя файла новой БД в диалоговом окне, вызываемом командой **Файл/Создать**. После этого открывается диалоговое окно **База данных**, которое является основным управляющим окном при работе в системе.

2. В *Access* создание таблиц может быть выполнено способом прямого конструирования и с помощью **Мастера таблиц**.

Построение с помощью **Мастера таблиц** выполняется:

- клавишей **Таблица** открывается окно **Создание таблиц**;
- в окне **Создание таблиц** открывается окно **Разработка таблиц**.

Здесь в образцах таблиц выбирается нужная: **Задачи**, **Заказчики** и т.д.

В образцах полей выбираются необходимые, которые будут именами столбцов таблицы. Выбор заканчивается нажатием кнопки  (**Далее**).

В последующих диалоговых окнах **Мастера таблиц** можно задать имя таблицы и другие параметры. В результате выводится таблица с именованными колонками и пустыми клетками, которые заполняются информацией как в электронной таблице.

3. В сформированной таблице или при ее заполнении возможны следующие корректировки:

- изменения в выделенных клетках выполняются поверх имеющихся данных;
- информацию клеток или их групп можно вырезать, копировать, вставлять соответствующими командами меню **Правка** или пиктограммами панели инструментов;
- перемещение выделенных фрагментов таблицы БД;
- вставка новой строки пункта **Правка/Вставка строки**.

Необходимо помнить, что внесение изменений в тип данных и форму таблицы возможны только в режиме конструирования, который включается из пункта **Вид**.

4. Создание форм. Запись БД в *Access* может быть представлена в виде формы, которая содержит названия колонок и данных одной записи.

Формы можно создавать отдельно или с их помощью просматривать и корректировать записи в уже созданных таблицах.

5. Сортировка записей выполняется при открытой таблице, где выделяются колонки для сортировки. Команда на сортировку (по возрастанию или убыванию) осуществляется пиктограммой или через пункт **Записи/Быстрая Сортировка**.

6. СУБД *Access* предоставляет несколько способов поиска: простой, с применением фильтра, запросы.

Простой поиск может быть выполнен скроллингом, но удобнее использовать информацию в форме, где указывается поле поиска, а далее пункт **Правка/Найти**, где в диалоговом окне указываются данные для поиска.

Фильтр – это поиск записей по заданным критериям. Выполняется из окна «Фильтр», открываемого пунктом **Записи/Изменить фильтр**. Нужные поля из описываемого бокса помещаются в таблицу, вводятся критерии. Выполнение фильтрации осуществляется пунктом **Записи/Применить фильтр**.

Запрос – это обращение к БД для поиска или изменения информации, соответствующей нескольким заданным критериям. При этом имеется возможность сохранения формы запроса для его многократного использования.

Вход в режим выполняется из ОБД вкладкой **Запрос/Создать**, далее в окне **Создание Запроса** используется кнопка **Новый Запрос**. Затем выбираются необходимые таблицы и в окне **Запрос – выборка** формируется задание на выборку аналогично построению фильтра. Просмотр результатов осуществляется через пункт **Запрос/Запуск**.

7. *Отчет* представляет собой документ в виде сводки необходимой информации, выбранной из БД. В эту сводку могут быть включены не все, а некоторые из столбцов ранее изготовленной таблицы или запроса. Часто – итоговые данные.

Создание отчета начинается с включения вкладки **Отчет/Создать** в окне ОБД и перехода в окно **Создание отчета**. Далее выбираются из списка необходимые таблицы (запросы) и используются подходящие **Мастера отчетов**. Перемещение по диалоговым окнам **Мастера** позволяет выбрать необходимые поля для отчета, внешний вид и заголовок. По окончании формирования отчета нажатием кнопки **Готово Access** выводит внешний вид отчета для просмотра. После сохранения отчет может быть отпечатан пункт **Файл/Печать**.

2.1.2. Реализация ТехноПро на базе СУБД MS ACCESS

Примером применения СУБД в машиностроении является система автоматизации технологического проектирования ТехноПро. Разнообразие методов автоматизации проектирования, заложенное в ТехноПро и реализованное с помощью программных средств СУБД, обеспечивает проектирование операционной технологии, включая операции: заготовительные, механической и термической обработки, нанесения покрытий, слесарные, технического контроля, сборки и др.

Система ТехноПро реализована на современных СУБД. Уже в первых версиях ТехноПро была заложена возможность работы группы пользователей с единой базой данных на уровне одного или нескольких отделов предприятия. Реализация системы на основе *Access* сделало ТехноПро полноценным приложением *MS Office* и существенно упростило освоение системы за счет применения знакомого всем *Office*-интерфейса и использования *MS Word* для формирования технологической документации. Открытая структура баз *MS Access* позволила многим предприятиям подключить ТехноПро к АСУ ТП предприятия. Новая версия ТехноПро реализована на базе корпоративных СУБД *MS SQL Server* и *Oracle*.

В ТехноПро реализовано множество передовых технологий проектирования процессов на базе СУБД, например:

Диалоговые Сценарии позволяют пользователям создавать и выполнять «на лету», то есть в ходе процесса проектирования, необходимые расчеты, выбор данных из таблиц, поиск и подбор оснащения. Пользователь может выбрать необходимые процедуры из базы «Условий и расчетов» и составить из них Сценарий. Нажатием одной кнопки все процедуры Сценария будут выполнены. Данная технология реализована на основе Запросов СУБД.

Ускоренный подбор оснащения позволяет управлять каждым последующим этапом подбора оснащения в зависимости от выбора на предыдущем этапе. Например, если выбрано наименование операции «Токарная», то при выборе оборудования будет предложено только токарное оборудование, после выбора марки оборудования будут предложены только соответствующие приспособления и далее инструмент. Область выбора можно существенно сужать путем задания диапазонов критериев поиска (параметров оснащения). Можно видеть, что основой этого метода являются Формуляры.

Нормирование. Для некоторых типов технологических процессов можно настроить автоматическое трудовое и материальное нормирование, для других можно использовать диалоговые средства. В том или другом случае система ТехноПро позволяет вводить и затем использовать любые методики нормирования, как уже давно применяемые на предприятии, так и новые, более прогрессивные.

Сборочные ТП. В ТехноПро технологические процессы разделены на два типа: «изготовления» и «сборки». Для проектирования ТП сборки можно использовать данные спецификации. Существенное ускорение формирования текстов сборочных переходов достигнуто применением шаблонов, определяющих расположение информации об элементе спецификации. Для сопряжения с САПР конструирования и системами управления проектами разработан формат обмена данными спецификации. Технологии нормирования и сборки основаны на применении Шаблонов СУБД.

Таким образом, ТехноПро формирует операционные, маршрутно-операционные и маршрутные технологические карты, карты контроля, ведомости оснастки, титульные листы и другие технологические документы. На многих предприятиях технологические карты отличаются от карт, принятых по ГОСТ, поэтому ТехноПро обеспечивает создание технологических документов произвольных форм через использование шаблонов *Microsoft Word*.

Система ТехноПро создана как средство, не подменяющее технолога, но существенно ускоряющее и упрощающее проектирование технологии, расчет режимов и норм, расчет технологических размерных цепей, формирование текстов переходов, выбор необходимой оснастки и инструментов, формирование документации и операционных эскизов. Во многом это стало возможным благодаря тому, что ТехноПро – это система проектирования технологических процессов на основе имеющихся и постоянно пополняющихся баз данных. Так, выбор технологического оснащения производится из информационной базы системы. В ней содержатся каталоги всех составляющих технологических процессов: наименования операций, оборудование, приспособления, вспомогательные материалы, тексты переходов, режущие, измерительные, вспомогательные инструменты, заготовки, комплектующие для сборочных технологических процессов. Любая информация в базе может быть изменена, добавлена или удалена. К каждому типу технологического оснащения в информационной базе можно добавлять параметры, признаки классификации и иллюстрации.

В заключение этого раздела перечислим основные функциональные возможности системы ТехноПро, которые реализованы на технологиях СУБД:

- проектирование операционной технологии, включая операции: заготовительные, механической и термической обработки, нанесения покрытий, слесарные, технического контроля, сборки и другие;
- создание технологических процессов с указанием: наименования операций, оборудования, приспособлений, вспомогательных и дополнительных материалов; режущих, измерительных и вспомогательных инструментов;
 - автоматическое и диалоговое формирование текстов переходов;
 - автоматический и полуавтоматический расчет технологических размеров с учетом припусков на обработку;
 - автоматический и полуавтоматический расчет режимов обработки и норм изготовления;
 - автоматический и диалоговый подбор режущего, измерительного и вспомогательного инструмента;
 - автоматическое формирование операционных, маршрутно-операционных и маршрутных технологических карт, карт контроля, ведомостей оснастки или материалов, титульных листов и других технологических документов;
 - использование оригинальных и традиционных методов проектирования технологических процессов;
 - работа в автоматическом, полуавтоматическом и диалоговом режимах;
 - накопление опыта наиболее квалифицированных специалистов предприятий;
 - использование накопленного опыта при проектировании новых технологических процессов;
 - автоматическое заполнение технологических документов произвольных форм в формате *MS Word*;
 - получение данных о конструкции изделия из параметрических электронных чертежей и трехмерных моделей, выполненных в конструкторских САПР;
 - формирование данных для автоматического перестроения параметрических операционных эскизов с выдачей их в технологические карты.

2.2. Автоматизация инженерных расчетов (Microsoft Excel и Mathcad)

2.2.1. Применение Excel для автоматизации инженерных расчетов

Табличный процессор *Microsoft Excel*, входящий в состав широко распространенного пакета программ *MS Office*, предназначен в основном для обработки табличных данных и работы с несложными базами данных. Однако широкое использование формульных данных, большое число встроенных функций, встроенные возможности решения ряда математических задач, наглядное и простое графическое представление данных, а также возможности программирования на языке *Visual Basic* делают табличный процессор *MS Excel* мощным и эффективным инструментом решения многих вычислительных экономических задач.

MS Excel обрабатывает документ табличной структуры, называемый *Книга*. Книга состоит из *Листов*, количество которых можно менять. Каждый Лист состоит из *Столбцов* и *Строк*, которые пронумерованы: Столбцы нумеруются одной или двумя буквами латинского алфавита, а Строки – натуральными числами. При этом адрес *Ячейки*, расположенной на пересечении столбца и строки, содержит буквенный номер столбца и числовой номер строки без разделителей, например: A1, F5, AB12.

Любой ячейке таблицы может быть присвоено *имя* с помощью команды *Вставка – Имя – Присвоить*. Имя отображается в *Поле имен* над таблицей слева от *Строки формул*. В этом поле имя может быть отредактировано. Вообще говоря, одной и той же ячейке можно присвоить несколько имен. Раскрывающийся список имен содержит все имена, присвоенные ячейкам в данном *Листе*. Выбором нужного имени в этом списке производится быстрый переход в именованную ячейку. Имя ячейки может использоваться вместо ее адреса при ссылках. Аналогично имя может быть присвоено предварительно выделенному блоку таблицы.

Различают относительные и абсолютные адреса ячеек таблицы.

Относительные адреса при использовании их в качестве ссылок в содержимом какой-либо ячейки могут изменяться программой при выполнении таких команд, как копирование и перемещение данных в ячейках.

Абсолютные адреса ячеек при их использовании в качестве ссылок не изменяются при выполнении данных операций. Абсолютные адреса формируются с помощью символа $\langle \$ \rangle$, например: $\$A\1 , $\$F5$, $AB\$12$.

Ссылка на *прямоугольный блок* таблицы образуется из адреса ячейки в *левом верхнем* углу блока и адреса ячейки в *нижнем правом* углу блока, между которыми ставится символ <:>. Заметим, что в адресах ячеек используются только буквы *латинского алфавита* (строчные или прописные – безразлично).

Основной информационной единицей таблицы является *ячейка (cell)*. Текущая (*активная*) ячейка таблицы отмечена табличным курсором – прямоугольной рамкой, который перемещается по таблице клавишами со стрелками или щелчком мыши по нужной ячейке.

В ячейку таблицы вводятся необходимые *данные*. *MS Excel* различает *типы данных*, вводимых в ячейки. Основные типы данных – *текст, числа, формулы*. Тип вводимых данных распознается программой по первому вводимому символу. Ввод текста начинается обычно с буквенного символа. Ввод числовых данных – с цифры или со знака числа. Признаком ввода формулы является символ <=> в первой позиции.

Можно указать в явном виде, что вводимые данные являются текстом, в частности, путем ввода символа <'> в первой позиции, который не отображается в ячейках таблицы.

По окончании ввода данных в ячейку нажимается клавиша <Enter> или клавиши со стрелками. В первом случае курсор таблицы обычно (в соответствии с установкой) перемещается вниз, во втором – по направлению стрелки.

По умолчанию вводимые текстовые данные выравниваются по левому краю ячейки, а числовые данные – по правому краю. Конечно, способ выравнивания данных может быть изменен, например, с помощью кнопок на панели инструментов **Форматирование**.

Текстовые данные, которые не умещаются по ширине ячейки, отображаются в документе полностью, если соседняя (или соседние) ячейка свободна. В противном случае данные «обрезаются» в соответствии с установленной шириной ячейки (столбца). Однако содержимое текущей ячейки полностью сохраняется и видно в **Строке формул** над таблицей, даже если все содержимое не отображается непосредственно в ячейке.

Ширина столбца таблицы изменяется элементарно путем перетаскивания (буксировки) мышью границы между столбцами на адресной строке. Аналогично можно изменить *высоту любой строки*.

При необходимости переход в режим редактирования содержимого текущей ячейки выполняется: 1) клавишей <F2>; 2) однократным щелчком мыши на строке формул или 3) двойным щелчком мыши по соответствующей ячейке. Выход из режима редактирования производится всегда с помощью клавиши <Enter>. При вводе новых данных в текущую ячейку старые данные затираются.

Одношаговые вычисления. Для того чтобы получить числовое значение формулы, содержащей только числа (константы) и встроенные в *MS Excel* функции с числовыми аргументами, необходимо ввести заданную формулу в качестве содержимого одной из ячеек таблицы. Признаком формулы является ввод знака равенства <=> в первую позицию.

Вводимая формула должна быть «вытянута» в строку (как и при программировании на алгоритмических языках) с соблюдением приоритета операций. При вычислениях последовательность действий следующая: при просмотре формулы слева направо сначала вычисляются фрагменты формулы, заключенные в скобки (круглые), затем – значения встроенных функций, выполняются операции возведения в степень, затем равнозначные операции умножения и деления и в последнюю очередь – равнозначные операции сложения и вычитания.

Знаки арифметических операций в *MS Excel* обычные, как принято в программировании: <^> – возведение в степень; <*> – умножение; </> – деление; <+> – сложение; <-> – вычитание.

Числа в формулах могут быть представлены в нормальном и показательном (с «плавающей» точкой) форматах. Обычно в качестве разделителя между целой и дробной частью десятичного числа по умолчанию используется запятая.

Ввод встроенных *Excel*-функций выполняется либо непосредственно с клавиатуры, если имя функции известно, либо с помощью **Мастера функций**. Функция всегда имеет *аргумент*, даже пустой, заключенный в круглые скобки, например, число π вводится функцией ПИ() с пустым аргументом. Между именем функции и скобками, окружающими ее аргумент, не может быть каких-либо символов, в том числе пробела. **Мастер функций** вызывается либо нажатием кнопки на панели инструментов с изображением <f>, либо выбором команды меню **Вставка/Функция**. Работа Мастера функций распадается на два шага.

На первом шаге выбирается нужная функция. Все встроенные функции сгруппированы по их назначению. Нужная группа (категория) выбирается в левом поле окна **Мастер функций/Шаг 1**. В правом поле отыскивается нужная функция из выбранной группы. Для выделенной в списке функции в нижней части диалогового окна показываются образец ее использования и справка, поясняющая смысл функции. После выбора функции и нажатия (щелчком мыши) на кнопку <ОК> открывается окно **Мастер функций/Шаг 2**, которое позволяет легко задать (описать) ее аргументы.

Окончанием ввода заданной формулы в ячейку является нажатие на клавишу ввода <Enter> или на клавишу со стрелкой. Если не допущены ошибки при вводе формулы, то в ячейке непосредственно размещается результат (числовой) вычисления по введенной формуле. Если ячейка с формулой активна, вся введенная формула видна в *строке формул* (непосредственно над таблицей).

При обнаружении ошибок (ячейка активна) для их исправления необходимо перейти в режим редактирования содержимого ячейки одним из способов, указанных выше. Вычисления по отредактированной формуле производятся только после нажатия на клавишу ввода <Enter>.

Возможно, что в результате вычислений в ячейке появится аварийное сообщение, которое указывает на недопустимость операции (например #ДЕЛ0! – деление на 0) или на неправильное использование встроенных функций. В этих случаях после анализа ситуации нужно произвести необходимые изменения в формуле и повторить вычисления.

Многошаговые вычисления. В случаях вычислений по громоздким формулам, в частности содержащим повторяющиеся фрагменты, на практике используют *промежуточные обозначения* для фрагментов. При этом в основной формуле вместо обозначенных (поименованных) фрагментов появляются их имена.

В табличном процессоре в качестве имени фрагмента формулы (для ссылки на соответствующее значение, вместо имени переменной в обычном случае) используются либо адрес, либо имя той ячейки таблицы, в которой располагается фрагмент. Таким образом, при многошаговых вычислениях сначала выбираются ячейки, в которые будут введены фрагменты (т.е. формируется *макет таблицы*), и затем производится ввод этих фрагментов. Визуально в выбранных ячейках видны числовые значения соответствующих фрагментов.

При вводе основной формулы в выбранную ячейку таблицы в ней используются ссылки на занятые фрагментами формулы ячейки в виде адресов ячеек или их имен. Ссылка на блок таблицы (прямоугольный фрагмент) делается путем указания адреса левого верхнего угла блока и через двоеточие правого нижнего угла либо путем указания предварительно присвоенного имени блока. Ссылку на ячейку или блок можно ввести с клавиатуры. Однако более эффективно вводить ссылку в позицию текстового курсора путем выделения мышью нужной ячейки или блока в таблице. Если значение какого-то операнда формулы (фрагмента) находится в известной ячейке, то для ссылки на нужное значение в основной формуле достаточно щелкнуть мышью по этой ячейке, и координата (адрес) этой ячейки будет помещена в формулу.

Если необходимо ввести аргумент функции в окне второго шага **Мастера функций**, то нужно щелкнуть по кнопке с красной стрелкой, направленной вверх, справа от текстового поля для ввода аргумента. В появляющейся на экране строке с текстовым полем будет размещена ссылка на ячейку или блок, который будет выделен с помощью мыши в таблице. Обратный возврат в окно **Мастера функций** выполняется нажатием на кнопку с красной стрелкой, направленной вниз, справа от текстового поля. Строка для ввода может быть перемещена в любое место экрана, если ее «зацепить» мышью за верхнюю (пустую) строку.

После ввода формулы со всеми необходимыми ссылками на другие ячейки (на фрагменты формулы) достаточно нажать на клавишу <Enter>, чтобы получить числовое значение формулы. Если ячейка активна, то введенная формула будет полностью видна в **Строке формул**. Ее можно визуально проверить и при необходимости внести исправления.

Анализ функциональных зависимостей относится к числу наиболее типичных инженерных задач. Функции, описывающие тот или иной физический процесс или физическое явление, обычно весьма сложны, и качественно оценить их характер часто очень трудно. В результате анализа функциональных зависимостей определяют интервалы возрастания и убывания функций, монотонности функций, знакопостоянства, наличие корней функции (пересечений с осью абсцисс), наличие экстремумов (максимальных и/или минимальных значений), точек перегиба, асимптот.

Необходимость в предварительном анализе функциональных зависимостей возникает и при численном анализе, поскольку успех в численном

решении задач часто обусловлен правильным представлением о характере функции. Например, при наличии у функции нескольких максимальных значений (локальных максимумов) только предварительный анализ позволит найти требуемый максимум. При наличии нескольких корней функции найти нужный можно также только после предварительного анализа (после «выделения» корней).

Традиционными методами анализа функциональных зависимостей являются:

- построение таблицы значений функции (табулирование) при изменении аргумента (аргументов) на заданном интервале с постоянным шагом;
- построение графика функции (графическое представление) на заданном интервале.

Математические программы позволяют решить указанные задачи быстро и эффективно. Быстрое получение результатов позволяет провести более глубокий анализ функции, например, на различных интервалах или при различных шагах изменения аргумента. При анализе функции обычно строят таблицу ее значений в равноотстоящих точках по аргументу. Значения аргумента при этом представляют собой арифметическую прогрессию.

Внимательное изучение числовых значений функции в таблице позволяет выяснить, например, примерное расположение корней функции (точек смены знака функции). Уменьшением расчетного интервала и шага дискретизации аргумента можно уточнить приближенные значения корней функции.

Графическое представление функции, которое может быть получено с помощью соответствующих программных средств, наглядно показывает характер функции на заданном интервале и позволяет инженеру быстро выяснить особенности функциональной зависимости.

Проблема существенно усложняется, если анализируемая функция имеет несколько аргументов. При этом обычным средством анализа является изучение сечений функции, которые получаются при фиксации всех аргументов (переменных), кроме одного, т.е. сведение задачи к анализу многих функций с одной переменной. Современные программные средства позволяют построить многомерные графики для функций нескольких переменных, однако часто многомерное представление настолько сложно и лишено наглядности, что анализ функции выполнить трудно.

Одним из эффективных средств анализа функций является табличный процессор *MS Excel*, который позволяет легко построить таблицы значений функций и получить их графическое представление.

Применяя *MS Excel*, необходимо сначала построить *макет таблицы*, т.е. выбрать ячейки и блоки таблицы, где будут располагаться исходные данные и результаты вычислений. Очевидно, макет таблицы значений функции одной переменной представляет собой два столбца с соответствующими заголовками: в одном столбце располагают значения аргумента, изменяющиеся с постоянным шагом на заданном интервале, в другом – соответствующие значения анализируемой функции.

После выбора столбцов исходной (пустой) таблицы для размещения значений аргумента и соответствующих значений функции необходимо *заполнить* эти столбцы числовыми значениями.

В столбце аргумента задается первое значение из выбранного интервала. Затем либо выделяется ячейка с начальным значением, либо выделяется целиком блок, который необходимо заполнить (вместе с первой ячейкой). В меню **Правка/Заполнить** отмечается позиция **Прогрессия...** В открывшемся диалоговом окне указываются, что нужно выполнить заполнение столбца (по столбцам), указывается тип прогрессии (*арифметическая*), *величина шага* и *конечное значение*, если блок предварительно не выделен. В результате получаем сразу заполненный столбец значений аргумента.

Другой способ заполнения столбца значениями арифметической прогрессии основан на зацеплении мышью маркера заполнения (копирования) в виде маленького квадрата в правом углу ячейки (блока ячеек). Если предварительно ввести в две соседние ячейки таблицы значения первых двух элементов арифметической прогрессии и их выделить, то последующее протягивание мышью маркера заполнения приводит к заполнению столбца требуемыми значениями.

В столбце, отведенном под значения функции, в первую ячейку вводится анализируемая функция. При этом вместо аргумента делается ссылка на адрес ячейки, где находится первое (из выбранного диапазона) значение аргумента. Затем введенная функция копируется (например путем зацепления маркера копирования) во все нижележащие ячейки. Легко заметить, что табличный процессор *MS Excel* при выполнении операции копирования формулы производит автоматическое изменение относительного адреса ссылки на соответствующее значение аргумента, как бы подстраиваясь под задачу.

Вообще говоря, при копировании формул возможны недоразумения, и иногда можно получить неправильный по смыслу результат. Поэтому после копирования формулы (функции) целесообразно выборочно проверить правильность копирования с точки зрения данной задачи.

Полученная таблица значений функции позволяет выполнить ее качественный анализ на выбранном интервале. При необходимости интервал анализа функции или шаг изменения аргумента легко можно изменить. Найденные по таблице приближенные значения корней функции, максимальных и минимальных значений и других значений, интересующих инженера, будут служить в качестве начальных приближений для итерационных процедур численного определения характерных точек функции.

Таблица значений функции служит основой для построения графика анализируемой функции. Для этой цели вызывается **Мастер диаграмм** либо щелчком по соответствующей кнопке на панели инструментов, либо выбором позиции **Диаграммы...** в меню **Вставка**.

Мастер диаграмм выполняется в четыре шага. На первом шаге выбирается тип графика (диаграммы). В инженерных задачах чаще всего применяется разновидность «линейного» графика (позиция **График** или **Точечная**). Для экономистов типичным является построение гистограмм и круговых диаграмм. Диалоговое окно второго шага позволяет задать блок значений функции, по которым строится график, и значения аргумента путем ссылки на соответствующие фрагменты таблицы. Для ссылки на значения аргумента используется вкладка **Ряд**. Другие два шага **Мастера диаграмм** позволяют оформить график (диаграмму), т.е. задать сетку, масштаб по осям, надписи вдоль осей, заголовки и пр.

Вообще говоря, для качественного исследования функциональной зависимости достаточно выполнить первые два шага **Мастера диаграмм**.

Следует отметить, что построенный таким образом график (диаграмма) представляет собой *объект*, встроенный в Документ *MS Excel*. Выделение графического объекта производится щелчком мыши. Затем объект можно масштабировать и перемещать по документу обычными средствами. Элементы диаграммы (графика), такие как заголовки, надписи вдоль осей, легенда и прочие являются в свою очередь объектами, которые подвержены тем же манипуляциям. Например, заголовок диаграммы можно редактировать после выделения щелчком мыши, перемещать в пределах области диаграммы.

Решение нелинейных уравнений. Решить уравнение – означает найти такое значение аргумента, которое превращает уравнение в тождество. Уравнение чаще всего задается в виде $f(x) = 0$. Известные численные методы, реализованные в математических программах, отыскивают значение одного корня функции $f(x)$. Алгоритмы численных методов представляют собой итерационные процедуры – процедуры последовательного приближения к искомому решению начиная от некоторого *начального значения*. Это начальное приближение задается либо явно при описании задачи, либо принимается по умолчанию равным значению, заданному в программе.

Если функция $f(x)$ монотонна на интервале и имеет один корень, то численное решение уравнения обычно сходится к искомому корню. Если функция имеет несколько корней, то поиск нужного корня возможен только после выполнения процедуры выделения *корней*, т.е. выявления таких интервалов, на каждом из которых находится по одному корню. Для выделения корней необходим предварительный анализ функции методами, описанными выше.

Табличный процессор *MS Excel* содержит встроенные средства, реализующие итерационные процедуры подбора параметров (аргументов), которые удовлетворяют некоторым критериям. В частности, эти средства предназначены для нахождения значений аргументов функции, при которых отклонение значения функции от заданного (например нулевого) значения будет минимальным.

В процедуре решения уравнения в программе *MS Excel* отражены особенности табличного процессора. В одну из ячеек таблицы вводится описание функции, аргумент которой – ссылка на ячейку, выбранную для помещения значения аргумента. Числовое значение в этой ячейке является *начальным приближением* к искомому корню. В диалоговом окне, которое вызывается командой **Сервис/Подбор параметра...**, указываются адреса ячеек, где находится описание функции (поле **Установить в ячейке**), начальное приближение (**Изменяя значение ячейки**) и число (**Значение**), с которым сравнивается функция (при поиске корня функции – с нулем). Результат решения уравнения виден в таблице: в ячейке аргумента находится искомое значение корня. Погрешность решения оценивается по итоговому значению функции.

Таким образом, для решения уравнения $f(x) = 0$ необходимо сначала в выбранные ячейки таблицы поместить описание функции, содержащее

вместо аргумента адрес ячейки, выбранной для значения аргумента, и ввести начальное значение аргумента (начальное приближение). Затем в диалоговом окне **Подбор параметра** заполняются текстовые поля. В качестве **Значения функции** указывается 0 (ноль).

Итерационная процедура поиска решения (подбора параметра) управляется заданием требуемой погрешности решения (по значению функции) и допустимого числа итераций. Эти установки делаются во вкладке **Вычисления** окна **Сервис/Параметры...**

Решить уравнение можно также с помощью команды **Сервис/Поиск решения...**, предназначенной для решения задач поиска экстремумов (решения задач оптимизации). Обратим внимание, что этот пункт меню появляется после активации надстройки **Поиск решения** с помощью команды **Сервис/Надстройки...** В окне **Поиск решения** в качестве целевой функции указывается ссылка на ячейку, где находится описание функции, выбирается кнопка **Равной значению**, в текстовом поле указывается значение функции, соответствующее задаче (ноль, если отыскивается корень уравнения $f(x) = 0$), и в поле **Изменяя ячейки** указывается адрес ячейки, где находится значение аргумента (*начальное приближение*). Решение располагается, как и ранее, в ячейке таблицы, выделенной для аргумента.

Решение системы линейных алгебраических уравнений. В инженерных и экономических задачах связи между параметрами исследуемых систем часто представляются линейными алгебраическими уравнениями. Система линейных алгебраических уравнений всегда может быть представлена в матричной форме:

$$\mathbf{A} \mathbf{X} = \mathbf{B}.$$

Решение этой системы имеет вид:

$$\mathbf{X} = \mathbf{A}^{-1} \mathbf{B}.$$

Поскольку матрица по определению является таблицей, для решения матричного уравнения удобно применить табличный процессор *Excel*. В *Excel* встроены функции, выполняющие матричные преобразования, в том числе вычисление обратной матрицы МОБР и произведение матриц МУМНОЖ. Указанные функции относятся к числу функций, обрабатывающих *массивы*. Этим определяются некоторые особенности их использования.

Для решения системы уравнений в матричной форме создается макет используемой таблицы, т.е. выбираются блоки таблицы, в которых будут размещаться исходные матрицы **A** и **B**, матрица, обратная к **A**, и вектор решений **X**. Затем нужно заполнить блоки **A** и **B** исходными данными, т.е. коэффициентами системы уравнений.

В частности, для вычисления матрицы, обратной к **A**, после ввода в таблицу коэффициентов матрицы **A** (например в блок A1:B2 в случае решения системы двух уравнений) выбирают в таблице место, где будет расположен результат (например блок A4:B5), и в одну из угловых ячеек этого блока вводят формулу =МОБР(A1:B2). Затем для вывода всего полученного массива (обратной матрицы) выделяют весь блок A4:B5, переходят в режим редактирования (в строке формул или командой <F2>) и вводят команду <Ctrl>+<Shift>+<Enter>, которая, собственно, и является командой вывода массива.

Аналогично выполняется умножение обратной матрицы на вектор правых частей системы уравнений с помощью функции:

$$=\text{МУМНОЖ}(A4:B5;D1:D2),$$

для которой задаются два аргумента, соответствующие двум сомножителям. Здесь предполагается, что вектор правых частей системы уравнений расположен в таблице в блоке D1:D2.

Результат помещается полностью в выделенный для **X** блок после перехода в режим редактирования и ввода приведенной выше клавишной команды вывода массива.

Можно, конечно, выполнить одним действием (одной формулой) и обращение матрицы, и умножение обратной матрицы на вектор правых частей:

$$=\text{МУМНОЖ}(\text{МОБР}(A1:B2);D1:D2).$$

Определение экстремальных значений функции. Одной из наиболее типичных инженерных задач является задача поиска *локального экстремума* (максимума или минимума) функции при наличии или отсутствии ограничений. К такой математической задаче сводится задача поиска оптимальных значений параметров какого-либо объекта (устройства, системы) при заданном критерии оптимальности.

Сформулируем задачу оптимизации. Требуется найти такие значения параметров **X**, при которых функция $F(\mathbf{X})$ достигает минимального (макс-

симального) значения, при выполнении некоторых условий (ограничений) $Q(\mathbf{X}) > 0$. Здесь \mathbf{X} – вектор параметров оптимизации, $F(\mathbf{X})$ – критерий оптимальности, $Q(\mathbf{X})$ – вектор-функция ограничений.

MS Excel включает в себя удобные и эффективные средства решения таких задач. Естественно, при решении практических задач поиска экстремума могут встретиться значительные трудности, связанные с отсутствием сходимости или медленной сходимостью процедуры поиска, неудачным заданием начального приближения и пр. Эти трудности связаны прежде всего со сложностью задач. Основную роль в успешном решении задач такого класса играет квалификация пользователя в предметной области, т.е. в области решаемых задач, что позволяет пользователю, в частности, задать подходящие начальные приближения. Большое значение имеет также предварительный анализ функции – критерия оптимальности.

Реализованные в *MS Excel* методы поиска экстремумов достаточно эффективны, однако, может оказаться, что более эффективными в конкретных случаях будут другие (более специализированные) программные средства.

Достоинство *Excel* состоит в том, что инженер обычно может не вникать в существо процедуры поиска. Он имеет дело с удобными и наглядными диалоговыми средствами, которые позволяют достаточно просто сформулировать и описать решаемую задачу и оценить результат. Имеются возможности наблюдать за процессом поиска, задавать различные начальные приближения, различные значения погрешности решения, допустимое число итераций, проводить вариантный анализ. Сообщения программы просты и понятны.

Поиск локального экстремума при отсутствии ограничений. Искомый локальный экстремум (максимум или минимум) будет почти наверняка найден даже в случае сложной функции, если задано начальное приближение, достаточно близкое к решению. Поэтому в практических задачах часто бывает необходимо провести предварительный анализ функции (качественно, табулированием или построением графика).

Как и в других задачах прежде всего формируется *макет таблицы*, т.е. выбираются те ячейки таблицы, в которые будут введены описание функции и значение аргумента (*начальное приближение*). Очевидно, в описании функции аргумент заменяется ссылкой на ту ячейку (ее адресом или именем), где находится значение аргумента. Обычными средствами

вводится формула, соответствующая заданной функции. Командой **Сервис/Поиск решения...** вызывается диалоговое окно, в котором указываются адрес ячейки, где находится описание функции, адрес ячейки, где находится аргумент (начальное приближение). В этом же окне задается: максимальное или минимальное значение функций. После щелчка по кнопке **Выполнить**, если решение найдено, выводится окно **Результаты поиска решения**, в котором предлагается сохранить результаты поиска или вернуться к исходным данным.

Результат поиска экстремума располагается в тех ячейках таблицы, где находятся описание функции и значение аргумента.

Аналогично выполняется поиск экстремума функции нескольких переменных. При этом в макете таблицы для размещения значений аргументов (начального приближения) выделяется соответствующий блок ячеек.

Описание задачи *поиска локального экстремума при наличии ограничений* (который часто называют не экстремумом, а оптимумом) отличается тем, что должны быть заданы ограничения в виде равенств или неравенств. Для этого в таблице выбираются некоторые ячейки, в которые вводятся левые и правые части ограничений (в виде формул или чисел). Затем в окне Поиск решения (команда **Сервис/Поиск решения...**) необходимо щелкнуть по кнопке **Добавить** в области **Ограничения**. В окне **Добавление ограничений** вводятся ссылки на соответствующие ячейки таблицы и выбирается тип условия.

Часто искомый экстремум оказывается на границе допустимой (в соответствии с ограничениями) области. В этих случаях в окне **Результаты поиска решения** может быть указано, что **решение не найдено**, однако значения функции и аргумента могут соответствовать решению, находящемуся на границе области. Поэтому прежде чем отбрасывать результат, необходимо его проанализировать.

Аналогичные подходы применяются при решении задач поиска экстремумов функции n переменных. Такие задачи в общем случае часто называют задачами нелинейного программирования.

Решение задач линейного программирования. К задачам линейного программирования сводится большое число задач техники, экономики и управления. Они формулируются как задачи поиска максимального или минимального значения линейной функции и переменных при наличии линейных ограничений в виде равенств или неравенств. Для решения та-

ких задач применяется обычно так называемый *симплекс-метод*. Используя *MS Excel*, мы можем не вникать в существо этого метода, нам необходимо только ввести описание задачи и оценить результат.

Отметим, что в задачах такого класса удачный макет таблицы имеет особое значение: решение задачи становится наглядным, а вероятность ошибок уменьшается, если в макет таблицы ввести поясняющие заголовки и комментарии (желательно их выделять форматированием).

В качестве примера рассмотрим классическую упрощенную задачу планирования производства нескольких типов изделий – задачу распределения ресурсов для получения максимальной прибыли. К ресурсам отнесем трудовые, финансовые, сырьевые. Количество ресурса каждого вида, необходимое для выпуска единицы каждого изделия, называется нормой расхода. Нормы расхода и прибыль, получаемая от реализации изделия каждого типа, относятся к исходной информации. Рассмотрим пример.

| Ресурсы | Изделие 1 | Изделие 2 | Изделие 3 | Изделие 4 | Критерий | Наличие |
|-----------------------|-----------|-----------|-----------|-----------|----------|---------|
| Трудовые, ч | 50 | 40 | 60 | 70 | <= | 10 000 |
| Финансовые, тыс. руб. | 1000 | 700 | 1300 | 1520 | <= | 12 000 |
| Прибыль, тыс. руб. | 600 | 800 | 1200 | 1900 | <= | 18 000 |

Пусть x_i ($i = 1, \dots, 4$) – количество выпускаемых изделий каждого типа. Тогда математически задача формулируется так:

$$50 x_1 + 40 x_2 + 60 x_3 + 70 x_4 \leq 10\,000;$$

$$1000 x_1 + 700 x_2 + 1300 x_3 + 1520 x_4 \leq 12\,000;$$

$$600 x_1 + 800 x_2 + 1200 x_3 + 1900 x_4 \leq 18\,000;$$

$$x_i \geq 0, i = 1, \dots, 4.$$

Целевая функция:

$$F = 700 x_1 + 900 x_2 + 1100 x_3 + 1500 x_4 \rightarrow \max.$$

Информация о задаче вводится в окно **Поиск решения...** так, как описано выше. Ограничения (левые и правые части) вводятся в этом окне после нажатия на кнопку **Добавить**.

Процессом решения задачи можно управлять с помощью окна **Параметры поиска решения**, вызываемого кнопкой **Параметры...** в окне **Поиск решения**. По информации в окне **Результаты поиска решения** можно судить о том, найдено ли решение задачи. Результаты решения распола-

гаются в исходной таблице в строке значений x_i и в ячейке с целевой функцией. По ячейкам, где помещены левые и правые части ограничений, можно судить о выполнении ограничений.

Аппроксимация экспериментальных данных. Более сложной и не менее интересной с практической точки зрения задачей является задача *аппроксимации* экспериментальных данных произвольной непрерывной функцией. Решение таких задач позволяет получить в качестве математических моделей реальных объектов непрерывные функции, которые облегчают исследования этих объектов.

Обычно задачи этого класса решаются на основе классического *метода наименьших квадратов*, суть которого сводится к следующему. Формируется так называемая функция *невязки*, которая представляет собой сумму квадратов отклонений значений аппроксимирующей функции в узлах аппроксимации от заданных экспериментальных значений. Аппроксимирующая функция содержит один или несколько параметров, которые определяются исходя из условия минимума функции невязки. Таким образом, минимизируется суммарное отклонение аппроксимирующей функции от заданных результатов экспериментов.

Очевидно, выбор подходящей аппроксимирующей функции является искусством и опирается на знания, опыт и интуицию инженера. Во многих случаях при выборе аппроксимирующей функции учитывается математическая формулировка соответствующих физических законов.

В качестве примера рассмотрим следующую задачу. Для построения математической модели сложного объекта (системы) часто используют его переходную характеристику, полученную экспериментально как реакцию на единичное ступенчатое воздействие.

Сигнал на выходе объекта измеряют в фиксированные моменты времени и полученный набор дискретных значений (таблицу) аппроксимируют на основе метода наименьших квадратов непрерывной функцией. При желании получить таким способом линейную динамическую модель (если это возможно) в качестве аппроксимирующей функции необходимо выбрать сумму затухающих экспоненциально синусоид, чтобы затем поставить в соответствие объекту линейное дифференциальное уравнение (систему уравнений) или передаточную функцию из набора типовых звеньев. При решении такой задачи чаще всего ограничиваются дифференциальным уравнением второго порядка.

Построим динамическую модель объекта по экспериментально снятой (в равноотстоящих точках t_k) переходной характеристике $y(t)$:

| | | | | | | | | | | | |
|--------|---|-----|------|-----|-----|------|------|-----|------|------|---|
| t | 0 | 0,5 | 1 | 1,5 | 2 | 2,5 | 3 | 3,5 | 4 | 4,4 | 5 |
| $y(t)$ | 0 | 0,6 | 1,05 | 1,2 | 1,1 | 0,95 | 0,95 | 1 | 1,03 | 1,03 | 1 |

В качестве аппроксимирующей функции примем функцию, соответствующую решению линейного дифференциального уравнения второго порядка (колебательный процесс):

$$F(t, \delta, \omega) = 1 - \frac{\sqrt{\delta^2 + \omega^2}}{\omega} e^{-\delta t} \sin \left[\omega t + \arcsin \left(\frac{\omega}{\sqrt{\delta^2 + \omega^2}} \right) \right],$$

где δ – затухание; ω – частота колебательного процесса.

Задача решается методом наименьших квадратов путем поиска минимума функции невязки относительно параметров δ и ω :

$$S(\delta, \omega) = \sum [y(t_k) - F(t_k, \delta, \omega)]^2 \rightarrow \min.$$

Для решения задачи аппроксимации построим следующий макет таблицы. Выделим ячейки (например А5 и В5) для размещения значений искомых коэффициентов δ и ω и поместим туда начальные приближения, например, $\delta = 4$; $\omega = 1$. Построим таблицу из 11 строк, в столбцах которой располагаются последовательно заданные (из таблицы, приведенной выше) значения t_k и $y(t_k)$, соответствующие значения аппроксимирующей функции $F(t, \delta, \omega)$ и значения квадратов отклонений $[y(t_k) - F(t_k, \delta, \omega)]^2$. В ячейку под столбцом отклонений поместим значение суммы квадратов отклонений (функция невязки S), минимум которой нужно найти. В таблицу введем заголовки столбцов и необходимые комментарии.

Таким образом, сформирована исходная информация для поиска минимума функции невязки с помощью режима **Поиск решения** программы *MS Excel*. Результат решения задачи – значения коэффициентов аппроксимирующей функции – располагается в ячейках А5 и В5.

2.2.2. Применение *Mathcad* для автоматизации инженерных расчетов

Решение инженерных задач, задач моделирования и исследования различных объектов машиностроения связано с решением уравнений, опи-

сывающих эти объекты. Это могут быть алгебраические уравнения и системы уравнений, дифференциальные уравнения и системы уравнений. Подходы к решению таких задач известны: это либо *аналитическое*, либо *численное* решение уравнений.

Аналитическое решение уравнений, конечно, предпочтительнее. Однако аналитически решить уравнения, т.е. получить в результате формулу, определяющую решение, удастся только в достаточно простых случаях. В реальных задачах обычно возможно только численное решение уравнений, которое при фиксированных параметрах объекта (коэффициентов уравнений) дает результаты в виде чисел для искомым характеристик объектов.

Аналитическое решение дифференциальных уравнений представляет собой функции, которые позволяют определить значения искомым характеристик, вообще говоря, при любых значениях параметров исследуемого объекта. Численное решение дифференциальных уравнений дает вместо функций таблицы приближенных значений характеристик и для определенного набора аргументов, например времени.

В программу *Mathcad* встроен мощный *символьный процессор*, который позволяет во многих случаях получить решение задач в аналитическом (символьном) виде в результате выполнения символьных преобразований исходных соотношений.

Вычислительные возможности программы *Mathcad* основаны на наиболее известных и проверенных практикой численных методах. Пользователь при решении конкретной задачи может во многих случаях не интересоваться тем, какой численный метод реализован в той или иной функции программы *Mathcad*, если решение задачи получено и не вызывает сомнений.

Однако многообразие практических задач, их сложность могут привести к желанию или необходимости применения нетипичных, более эффективных для данного класса задач или новых численных методов. В ряде случаев оказывается необходимым проследивать ход решения, что не всегда удастся при использовании «замкнутых» функций программы *Mathcad*.

Библиотека встроенных в *Mathcad* функций постоянно расширяется, что обеспечивается возможностями программирования в *Mathcad*, добавлением новых функций в библиотеку, в том числе в виде модулей, написанных на известных языках программирования.

С методической точки зрения программу *Mathcad* можно эффективно использовать для знакомства с существом и особенностями различных численных методов и проведения *численных экспериментов*. Представления о *численных методах* необходимо иметь для понимания методов и средств решения различных инженерных задач на основе современных информационных технологий.

В этом модуле рассмотрены подходы к численному решению наиболее типичных инженерных задач, связанных с решением *обыкновенных дифференциальных уравнений* (расчет переходных процессов) и поиском *экстремальных значений функций* (задачи оптимизации).

Численному решению инженерных задач с помощью ЭВМ должен предшествовать анализ задачи, который призван определить цель, средства, необходимые для ее достижения, принципы представления и обработки данных при решении задачи.

Описание цели решения задачи связано с четким представлением о содержании и форме представления *исходных данных* и определением содержания и формы представления *результатов решения задачи*.

Сформулируем последовательность действий (алгоритм) при численном решении инженерных задач с помощью ЭВМ.

Постановка задачи. Задаются структура исследуемого объекта (системы) и параметры элементов, начальные условия. Устанавливаются цели расчета и анализа, искомые функции и интересующие параметры процессов.

Формализация задачи. При поставленной цели решения задачи она должна быть формализована, и процесс ее решения представлен в виде алгоритма.

Под *алгоритмом* решения задачи будем понимать представление процесса решения задачи в виде однозначной последовательности действий. Выполнение этих действий приводит к решению поставленной задачи. В алгоритме отражаются принципы представления и обработки данных (числовых, символьных, графических). Алгоритм решения задачи описывается обычно словами или представляется графически в виде блок-схемы.

Описание исследуемой системы. На основе фундаментальных законов (физических, экономических и т. п.) формируются уравнения, отображающие исследуемые переходные процессы. В соответствии с разработанным алгоритмом составляется математическое описание задачи.

Математическое описание определяется характером задачи, целями ее решения. Инженерные задачи сводятся обычно к формулам или уравнениям (алгебраическим, дифференциальным), описывающим анализируемую систему (объект).

Выбор методов решения задачи. В соответствии с целями анализа, особенностями анализируемого объекта (системы) и математическим описанием задачи выбирают методы ее решения:

- аналитическое решение;
- численное решение и выбор численных методов решения.

Аналитическое решение реальных задач возможно лишь в достаточно простых случаях.

Выбор инструментария решения задачи. Выбираются или разрабатываются в рамках какой-либо системы программирования необходимые программные средства. Предпочтение отдается мощным математическим программам типа программы *Mathcad*.

Ввод описания задачи. Формируется рабочий документ (программы *Mathcad*), содержащий описание исследуемой системы и средства решения задачи, возможно, с использованием средств программирования.

Выполнение расчетов. На основе созданного описания задачи решение и вывод результатов расчета выполняются в удобной форме (числовой, табличной, графической).

Оценка достоверности результатов. На основе общих представлений об исследуемой системе или *контрольного примера* делают вывод об адекватности решения.

Проведение численных экспериментов. Выполняют ряд расчетов при изменении существенных параметров системы. Определяют необходимые характеристики исследуемого объекта.

Каждый из перечисленных этапов решения задачи требует определенных навыков и умения. Особое внимание должно быть уделено эффективности решения задачи (затраты труда, времени), что вызвано обычно необходимостью проведения ряда численных экспериментов. Эффективность решения непосредственно связана с выбором численных методов решения уравнений, описывающих исследуемую систему.

Процесс исследования сложных систем и объектов на основе численного решения уравнений с применением компьютеров и современных программных средств называют *математическим моделированием* этих систем и объектов.

Методы численного решения обыкновенных дифференциальных уравнений

Анализ переходных процессов в системах различной физической природы (электрических, механических, экономических и пр.) является важнейшей задачей в инженерной практике. Наиболее тяжелые воздействия на элементы любой системы происходят именно в течение переходных процессов изменения состояния системы.

Расчет переходных процессов сводится к определению зависимостей от времени основных характеристик системы путем решения соответствующих дифференциальных уравнений при заданных начальных условиях.

Для расчета переходных процессов интерес представляют средства численного решения задачи Коши, т. е. задачи решения системы обыкновенных дифференциальных уравнений при заданных начальных условиях

В отличие от аналитического решения задачи, когда переходные процессы представлены функциями от времени и параметров системы, значения которых могут быть вычислены в любой точке, результатом численного решения является таблица приближенных значений искомых функций на заданном отрезке в точках, которые называются *узлами сетки*.

Во многих практических задачах расчета переходных процессов в динамических системах дифференциальное уравнение или система дифференциальных уравнений при заданных начальных условиях (задача Коши) *могут быть решены аналитически* и, следовательно, могут быть получены замкнутые выражения (функции времени), описывающие переходные процессы. При этом анализ переходных процессов (характерные значения, длительность переходных процессов и пр.) выполняется путем анализа полученной функциональной зависимости, в частности средствами *Mathcad*.

При невозможности получения аналитического решения (или если решение чрезвычайно громоздко) применяют *численные методы решения дифференциальных уравнений*, которые дают приближенные дискретные значения переменных в узловых точках сетки.

Различают численные методы решения задачи Коши для дифференциального уравнения (системы уравнений), использующие:

- *постоянную величину шага интегрирования* уравнений (равномерная сетка);
- *переменную величину шага интегрирования* (неравномерная сетка), что позволяет на основе изменения величины шага выполнять контроль

погрешности численного решения и обеспечивать заданный уровень погрешности. Эти методы иногда называют *адаптивными*.

В вычислительной математике известно большое число разнообразных численных методов решения обыкновенных дифференциальных уравнений (систем уравнений). Алгоритмы, основанные на различных методах, в той или иной степени учитывают особенности систем уравнений. В частности, разработаны специальные методы и алгоритмы, ориентированные на решение так называемых *жестких* систем уравнений, признаком которых является большой разброс по величине собственных значений матрицы системы уравнений (линейные системы) или собственных значений матрицы Якоби (нелинейные системы), чему соответствует большой разброс постоянных времени анализируемой системы. Существуют также эффективные численные методы, предназначенные для решения *линейных систем* дифференциальных уравнений.

Классические численные методы решения задачи Коши для обыкновенных дифференциальных уравнений (систем уравнений) основаны на разложении искомого решения в степенной *ряд Тейлора* вокруг исходной точки для получения приближенного значения решения в следующей точке сетки.

Порядком численного метода решения дифференциальных уравнений обычно называют число учитываемых слагаемых (членов) в ряду Тейлора.

Различают две большие группы численных методов решения дифференциальных уравнений: явные и неявные методы.

В *явных методах* решение в следующей точке (в следующем узле, на следующем шаге интегрирования) проводится по формулам на основе одного или нескольких *предыдущих* значений переменных в узлах сетки.

В *неявных методах* решение на следующем шаге вычисляется на основе одного или нескольких предыдущих значений путем решения уравнения относительно значения в следующем узле сетки.

В зависимости от того, используются ли для вычисления решения в следующей точке *одно* или *несколько* значений в предыдущих точках, численные методы разделяют на *одношаговые* и *многошаговые*.

Большую группу численных методов решения дифференциальных уравнений составляют *многошаговые методы предсказания и коррекции*, в которых по различным формулам решение в следующей точке сначала предсказывается, а затем уточняется. Методы отличаются друг от друга

способами учета значений в предыдущих узлах сетки и способами предсказания решения в следующей точке (на следующем шаге).

Погрешности решения задач численными методами

Важнейшей проблемой математического моделирования реальных объектов и систем (численного решения инженерных задач) является проблема анализа погрешностей (ошибок), которые связаны с постановкой задач, применяемым методом и тем, что для реализации численного метода используется компьютер, который выполняет вычисления с ограниченным числом значащих цифр.

Погрешности решения задачи с помощью ЭВМ обусловлены следующими причинами:

- математическое описание задачи является неточным, в частности, не точно заданы исходные данные используемой математической модели (параметры и начальные условия);

- применяемый для решения задачи численный метод не является точным; получение точного решения уравнений математической модели чрезвычайно громоздко, даже если возможно;

- при вводе данных, выполнении арифметических операций компьютером и выводе данные округляются.

- Погрешности, вызванные этими причинами, называют соответственно:

- неустраняемыми погрешностями (наследственные ошибки);
- погрешностями метода (ошибки ограничения);
- вычислительными погрешностями (погрешности округления).

Название первого вида погрешности – *неустраняемая* – соответствует ее существу: она может быть уменьшена только путем более точного описания решаемой задачи и более точного задания параметров и в этом смысле является неконтролируемой в процессе численного решения. Иногда неустраняемой (наследственной) называют только погрешность, которая является следствием неточности задания исходных данных. Погрешность, вызванную несоответствием математического описания реальности, называют *погрешностью математической модели*.

Исходные данные очень редко являются точными, так как чаще всего они получены экспериментально или основаны на приближенных оценках. Наследственные ошибки обязаны своим происхождением таким причинам, как неточность измерений, неточность предыдущих вычислений и т.д.

При решении практических задач нет смысла применять методы, обеспечивающие погрешность, существенно меньшую, чем величина неустранимой погрешности.

Ошибки ограничения присущи самим численным методам. Например, при разложении тригонометрических функций в ряд Тейлора невозможно использовать все члены ряда. Отброшенные члены ряда, число которых бесконечно, вносят в результат некоторую ошибку, которую и называют ошибкой ограничения (или отбрасывания).

При выполнении арифметических операций точный результат можно получить, если операции выполняются при учете бесконечного числа значащих цифр. Поскольку реально операции производятся с учетом ограниченного числа значащих цифр, в результат обычно вносится некоторая погрешность, называемая **погрешностью округления**.

Таким образом, при численном решении задач (при *цифровом моделировании* объектов) ошибки (погрешности) всегда имеют место. Анализ поведения ошибок в процессе вычислений и влияния их на адекватность моделирования имеет большое значение.

Погрешность ограничения, привносимая численным методом решения дифференциальных уравнений *на одном шаге интегрирования*, называют *локальной* погрешностью. Ошибку *на всем расчетном интервале*, вызванную этими же причинами, называют *глобальной*. В большинстве методов и алгоритмов предусмотрен контроль величины локальной ошибки; задача оценки глобальной ошибки не решена и теоретически.

Погрешность численного решения дифференциальных уравнений оценивается величиной отброшенных (неучитываемых) членов ряда Тейлора. Для ряда методов получены формулы, позволяющие оценить величину локальной погрешности, что позволяет принимать решения об уменьшении или увеличении шага интегрирования. Погрешность численного решения дифференциальных уравнений существенно зависит от используемой *величины шага интегрирования* (шага сетки).

На практике применяют, например, следующие подходы. Вычисляют решение с шагом h или $h/2$ (два шага). Сравнивая полученные значения, делают вывод о необходимости уменьшения основного шага интегрирования или о возможности его увеличения.

Численная устойчивость методов решения дифференциальных уравнений

Неправильный выбор величины шага может привести к расхождению численного решения или получению решения, неадекватного задаче. Соответствующие примеры неустойчивости методов можно найти в *электронной справочной книге Mathcad (Resource Centre)*.

Известно, что каждой составляющей собственного движения системы или объекта (при отсутствии внешнего воздействия) приписывают понятия *частота* и *затухание*. Величину, обратную *затуханию* каждой составляющей собственного движения, называют *постоянной времени t*.

Системы дифференциальных уравнений с сильно различающимися по величине собственными значениями матрицы системы (или матрицы Якоби) называют *жесткими системами*. Методы решения уравнений, численно устойчивые при изменении шага интегрирования в широких пределах, называют *жестко устойчивыми методами*.

Численное решение обыкновенных дифференциальных уравнений средствами программы Mathcad

В *Mathcad* встроен ряд *специальных функций*, предназначенных для численного решения систем обыкновенных дифференциальных уравнений с начальными условиями (задача Коши). Основными встроенными функциями являются:

- *rkfixed*($y, x1, x2, npoints, D$) – решение задачи на *отрезке* методом Рунге – Кутты с постоянным шагом интегрирования;
- *Rkadapt*($y, x1, x2, npoints, D$) – решение задачи на *отрезке* методом Рунге – Кутты с автоматическим выбором шага интегрирования;
- *rkadapt*($y, x1, x2, acc, npoints, D, kmax, save$) – решение задачи Коши в *заданной точке* методом Рунге – Кутты с автоматическим выбором шага интегрирования;
- *Bulstoer*($y, x1, x2, npoints, D$) – решение задачи на *отрезке* методом Булирша – Штера;
- *bulstoeri* ($y, x1, x2, acc, npoints, D, kmax, save$) – решение задачи в *заданной точке* методом Булирша – Штера; эффективен для уравнений, имеющих гладкое решение.

Во всех перечисленных функциях параметры имеют следующий смысл: y – вектор начальных условий; $x1, x2$ – начальная и конечная точки *отрезка* интегрирования; для функций, вычисляющих решение в *заданной*

точке, $x1$ – начальная точка, $x2$ – заданная точка; $npoints$ – число узлов на отрезке; D – имя вектор-функции правых частей уравнений; J – имя матрицы-функции, содержащей матрицу Якоби частных производных правых частей уравнений; ase – параметр, контролирующий погрешность решения при автоматическом выборе шага интегрирования (если погрешность решения больше параметра ase , то шаг сетки уменьшается, но он не может быть меньше величины $save$); $kmax$ – максимальное число узлов сетки, в которых может быть вычислено решение на отрезке; $save$ – наименьшее допустимое значение шага неравномерной сетки.

Кроме указанных выше функций, решающих численно системы обыкновенных дифференциальных уравнений, приведенных к нормальной форме (с выделенными первыми производными), в *Mathcad* введена функция $odesolve(x,b[,steps])$, которая возвращает решение дифференциальных уравнений, описанных в стандартном для *Mathcad* вычислительном блоке *Given*. В этом же блоке задаются начальные условия. Вычислительный блок открывается служебным словом *Given* и заканчивается строкой, содержащей функцию $odesolve$.

Главное достоинство использования функции $odesolve$ заключается в том, что описываемые в блоке *Given* дифференциальные уравнения *могут быть не приведены к нормальной форме*.

Вместо использования встроенных в *Mathcad* функций для численного решения систем обыкновенных дифференциальных уравнений можно разработать в виде программного модуля собственное описание функции, реализующей какой-либо другой численный метод решения дифференциальных уравнений. Для этого используются средства программирования, имеющиеся в *Mathcad*.

2.3. Инженерный анализ и автоматизация проектирования

2.3.1. Понятие о CAD/CAM/CAE-системах

Автоматизированное проектирование представляет собой технологию, состоящую в использовании компьютерных систем для облегчения создания, изменения, анализа и оптимизации проектов. Для этой цели используются собственно системы автоматизированного проектирования и конструирования, автоматизированного производства и автоматизированного инженерного анализа.

Можно сказать, что любая программа, работающая с компьютерной графикой, так же, как и любое приложение, используемое в инженерных расчетах, относится к системам автоматизированного проектирования (САПР). Множество средств САПР включает как программы для работы с геометрическими моделями, так и специализированные приложения для анализа и оптимизации. К САПР относятся и программы для анализа допусков, расчета масс-инерционных свойств, моделирования методом конечных элементов и визуализации результатов анализа.

Автоматизированное проектирование и конструирование (*computer-aided design – CAD*) – это технология, состоящая в использовании компьютерных систем для геометрического моделирования изделия. Основная функция *CAD* – определение геометрии конструкции (детали механизма, архитектурные элементы, электронные схемы, планы и т.п.), поскольку геометрия определяет все последующие этапы жизненного цикла продукта. Для этой цели обычно используются системы разработки рабочих чертежей и геометрического моделирования. Вот почему эти системы обычно и считаются *CAD*-системами. 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.

Автоматизированное производство (computer-aided manufacturing – CAM) – это технология, состоящая в использовании компьютерных систем для планирования, управления и контроля операций производства через прямой или косвенный интерфейс с производственными ресурсами предприятия. Одним из наиболее зрелых подходов к автоматизации производства является числовое программное управление (ЧПУ). ЧПУ заключается в использовании запрограммированных команд для управления станком. Современные *CAM*-системы способны автоматически генерировать программы для станков с ЧПУ на основании геометрических параметров изделий из базы данных *CAD* и дополнительных сведений, предоставляемых пользователем.

Планирование процессов также постепенно автоматизируется с помощью нового поколения *CAM*-систем и автоматизированных систем управления производственными процессами (АСУ ПП). План процессов

может определять последовательность операций по изготовлению устройства от начала и до конца на всем необходимом оборудовании. Хотя полностью автоматизированное планирование процессов практически невозможно, план обработки конкретной детали вполне может быть сформирован автоматически, если уже имеются планы обработки аналогичных деталей. Для этого разработана технология группировки, позволяющая объединять схожие детали в семейства. Детали считаются подобными, если они имеют общие производственные особенности (гнезда, пазы, фаски, отверстия и т. д.). Для автоматического обнаружения схожести деталей необходимо, чтобы база данных *CAD* содержала сведения о таких особенностях. Эта задача осуществляется при помощи объектно-ориентированного моделирования или распознавания элементов.

Автоматизированный инженерный анализ (computer-aided engineering – CAE) – это технология, состоящая в использовании компьютерных систем для анализа геометрии *CAD*, моделирования и изучения поведения продукта для усовершенствования и оптимизации его конструкции. Средства *CAE* могут осуществлять множество различных вариантов анализа. Программы для кинематических расчетов, например, способны определять траектории движения и скорости звеньев в механизмах. Программы динамического анализа с большими смещениями могут использоваться для определения нагрузок и смещений в сложных составных устройствах типа автомобилей. Программы верификации и анализа логики и синхронизации имитируют работу сложных электронных цепей.

Из всех методов компьютерного анализа наиболее широко в конструировании используется *метод конечных элементов (finite-element method – FEM)*. С его помощью рассчитываются напряжения, деформации, теплообмен, распределение магнитного поля, потоки жидкостей и другие задачи с непрерывными средами, решать которые каким-либо иным методом оказывается просто непрактично.

Для использования метода конечных элементов нужна абстрактная модель подходящего уровня, а не сама конструкция. Абстрактная модель отличается от конструкции тем, что она формируется путем исключения несущественных деталей и редуцирования размерностей, например, трехмерный объект небольшой толщины может быть представлен в виде двумерного листового изделия. Программные средства, позволяющие конструировать абстрактную модель и разбивать ее на конечные элементы, на-

зываются *препроцессорами*. Проанализировав каждый элемент, компьютер собирает результаты воедино и представляет их в визуальном формате. Программные средства, обеспечивающие визуализацию результатов конечно-элементного анализа, называются *постпроцессорами*.

К *CAE*-системам также относятся программные средства для оптимизации конструкций. В этих системах исходная форма конструкции предполагается простой, как, например, у прямоугольного двумерного объекта, состоящего из небольших элементов различной плотности. Затем выполняется процедура оптимизации, позволяющая определить конкретные геометрические параметры, позволяющие достичь определенной цели с учетом ограничений на напряжения. Обычно целью оптимизации является минимизация веса или габаритов.

Достоинство методов анализа и оптимизации конструкций заключается в том, что они позволяют конструктору увидеть поведение конечного продукта и выявить возможные ошибки до создания и тестирования реальных прототипов, избежав определенных затрат. Поскольку стоимость конструирования на последних стадиях разработки и производства продукта достаточно велика, ранняя оптимизация и усовершенствование ведут к значительному снижению сроков и стоимости разработки.

Широко известные коммерческие программы *CAD/CAM/CAE* перечислены в таблице. Интегрированные системы в крайнем правом столбце таблицы представляют функциональность *CAD*, *CAM* и *CAE* посредством подключаемых модулей.

| Область применения | Программы | Интегрированные системы |
|---|---|---|
| <i>CAD</i> : двумерные чертежи | <i>CADAM, AutoCAD, MicroCADAM, VersaCAD</i> | <i>Pro/ENGINEER</i> <i>Unigraphics</i> |
| <i>CAD</i> : твердотельное моделирование | <i>Solid Edge, SolidWorks, SolidDesigner, Mechanical Desktop</i> | <i>CATIA</i> <i>I-DEAS</i> <i>I/EMS</i> |
| <i>CAM</i> | <i>BravoNCG, VERICUT, DUCT, Camand, Mastercam, PowerMILL, T-Flex, ТехноПро</i> | <i>EUCLID-IS</i> КОМПАС |
| <i>CAE</i> | <i>ANSYS, MSC/NASTRAN, PATRAN, DADS, ADAMS, C-Mold, MoldFlow, Design Works, ПОЛИГОН</i> | |

2.3.2. Системы геометрического моделирования

Процесс проектирования может рассматриваться как постепенная детализация формы по мере развития идей разработчика. Программное обеспечение автоматизированного проектирования – это просто одно из средств, облегчающих этот процесс. Типичные *CAD*-программы могут быть разделены на две группы. Системы автоматизированной разработки чертежей помогают проектировщику реализовать свои идеи в двумерном пространстве. Системы геометрического моделирования позволяют работать с формами в трехмерном пространстве.

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. Трехмерная 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. Системы геометрического моделирования делятся на каркасные, поверхностные и твердотельные (перечислены в порядке появления).

В *системах каркасного моделирования* форма представлена в виде набора характеризующих ее линий и конечных точек. Линии и точки используются для создания трехмерных объектов на экране, а изменение формы осуществляется путем изменения положения и размеров отрезков и точек. Другими словами, визуальная модель – каркасный чертеж формы, а соответствующее математическое описание – 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 с помощью последовательных простых действий, так что пользователям было достаточно легко создавать формы самостоятельно. Однако визуальная модель, состоящая из одних линий, может быть неоднозначной. Более того, соответствующее матема-

тическое описание не содержит сведений о внутренних и внешних поверхностях моделируемого объекта. Без этих сведений невозможно рассчитать массу объекта, определить траектории перемещения инструмента при его обработке 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 *системах поверхностного моделирования* 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. Математическое описание может включать сведения о связности поверхностей, то есть о том, как поверхности соединяются друг с другом и по каким кривым.

Существует три стандартных метода создания поверхностей в системах поверхностного моделирования:

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

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.

Системы твердотельного моделирования (solid modeling systems) предназначены для работы с объектами, состоящими из замкнутого объема. 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.

Однако создание модели в виде замкнутого объема требует большего количества входных данных по сравнению с количеством данных, дающих математическое описание. Это одна из причин, по которым были разработаны системы моделирования *немногообразных объектов*. Такие системы моделирования позволяют работать с поверхностями и замкнутыми объемами одновременно.

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*. К этой же группе относятся функции добавления и вычитания объема – *булевские операторы*. Функции моделирования из первой группы позволяют проектировщику быстро создать форму, близкую к окончательной форме детали.

Ко второй группе относятся функции создания объемных тел путем перемещения поверхности. Функция *заметания (sweeping)* позволяет создавать объемное тело трансляцией или вращением плоского сечения. Задавая замкнутую плоскую область, пользователь может указывать геометрические ограничения или вводить данные о размерах, а не рисовать форму вручную – 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. 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 др. 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 *моделирова-*

нием границ (boundary modeling). В последнюю группу входят функции, используя которые проектировщик может моделировать твердое тело при помощи свободных форм. Например, он может давать системе команды типа «сделать отверстие определенного размера в определенном месте» или «сделать фаску определенного размера в определенном 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.

Объектно-ориентированное моделирование 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 определенного размера в определенном месте» или «сделать фаску определенного размера в определенном 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 автоматически сформировать план технологического процесса. Таким образом, лучше моделировать объекты с использованием команд объектно-ориентированного моделирования, а не просто булевских операций. Модель, созданная при помощи таких команд, облегчит планирование технологического процесса, если не сделает его полностью автоматическим.

Один из недостатков объектно-ориентированного моделирования заключается в том, что система не может полностью предоставить нужные элементы для всех возможных приложений. Для каждой задачи может по-

требоваться 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.

2.3.3. Системы инженерного анализа методом конечных элементов

В современном проектировании широко используются различные программные пакеты инженерного анализа, которые оценивают проекты на каждом этапе процесса разработки. Средства *CAE* позволяют анализировать кинематику или динамику поведения проектируемого агрегата. При этом каждый компонент агрегата рассматривается как тело с сосредоточенной массой. Более того, пакеты математического моделирования (*Mathcad*, *MATLAB*, *Maple*), программы проектирования систем (*LabVIEW*, *SIMULINK*, *VisSim*, *P-CAD*) и электронные таблицы в той или иной мере можно отнести к системам инженерного анализа, так как все они позволяют автоматизировать труд инженера. Однако большинство инженерных задач проектирования: определение напряжений, вибраций или температур в механических компонентах, рассчитанных на физическую, динамическую или тепловую нагрузку, решаются при помощи метода конечных элементов (МКЭ). Поэтому сформировалась тенденция к системам инженерного анализа относить *CAE*-системы, основанные на МКЭ.

Метод конечных элементов появился для решения задач строительной механики. Однако МКЭ имеет более широкую область применения: задачи теплопереноса, распределения электростатического потенциала, механики жидкостей, вибрационного анализа и многие другие. В качестве примеров программ для решения задач механики жидкостей методом конечных элементов можно привести пакеты *C-Mold* и *MoldFlow*, предназначенные для моделирования течения жидкого пластика в форме для литья под давлением, отечественный пакет ПОЛИГОН анализа литейных процессов в отливке. Примерами универсальных коммерческих программ конечно-элементного анализа являются *NASTRAN* и *ANSYS*.

Метод конечных элементов является наиболее популярным численным методом решения таких задач. Универсальность этого метода удовлетворяет требованиям современных сложных систем конструирования, для которых обычно отсутствуют замкнутые решения уравнений равновесия. Анализ МКЭ начинается с аппроксимации исследуемой области (области задачи) и делении ее на ячейки сетки.

Выбор элементов определяется областью задачи, ее типом, а также конкретным пакетом анализа. Выбор подходящих элементов с нужным количеством узлов из библиотеки доступных элементов является одним из наиболее важных решений, которые приходится принимать пользователю пакета конечно-элементного анализа. Классическая форма метода конечных элементов называется *h-версией*. В качестве функции формы в этой версии используются кусочные полиномы фиксированных степеней, а повышение точности достигается уменьшением размеров ячеек. В *p-версии* используется фиксированная сетка, а точность повышается благодаря увеличению степени функции формы. Общее правило состоит в том, что чем больше количество узлов и элементов (в *h-версии*) или чем выше степень функции формы (в *p-версии*), тем точнее оказывается решение, но тем дороже оно стоит с вычислительной точки зрения (функции формы – независимые полиномы, определяющие аппроксимацию переменной, относительно которой решается задача).

Следующий шаг – создание ячеек сетки и распределение узлов. Построение сетки является важнейшим и сложнейшим этапом МКЭ моделирования. Для упрощения этой задачи практически все системы на сегодняшний день предлагают различные функции автоматизации. Наиболее типично использование тетраэдрических элементов для объемных тел и четырехугольных или треугольных элементов для трехмерных поверхностей, оболочек и двухмерных объектов. Многие системы предоставляют пользователям возможность изменять параметры автоматически формируемых сеток, в частности плотность ячеек. Кроме того, в таких системах обычно имеются функции ручного локального редактирования, позволяющие уточнить сетку в критических областях. Ряд систем связывают сетку с геометрической моделью, так что изменение последней автоматически влечет за собой изменение первой.

От сложности сетки зависят размер глобальной матрицы жесткости, численная сложность задачи и объем требуемых вычислительных ресур-

сов. Точность решения можно повысить увеличением количества ячеек или использованием функций формы более высоких порядков. Конечные элементы должны удовлетворять определенным требованиям. Во-первых, размерность элементов должна совпадать с размерностью области задачи. Для одномерных задач используются одномерные элементы, для двумерных – двумерные и т. д. Во-вторых, конечные элементы должны поддерживаться выбранной САЕ-программой. Чем больше элементов в библиотеке программы, тем больше задач она может решить.

Другой подход к решению проблем формирования сетки предлагает *p*-версия МКЭ. *P*-версия использует простые сетки, формируемые автоматически, но зато в этой версии может изменяться степень функции формы (также автоматически). Существует достаточно много *FEA*-программ, поддерживающих *p*-версию анализа, но только две программы разработаны специально для этой версии: *Pro/MECHANICA* фирмы PTC и *PolyFEM* фирмы CADSI. Преимущества этого подхода не ограничиваются простотой сеток. *P*-версия позволяет задавать конкретные ограничения на точность, а также лучше аппроксимировать геометрические модели из *CAD*-программ. Низкий уровень точности позволяет конструктору быстро получить результаты анализа на предварительном этапе разработки.

За выбором элементов следует задание типа анализа (статический или динамический, линейный или нелинейный, анализ деформаций, напряжений и т.д.). С каждым узлом связываются неизвестные или степени свободы. К неизвестным относятся смещения, повороты, температура, тепловые потоки и т. п. Затем задаются граничные условия. Для непрерывных границ объекта известными могут быть смещения, внешние силы и температура. Эти сведения должны быть выражены в виде значений соответствующих параметров в конкретных граничных узлах. Иногда требуется формирование конечных элементов без граничных условий. Если необходимо учесть точечные воздействия, в соответствующих точках должны располагаться узлы. Большинство систем анализа, интегрированных с *CAD*, дают пользователю возможность задавать граничные условия непосредственно на геометрической модели, после чего эти граничные условия преобразуются к эквивалентным условиям на узлах системы. Нагрузки и граничные условия задаются множеством способов, что позволяет решать задачи самого широкого круга и моделировать реальные условия достаточно точно.

Для каждого элемента обязательно задание свойств материала. Обычно эти параметры включают модуль Юнга и коэффициент Пуассона (для задач строительной механики). Толщина оболочек и пластин рассматривается скорее как свойство материала, чем как геометрический параметр, что позволяет избежать перехода к трем измерениям. Для задач других типов возможно будут заданы теплоемкость или вязкость. Разные элементы могут иметь разные свойства, благодаря чему пользователь может анализировать составной объект, о чем уже говорилось выше. Основные сложности в описании составных объектов возникают при задании интерфейсов.

Полностью определенная конечно-элементная модель со всеми параметрами передается программе анализа. Решенная задача подготавливается к исследованию постпроцессором. Большинство пакетов позволяют вычислять различные параметры, выводить их в виде таблиц или графиков. Чаще всего требуется вывод данных о деформациях, напряжениях и изменении формы. Для этой цели традиционно используются контурные графики, на которых распределение параметров кодируется различными цветами непосредственно на изображении объекта. Большинство пакетов уже ушли от столь примитивной графики. Пользователь современной системы может выводить на экран изоповерхности (поверхности с постоянными значениями какого-либо параметра) или поперечные сечения. Для динамического анализа удобно наличие средств анимации, позволяющих проводить нелинейный анализ временной эволюции систем. Все более возрастает потребность в выводе графиков и роликов в форматах, пригодных для использования в других программах, документах, презентациях и сети Интернет.

2.3.4. Системы автоматизированного производства

Спроектированная деталь воплощается в готовый продукт средствами производства. Автоматизация производства обеспечивается соответствующим программным обеспечением (*CAM*-система). Таких программных продуктов существует достаточно много. В состав типичного пакета *CAM* входят система автоматизированной технологической подготовки производства, система числового управления, позволяющая изготавливать деталь при помощи станков с ЧПУ, программы контроля и управления роботами, используемые на этапах проверки и сборки.

Первоочередной задачей *CAM*-систем является полная автоматизация технологической подготовки производства, потому что этот этап связывает проектирование и производство. Именно подготовка производства остается основным препятствием на пути к интеграции *CAD* и *CAM*. Основные усилия исследователей направлены на создание систем автоматизированной подготовки производства, которые должны обеспечить взаимодействие инженеров-проектировщиков и инженеров-технологов.

Все производство может быть поделено на дискретное и непрерывное. Под *дискретным производством* понимается изготовление продукта, проходящего через конечное число технологических и сборочных операций. *Непрерывное производство* подразумевает изготовление продукта, претерпевающего непрерывные изменения, например в ходе химических реакций, в результате которых заготовка преобразуется в готовую деталь. Мы остановимся на дискретном производстве, а конкретнее – на механообработке, которая является типичным методом изготовления деталей.

Когда готовый проект передается в производственный отдел, инженер-технолог преобразует описание деталей и устройств в технологические инструкции. Эти инструкции подробно описывают процессы, необходимые для превращения необработанных заготовок в готовые детали, а также последующие операции сборки этих деталей в конечный продукт. Процедура, таким образом, заключается в сопоставлении требований к деталям и имеющихся производственных мощностей. Инженер-технолог должен интерпретировать инженерно-техническую документацию, принимать решения о том, как разрезать листы и собирать детали, определять порядок выполнения операций, выбирать инструмент, станки и крепеж, решать другие подобные задачи. Эта задача значительно упрощается, если у инженера уже есть готовый план для аналогичного продукта. Поэтому похожие детали обычно группируются в семейства, что позволяет использовать концепцию групповой обработки.

После завершения фазы технологической подготовки начинается реальное производство детали в соответствии с инструкциями, подготовленными на предыдущем этапе. Если для обработки детали используются станки с ЧПУ, оператор станка должен написать соответствующую программу. Существует множество программных средств, позволяющих создать программу для станка с ЧПУ непосредственно по базе данных системы автоматизированного проектирования. Готовые детали проверяются в соответствии с разработанными стандартами качества.

Таким образом, интерфейсом между проектированием и производством оказывается технологическая подготовка. Интеграция средств проектирования и производства не будет завершена до тех пор, пока технологическая подготовка не будет автоматизирована.

Технологическая подготовка производства (process planning) заключается в выборе технологических процессов и их параметров, а также оборудования для проведения этих процессов. Задача состоит в том, чтобы превратить заготовку в деталь, изображенную на техническом чертеже. Альтернативное определение технологической подготовки гласит, что под этим термином подразумевается подготовка подробных технологических инструкций для станка или сборщика агрегата из деталей.

На выходе этапа технологической подготовки получается план, описывающий последовательность технологических процессов или сборочных операций. План производства иногда называется операционной картой, маршрутной картой или сводкой планирования операций. Помимо выбора и упорядочения операций важную часть плана составляет выбор инструментов и крепежа. Выбор инструмента включает также выбор станка, на котором этот инструмент будет установлен. Крепежные устройства направляют инструмент или держат обрабатываемую деталь.

План производства детали или агрегата зависит от множества факторов. К ним относятся геометрия детали, требуемая точность и качество поверхности, количество деталей и используемый материал. Например, для изготовления очень гладкой поверхности может потребоваться шлифовка, тогда как для более грубой поверхности достаточно токарной обработки (при той же самой геометрии). Небольшое количество деталей можно изготовить на станке, а большие количества выгоднее штамповать на прессе. Выбор операций также во многом определяется имеющимися средствами.

Традиционно планирование производства выполнялось вручную. Неавтоматизированный подход состоит в том, что опытный технолог изучает чертеж детали и подготавливает инструкции по ее изготовлению, т.е. план производства. Типичная последовательность этапов планировки производства следующая:

1. *Изучение формы детали в целом.* Технолог изучает инженерно-техническую документацию, определяет общую структуру детали и потенциальные трудности, которые могут возникнуть при ее производстве.

2. *Определение оптимальной формы заготовки, если она не задана в документации.* По чертежу планировщик определяет форму детали. Это помогает ему выбрать форму заготовки, из которой данная деталь может быть изготовлена с минимальным объемом отходов.

3. *Определение базовых поверхностей и конфигураций.* Инженер-технолог определяет минимальное количество конфигураций, необходимых для получения базовых поверхностей механической обработкой. Затем он записывает операции для каждой конфигурации.

4. *Определение элементов детали.* Инженер-технолог выделяет элементы детали, то есть геометрические формы, которые должны быть вырезаны на заготовке. Форма элементов определяет форму инструментов и траекторию их перемещения при обработке заготовки.

5. *Группировка элементов по конфигурациям.* Инженер-технолог группирует элементы таким образом, что каждая группа формируется в рамках одной и той же конфигурации. Некоторые детали могут быть произведены в конфигурациях, определенных ранее для базовых поверхностей, другие могут потребовать задания новых конфигураций. Затем формируется список операций по изготовлению элементов детали для каждой конфигурации.

6. *Упорядочение операций.* Внутри каждой конфигурации порядок операций по производству соответствующих базовых поверхностей и элементов определяется взаимозависимостями этих операций и их взаимным влиянием.

7. *Выбор инструментов для каждой операции.* Технолог старается использовать один и тот же инструмент для нескольких операций. Ему приходится учитывать время на смену инструмента и время на обработку.

8. *Выбор или проектирование зажимов для каждой конфигурации.* Этот этап планирования сильно зависит от опыта технолога, потому что количество стандартных зажимов и крепежных приспособлений невелико. Правильный выбор зажимов очень важен для достижения высокого качества продукта.

9. *Итоговая проверка плана.* Инженер-технолог проверяет осуществимость конкретных конфигураций, вероятность создания помех инструментам крепежными приспособлениями и т.д.

10. *Уточнение плана производства.* Технолог добавляет в план подробности по изготовлению отдельных элементов, выбирает скорость подачи и обработки, оценивает затраты и время изготовления и т.д.

11. *Подготовка документации.* Готовый технологический план производства отдается главному технологу.

В компьютеризованных системах автоматизированной технологической подготовки применяются два подхода – модифицированный и генеративный.

Модифицированный подход – модификация неавтоматизированного. Типичный технологический план производства подобной детали может автоматически извлекаться из такого файла после описания анализируемой детали в соответствии с определенной системой кодирования. Выбранный план производства может редактироваться в интерактивном режиме; в него вносятся поправки, соответствующие специфике конкретной детали. Таким образом, модифицированный подход требует наличия базы данных со стандартными планами производства для каждого семейства деталей. Такой план должен содержать все инструкции, которые будут входить в план производства любой детали из данного семейства. Детали классифицируются по семействам на основании концепции групповой технологии. Согласно этой концепции каждой детали присваивается код, зависящий от ее элементов, после чего детали группируются в семейства в соответствии с присвоенными кодами.

Модифицированный подход к разработке плана производства выражается в следующем. Технологическая подготовка производства новой детали начинается с кодирования ее особенностей, что эквивалентно описанию детали на языке групповой технологии. Затем деталь может быть отнесена к какому-либо семейству на основании ее кода. После этого из базы данных извлекается стандартный план производства для деталей этого семейства. В этом плане содержатся общие инструкции по производству любых деталей семейства, поэтому может потребоваться его редактирование для получения плана нужной детали. Редактирование осуществляется средствами компьютерной системы. Часто изменения оказываются незначительными, потому что новый план представляет собой лишь небольшую модификацию стандартного. Благодаря этому на этапе подготовки плана экономится масса времени, а готовые планы оказываются гораздо более последовательными, чем разрабатываемые вручную. Если деталь не может

быть отнесена к одному из существующих семейств, технолог может разработать новый стандартный план производства в интерактивном режиме.

Генеративный подход состоит в том, что технологический план вырабатывается автоматически на основании технических требований к детали. В технические требования должны включаться подробные сведения о материале, особенностях обработки и предлагаемых методиках проверки, а также графическое изображение формы детали.

На первом этапе разработки плана производства новой детали в генеративном подходе технические требования вводятся в компьютерную систему. В идеале они должны считываться непосредственно из базы данных САПР. Для этого необходимо, чтобы автоматизированная система технологической подготовки могла распознавать элементы детали, требующие машинной обработки, такие как отверстия, пазы и выемки. Реализация первого этапа значительно упрощается, если при моделировании детали используется объектно-ориентированный подход. Однако даже конструктивные элементы, используемые в системе объектно-ориентированного моделирования, могут потребовать преобразования к элементам, которые могут быть изготовлены машинной обработкой. Некоторые конструктивные элементы однозначно сопоставляются технологическим, тогда как преобразование других представляет собой не слишком тривиальную процедуру. Кроме того, информации об элементах, вообще говоря, недостаточно для технологической подготовки производства. Например, большинство моделей *CAD* не содержат сведений о допусках и материалах и их приходится вводить вручную. Это лишь часть причин, задерживающих разработку полностью автоматизированных систем технологической подготовки производства до настоящего времени. Вместо этого технические требования к детали часто кодируются вручную. Схема кодирования должна определять все геометрические элементы и их параметры, в частности положение, размеры и допуски. Закодированные данные сопровождаются информацией в текстовом формате. Наконец, система должна иметь сведения о форме заготовки.

На втором этапе закодированные данные и текстовая информация преобразуются в подробный технологический план производства детали. На этом этапе определяются оптимальная последовательность операции и условия их выполнения. К условиям относятся используемые инструменты, крепления, измерительные приборы, зажимы, схемы подачи и скорости

обработки. Для построения столь подробного плана производства детали произвольной сложности требуются большая база данных и сложная логическая система. Поэтому на сегодняшний день автоматизированный подход ограничивается отдельными классами деталей с относительно ограниченным набором элементов.

В настоящее время наибольшее распространение получила модифицированная концепция применения компьютера для автоматизации проектирования технологии, основанная на создании человеко-машинных систем, в которых общение технолога с ЭВМ происходит в интерактивном (диалоговом) режиме. Такие системы получили название *Системы автоматизированного проектирования технологических процессов (САПР ТП)*. На западе эти системы получили название *Computer Automated Process Planning (CAPP)*.

При проектировании технологических процессов у технолога имеются следующие сферы деятельности:

- *оформление технологической документации* (маршрутные, операционные карты и другие документы). Это рутинная работа и может быть полностью автоматизирована;
- *поиск информации* (поиск инструмента, приспособлений, оборудования, заготовок, припусков, нормативов по режимам резания и нормам времени и т.д.). Эта процедура автоматизируется на основе использования информационно-поисковой системы и систем управления базами данных и базами знаний. При использовании ИПС условие поиска технолог вводит в режиме диалога. Условия поиска, которые являются стабильными, можно хранить в базе знаний;
- *стандартные расчеты* (расчет припусков, операционных заготовок, режимов резания и т. п.). Такие расчеты можно полностью автоматизировать.
- *принятие сложных логических решений* (выбор структуры процесса и операций, выбор баз и т. д.). Несмотря на то, что процесс принятия таких решений полностью автоматизировать не удастся, применение экспертных систем в большинстве случаев позволяет технологу находить решение.

2.3.5. Системы управления данными об изделии

Цикл разработки продукта включает в себя не только проектирование и производство, но и анализ, контроль качества, упаковку, доставку и мар-

кетинг. Цель компьютеризации состоит в том, чтобы интегрировать все эти виды деятельности посредством общей базы данных и единого подхода, поэтому существует потребность в механизме передачи информации между ними.

Так, маркетинговые концепции должны передаваться в отдел планирования продуктов и в группу проектирования. Группа проектирования взаимодействует с производственным отделом. Сведения об обнаруженных проблемах возвращаются в отделы проектирования и производства. Сведения о продажах и проектах передаются поставщикам и партнерам. Данные о расходах обрабатывает бухгалтерия. Все эти виды деятельности тесно связаны друг с другом, поэтому изменения в содержании или состоянии проекта должны быть доступны всем участникам во избежание серьезных ошибок. Объем технических данных по мере разработки чрезвычайно возрастает, из-за чего просмотр и поиск по базе данных становятся крайне неэффективными. Для решения подобных задач существуют программные пакеты, называемые *системами управления данными о продуктах (product data management – PDM)*.

Системы *PDM* были разработаны для управления электронными данными, создаваемыми системами *CAD*, *CAM* и *CAE*. Важность *PDM* особенно возросла, когда детали стали разрабатываться в разных *CAD*. Собственная система управления, встроенная в программу *CAD*, могла достаточно эффективно работать с чертежами и моделями, созданными в этой конкретной программе, но чаще всего оказывалась неспособной взаимодействовать с другими программами *CAD* или продуктами третьих фирм. Системы *PDM*, связанные со множеством различных пакетов приложений, лучше справляются с управлением данными в масштабе предприятия. Вдобавок *PDM* облегчают доступ к вспомогательным данным: номерам деталей, техническим требованиям, результатам тестов и анализов.

Затем возможности *PDM* были расширены. К ним добавилась поддержка проектирования путем автоматизированной маршрутизации документации на этапе корректирования. Безбумажный документооборот сократил длительность цикла разработки продукта. Появилась возможность выделять важные данные и отслеживать историю каждого продукта и его компонентов вместе со всеми данными по различным версиям и модификациям продуктов. Вскоре область применения *PDM* была еще более расширена, охватив не только проектирование и разработку, но и отделы продаж, производства и поддержки.

В последнее время системы *PDM* стали еще более популярны благодаря развитию *Internet*, *Web* и интрасетей. Вообще говоря, практически все производители *PDM* рекламируют поддержку работы с *Web*, а *Web*-технологии действуют как катализатор внедрения *PDM* в новые корпорации. Концепция *Web* стала популярной благодаря универсальности, дешевизне и доступности, а также аппаратной независимости. Самым большим препятствием на пути к широкому признанию *PDM* были трудности с доступом у конечных пользователей. *Web* решает эту проблему, предоставляя простой и универсальный интерфейс пользователя при очень низких затратах на поддержку. Благодаря поддержке *Web*-технологий *PDM* стали играть ключевую роль в обеспечении доступа к актуальной информации множеству групп. *PDM* обеспечивает согласованность данных и управляет документооборотом, гарантируя доставку информации адресату в самое подходящее для принятия решений время.

2.3.6. Сетевая работа над проектом

Производители систем *CAD/CAM/CAE* уверенно продвигаются в направлении интеграции своих систем с Интернетом. Встроенные возможности работы с *Internet* уже начинают появляться в таких системах. Используя *Internet* в качестве расширения системы геометрического моделирования, команда, работающая над проектом, может при помощи браузеров и подключаемых модулей просматривать модели и работать с ними уже на ранних стадиях процесса разработки.

Одним из способов распространения геометрических данных в *Internet* является сохранение их на *Web*-странице в формате, доступном для чтения. Если пользователь введет в адресной строке браузера универсальный локатор ресурса страницы с моделью (*URL*), то увидит изображение геометрической модели.

Производители систем геометрического моделирования поддерживают публикацию, включая в свои системы возможность экспорта геометрических данных в форматах, доступных для чтения браузерами. Одним из таких форматов является язык описания виртуальной реальности (*Virtual Reality Modeling Language – VRML*). Можно сохранить данные и в форматах типа *Computer Graphics Metafile (CGM)* или *Drawing Web Format (DWF)* фирмы *Autodesk*, что потребует от пользователя установки специального подключаемого модуля для браузера.

VRML – стандартный формат для описания трехмерных графических структур и сцен, тогда как *CGM* и *DWF* являются двумерными форматами. Хотя *VRML* имеет некоторые ограничения, в частности низкую точность и большие размеры файлов, большинство производителей решили поддержать возможность сохранения в формате *VRML*, поскольку этот формат является трехмерным. Некоторые пошли еще дальше, предоставив пользователям средства для встраивания *VRML* в *Web*-страницу с текстом и гиперссылками. Файлы *VRML*, так же как и гипертекстовые файлы, могут содержать ссылки на любую информацию в Сети, в том числе и на другие *VRML*-файлы. Пользователь, просматривающий такой файл, видит перед собой объемную модель трехмерного тела. Ссылки на *URL* идентифицируются по изменению положения курсора, точно так же, как и при работе с обычными *Web*-страницами. Щелчок мыши на ссылке приводит к загрузке файла, на который она указывает. Например, щелкая на разных частях модели, пользователь может открывать окна с информацией, предоставленной другими членами команды. Щелкнув в углу картинки, пользователь увидит технические характеристики модели и узнает, что о ней думают коллеги. Гиперссылка может указывать на каталог деталей, руководство по стандартам или на адрес электронной почты руководителя проекта.

Недостатком *VRML*-модели является то, что она, несмотря на полноценную трехмерную визуализацию, не содержит параметрической информации об изделии и не пригодна для редактирования, анализа и использования в *CAD/CAM/CAE*-системах. Для интерфейса между разными системами с сохранением предложен ряд так называемых «нейтральных» форматов: *IGES*, *STEP*, *STL*.

STEP (*STandard for the Exchange of Product model data*) – стандарт для обмена данными о модели изделия. В соответствии с названием стандарт *STEP* определяет нейтральный формат представления данных об изделии в виде информационной модели. Данные об изделии включают в себя: состав и конфигурацию изделия; геометрические модели разных типов; административные данные; специальные данные.

Для обеспечения возможности единообразного описания изделий в различных прикладных областях предполагается, что информационные модели (в терминах стандарта «прикладные протоколы» или «протоколы применения») создаются на базе типовых блоков («интегрированных ресурсов»), причем для описания схем данных используется специально введенный язык *Express*.

IGES (Initial Graphics Exchange Specification) – исходный стандарт обмена геометрическими данными, разработанный как нейтральный формат для передачи инженерной графики между различными системами типа САПР. В настоящее время наиболее распространен в качестве связи между гетерогенными *CAD*-системами.

STL (STereo Lithography) – формат для хранения и передачи данных о геометрии модели для изготовления методом быстрого прототипирования на установке лазерной стереолитографии.

Несмотря на то, что при экспорте в нейтральный формат часть параметрической информации теряется (например встроенные в модель зависимости) дальнейшее применение таких моделей не вызывает трудностей.

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Васильков, Ю. В.* Компьютерные технологии вычислений в математическом моделировании: учеб. пособие / Ю. В. Васильков, Н. Н. Василькова. – М.: Финансы и статистика, 2002. – 256 с. – ISBN 5-272-00069-2.
2. *Дьяконов, В. П.* Mathcad 8/2000: спец. справ. / В. П. Дьяконов. – СПб.: Питер, 2000. – 592 с. – ISBN 5-272-00069-2.
3. *Дьяконов, В. П.* Mathcad 8 Pro в математике, физике и Internet / В. П. Дьяконов, И. В. Абраменкова. – М.: Нолидж, 1999. – 512 с. – ISBN 5-89251-068-9.
4. *Журавлева, Т. Ю.* Основы информатики / Т. Ю. Журавлева, Ф. Б. Конев [и др.]. – М.: ЭКСИМ, 2000. – 288 с.
5. Информатика: Базовый курс / С. В. Симонович [и др.]. – СПб.: Питер, 2001. – 640 с. – ISBN 5-3046-0134-2.
6. Компьютерные технологии в науке и образовании : метод. указания к практ. занятиям / сост. В. Н. Арефьев. – Ульяновск, 2001. – 42 с.
7. *Конев, Ф. Б.* Информатика для инженеров : учеб. пособие / Ф. Б. Конев. – М.: Высш. шк., 2004. – 272 с. – ISBN 5-06-004871-3.
8. *Кудрявцев, Е. Ж.* Mathcad 8 / Е. Ж. Кудрявцев. – М.: ДМК, 2000. – 320 с. – ISBN 5-94074-006-5.
9. *Ли, К.* Основы САПР (CAD/CAM/CAE) / К. Ли. – СПб.: Питер, 2004. – 560 с. – ISBN 5-94723-770-9.
10. *Михеева, Е. В.* Практикум по информационным технологиям в профессиональной деятельности: учеб. пособие / Е. В. Михеева. – 2-е изд. – М.: Академия, 2004. – 256 с. – ISBN 5-7695-1494-9.
11. *Новикова, Е. А.* Основы математического моделирования: лаб. практикум / Е. А. Новикова; Владим. гос. ун-т. – Владимир, 2004. – 80 с. – ISBN 5-89368-468-0.
12. *Плис, А. М.* Mathcad. Математический практикум для инженеров и экономистов: учеб. пособие / А. М. Плис, Н. А. Сливина. – М.: Финансы и статистика, 2003. – 656 с. – ISBN 5-279-02550X.
13. Практикум по информатике / А. А. Землянский, Г. А. Кретьева [и др.]. – М.: КолоС, 2003. – 384 с. – ISBN 5-9532-0046-3.

14. *Фуфаев, Э. В.* Пакеты прикладных программ: учеб. пособие / Э. В. Фуфаев, Л. И. Фуфаева. – М.: Академия, 2004. – 352 с. – ISBN 5-7695-1215-6.

15. *Херхагер, М.* Mathcad 2000: полн. Рук. / М. Херхагер, Х. Партоллер. – Киев: ВНУ, 2000. – 416 с. – ISBN 966-552-032-6.

16. Mathcad 6.0 PLUS. Финансовые, инженерные и научные расчеты в среде Windows 95. – М.: Филинь, 1996. – 712 с. – ISBN 5-900855-35-X.

17. Microsoft Office PowerPoint 2003 для Windows / Р. Б. Альтман. – М.: ДМК Пресс, 2004. – 416 с. – ISBN 5-94074-249-1.

Образовательные ресурсы INTERNET

Поисковые системы

РОЛ/Поисковая система «АПОРТ» – <http://aport.ru/>

Яндекс – <http://yandex.ru/>

Google – <http://www.google.ru/>

Rambler: информационно-поисковая система – <http://www.rambler.ru/>

Полнотекстовые электронные библиотеки

Библиотека Ивановского государственного энергетического университета – <http://www.ispu.ru/portal/index.pl?id=2224&isa=Category&op=show>

Библиотека механико-технологического факультета ВлГУ – <http://space.vpti.vladimir.ru/library/library.htm>

Научная электронная библиотека – <http://elibrary.ru/>

Образовательный математический сайт Exponenta.ru – <http://exponenta.ru/>

Система дистанционного обучения СПбГУ ИТМО – <http://cde.ifmo.ru/>

Центр дистанционного обучения МГТУ им. Н.Э. Баумана – <http://www.cdl.bmstu.ru/>

Центральная библиотека образовательных ресурсов – <http://www.edulib.ru/>

Электронная библиотека полнотекстовых учебных и научных изданий УлГТУ – <http://venec.ulstu.ru/lib/>

OSP.RU: Издательство «Открытые системы» – <http://www.osp.ru/>

Информационные порталы

ГосНИИ ИТТ «Информика» – <http://www.informika.ru/>

ИНЖЕНЕР – Научно-образовательный портал МГТУ им.

Н.Э. Баумана – <http://engineer.bmstu.ru/>

Инженерное образование: Каталог ресурсов –
<http://techno.stack.net/db/catalog.html>

ИнфоНТР – Портал науки, технологий и промышленности –
<http://www.infontr.ru/>

Российское образование. Федеральный портал – <http://www.edu.ru/>

ЦИТ-Форум: Море (!) аналитической информации –
<http://www.citforum.ru/>

Патентная информация

Моспатент – Регистрация товарных знаков, программ для ЭВМ,
оформление патентов на изобретения... – <http://www.mospatent.ru/>

Роспатент – Федеральный институт промышленной собственности –
<http://www.fips.ru/>

Учебное издание

НОВИКОВА Елена Александровна

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В МАШИНОСТРОЕНИИ
И НАУЧНЫХ ИССЛЕДОВАНИЯХ

Учебное пособие

Редактор Е.А. Амирсейидова

Корректор Е.В. Афанасьева

Компьютерная верстка Е.Г. Радченко

Дизайн обложки А.О. Седова

ЛР № 020275. Подписано в печать 30.05.05.

Формат 60x84/16. Бумага для множит. техники. Гарнитура Таймс.
Печать на ризографе. Усл. печ. л. 6,28. Уч.-изд. л. 6,67. Тираж 300 экз.

Заказ

Редакционно-издательский комплекс
Владимирского государственного университета.
600000, Владимир, ул. Горького, 87.