

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	
1 АНАЛИЗ СОВРЕМЕННЫХ ПРОГРАММНЫХ СРЕДСТВ ЭВМ.....	
1.1 Информатика и вычислительная техника.....	
1.1.1 Задание на лабораторную работу № 1.....	
1.1.2 Задание на лабораторную работу № 2.....	
1.1.3 Задание на лабораторную работу № 3.....	
1.2 Прикладные программные средства.....	
1.2.1 Текстовые процессоры. Приложение Microsoft Word.....	
1.2.2 Табличные процессоры. Процессор Microsoft Excel.....	
1.2.3 Microsoft Power Point.....	
1.2.4 Задание на лабораторную работу № 4.....	
1.2.5 Задание на лабораторную работу № 5.....	
1.2.6 Задание на лабораторную работу № 6.....	
1.3 Компьютерные сети и коммуникации. Элементы Internet-технологий.....	
1.3.1 Понятие компьютерных сетей.....	
1.3.2 Основы Internet-технологий.....	
1.3.3 Адресация и виды информации в Internet.....	
1.3.4 Задание на лабораторную работу № 7.....	
1.3.5 Задание на лабораторную работу № 8.....	
1.3.6 Задание на лабораторную работу № 9.....	
1.4 Инструментальные программные средства.....	
2 ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ СОВРЕМЕННЫЕ ПРОГРАММНЫЕ СРЕДСТВА ЭВМ.....	
2.1 Matlab как среда моделирования и программирования высокого уровня.....	
2.1.1 Задание на лабораторную работу № 10.....	
2.1.2 Задание на лабораторную работу № 11.....	
2.1.3 Задание на лабораторную работу № 12.....	
2.1.4 Задание на лабораторную работу № 13.....	
2.1.5 Задание на лабораторную работу № 14.....	
2.1.6 Задание на лабораторную работу № 15.....	
2.1.7 Задание на лабораторную работу № 16.....	
2.1.8 Задание на лабораторную работу № 17.....	
2.1.9 Задание на лабораторную работу № 18.....	
2.1.10 Задание на лабораторную работу № 19.....	
2.1.11 Задание на лабораторную работу № 20.....	
2.1.12 Задание на лабораторную работу № 21.....	
2.1.13 Задание на лабораторную работу № 22.....	
2.1.14 Задание на лабораторную работу № 23.....	
2.1.15 Задание на самостоятельную работу № 1.....	
2.1.16 Задание на самостоятельную работу № 2.....	

2.1.17	Задание на самостоятельную работу № 3	.....
2.2	Mathcad – универсальная вычислительная система	.....
2.2.1	Задание на лабораторную работу № 24	.....
2.2.2	Задание на лабораторную работу № 25	.....
2.2.3	Задание на лабораторную работу № 26	.....
2.2.4	Задание на лабораторную работу № 27	.....
2.2.5	Задание на лабораторную работу № 28	.....
2.2.6	Задание на лабораторную работу № 29	.....
2.2.7	Задание на лабораторную работу № 30	.....
2.2.8	Задание на лабораторную работу № 31	.....
2.2.9	Задание на лабораторную работу № 32	.....
2.2.10	Задание на лабораторную работу № 33	.....
2.2.11	Задание на самостоятельную работу № 4	.....
2.2.12	Задание на самостоятельную работу № 5	.....
2.3	Моделирование структурных схем (VisSim)	.....
2.4	Анализ электронных и электрических схем (Pspice)	.....
2.5	Моделирование систем управления (Model vision studium)	....
2.5.1	Задание на самостоятельную работу №6	.....
2.6	Создание виртуальных панелей управления (LabView)	.....
2.7	Интегрированная среда разработки ATMEL AVR Studio	.....
2.8	Общие понятия и структура SCADA-систем	.....
2.9	Сетевые технологии в электроприводе	.....
<b>3 СТРУКТУРНЫЙ АНАЛИЗ И МОДЕЛЬ ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ СИСТЕМ</b>		
3.1	Понятие и особенности системного анализа	.....
3.2	Блочный-иерархический подход к проектированию систем	.....
3.3	Аспекты объектно-ориентированного анализа и проектирования систем	.....
3.3.1	Диаграмма переходов состояний	.....
3.3.2	SADT – технология	.....
3.3.3	Диаграммы потоков данных	.....
3.3.4	Задание на лабораторную работу № 34	.....
3.3.5	Задание на лабораторную работу № 35	.....
3.3.6	Задание на лабораторную работу № 36	.....
3.3.7	Задание на лабораторную работу № 37	.....
3.3.8	Задание на лабораторную работу № 38	.....
3.3.9	Задание на лабораторную работу №7	.....
<b>ЗАКЛЮЧЕНИЕ</b>		
<b>СПИСОК ЛИТЕРАТУРЫ</b>		
<b>ПРИЛОЖЕНИЕ А. Требования к оформлению лабораторных работ</b>		

## ВВЕДЕНИЕ

Курс «Современные программные средства в электроприводе» посвящен эффективному применению современных компьютерных технологий в проектировании, создании и эксплуатации электроприводов.

Инженерам и проектировщикам электромеханических систем часто необходимо решать сложные прикладные вычислительные задачи, большинство из которых решаются не аналитическими, а так называемыми численными методами, реализация которых на алгоритмических языках высокого уровня требует от пользователя высокой квалификации, глубокого понимания сущности методов на стадии алгоритмизации, программирования и отладки. Кроме того, подобные задачи можно успешно решать с помощью современных компьютерных программных средств.

Компьютерные технологии в электроприводе интенсивно развиваются за счет развития самого электропривода, совершенствования компьютерной техники и программного обеспечения и все более широкого применения цифровых технологий [1].

Компьютерные технологии в электроприводе можно разделить на несколько групп по применению:

- а) анализ и синтез электропривода (моделирование);
- б) проектирование электронной части ЭП и печатных плат;
- в) программирование и отладка цифровых систем управления (МК, ПЛИС);
- г) построение распределенных систем, использование виртуальных панелей управления и контроля.

Моделирование широко и давно применяется для анализа и синтеза электропривода. Первые модели были физические, затем – на основе аналоговых элементов, сейчас им на смену пришло компьютерное моделирование. Существует большое количество моделирующих программ, которые можно разделить на следующие группы:

- а) математического моделирования;
- б) моделирования структурных схем по передаточным функциям;
- в) моделирования электронных схем;
- г) моделирования физических процессов, полей и т. п.

Программы математического моделирования применяются на этапе первичного анализа и позволяют определить некоторые численные характеристики разрабатываемой системы.

Программы моделирования структурных схем позволяют оценить реакцию системы на различные управляющие и возмущающие воздействия, провести анализ переходных процессов, а также провести частотный анализ. Используя такие программы, оценивается правильность выбора структурной схемы и корректирующих звеньев.

Программы моделирования электронных схем применяются для анализа и оптимизации электронной части электропривода с учетом

характеристик конкретных используемых элементов, вплоть до анализа помехозащищенности и тепловых режимов работы.

Программы моделирования полей позволяют рассчитывать электрические, магнитные и тепловые поля различных систем.

Пакеты программ для проектирования электронных схем позволяют собирать из элементов принципиальные схемы электронных узлов электропривода, проводить их оптимальное размещение в заданных габаритах печатной платы, проводить автоматическую трассировку платы, подготавливать проектную документацию для изготовления печатных плат.

Создание цифровых систем управления электроприводом также невозможно без использования компьютерных технологий. Все управляющие сигналы, а также данные с датчиков обрабатываются микроконтроллером или программируемой логикой. При этом проектировщик должен разработать и отладить управляющую программу для микроконтроллера. Хотя языки программирования различных микроконтроллеров очень похожи, каждая фирма-производитель имеет свое программное обеспечение для работы со своими микроконтроллерами. Многие из них включают в пакет программ не только компилятор для перевода программы, написанной разработчиком в исполняемый код микроконтроллера, но и средства отладки и эмуляторы микроконтроллеров.

Относительно новым блоком компьютерных технологий в электроприводе является построение распределенных систем и использование виртуальных панелей управления и контроля. Эти технологии позволяют дистанционно анализировать работу удаленных электроприводов и управлять ими с единого диспетчерского пульта.

В пособии будут рассмотрены наиболее распространенные и широко применяемые пакеты программ.

# 1 АНАЛИЗ СОВРЕМЕННЫХ ПРОГРАММНЫХ СРЕДСТВ ЭВМ

## 1.1 Информатика и вычислительная техника

Информатика – новая информационная индустрия, связанная с использованием персональных компьютеров и сети Интернет. В наступившем новом тысячелетии большая часть информации, связанной с деятельностью людей, будет храниться в памяти ЭВМ [2].

ЭВМ – электронные вычислительные машины – одно из важнейших изобретений XX века, получившее название компьютер. Компьютеры используются как универсальные устройства обработки, передачи и накопления самой различной информации.

Новым инструментом в передаче и накоплении информации в человеческом обществе стала сеть Интернет.

Интернет – это международная компьютерная сеть, связывающая компьютеры во всех странах и континентах, хранящая гигантские объемы информации и дающая оперативный доступ к этой информации практически всем людям.

Компьютеры первого поколения создавались как электронные вычислительные машины для автоматизации сложнейших вычислений. Объем и сложность вычислений, выполнявшихся первыми компьютерами, были недоступны даже самым сильным математикам и вычислителям, но посильными для современных персональных компьютеров.

Компьютеры второго поколения созданы в качестве универсальных машин для решения задач обработки и накопления информации с использованием различных устройств ввода и вывода [2]. Для компьютеров этого поколения появились первые накопители информации на магнитных лентах. Для этих машин были созданы первые операционные системы, первые системы программирования, первые системы искусственного интеллекта и первые вычислительные комплексы.

Компьютеры третьего поколения были созданы как серийные вычислительные комплексы для накопления и обработки служебной информации. На базе этих ЭВМ были созданы различные банки данных, системы автоматизации проектирования, производства, обучения и многие другие автоматизированные системы.

Четвертое поколение – это компьютеры, создаваемые на базе микропроцессоров массовых серий. С четвертого поколения ЭВМ началось массовое производство персональных компьютеров – малогабаритных ЭВМ, снабжаемых дисплеями и накопителями информации на магнитных дисках.

Персональные компьютеры – это малогабаритные вычислительные машины, которые могут быть установлены на любом рабочем месте. Наиболее известны и распространены персональные компьютеры IBM PC

и Macintosh, типовая структура которых представлена на рисунке 1.1.

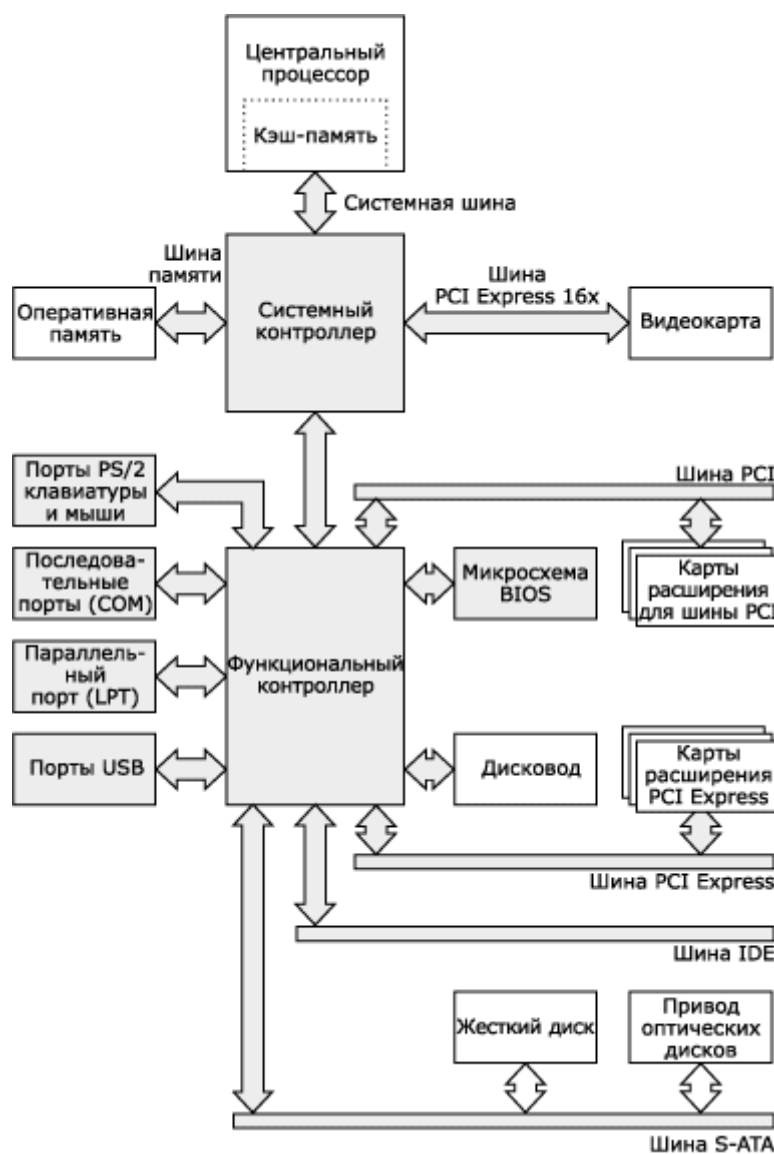


Рисунок 1.1 – Типовая структура персонального компьютера

Минимальный состав персональных компьютеров:

- а) системный блок;
- б) дисплей;
- в) клавиатура.

Дисплей – это устройство отображения информации на экране.

Клавиатура – устройство для ввода чисел, слова, команд управления компьютером.

Мышь – манипулятор для ввода информации в компьютер.

Системный блок содержит процессор и оперативную память. Возможности компьютеров зависят от типа и быстродействия процессора, а также от объемов оперативной и долговременной памяти.

Процессор – это устройство управления компьютером.

Оперативная память – память для временного хранения данных в

компьютере, используется только, когда компьютер работает. От объема и скорости оперативной памяти зависит быстродействие компьютера.

Быстродействие компьютеров определяется числом операций, выполняемых процессором за одну секунду. Основной функцией процессоров является автоматическое управление работой ЭВМ с помощью программ, размещаемых в оперативной памяти.

Минимальной единицей информации считается бит [3]. Бит – это величина, принимающая значение 0 или 1. Любая другая информация может быть закодирована последовательностью из нулей и единиц. В подобном виде вся информация представляется в памяти ЭВМ.

Единицей памяти в современных ЭВМ считается байт. Байты – это 8-разрядные двоичные числа вида — 00000000, 00000001, ..., 11111111. Один байт записывается в виде 8 двоичных знаков информации — нулей и единиц:

$$1 \text{ байт} = 8 \text{ бит.}$$

Для измерения памяти большого объема используются следующие единицы:

$$1 \text{ Кбайт} = 1024 \text{ байт (1 килобайт);}$$

$$1 \text{ Мбайт} = 1024 \text{ Кбайт (1 мегабайт);}$$

$$1 \text{ Гбайт} = 1024 \text{ Мбайт (1 гигабайт).}$$

Скорость передачи информации по линиям связи оценивается в бодах и килободах. Скорость в один бод — это передача одного бита в секунду:

$$1 \text{ бод} = 1 \text{ бит/секунда;}$$

$$1 \text{ Кбод} = 1024 \text{ бод.}$$

Жесткий диск – служит для длительного хранения информации, на нем расположены программы необходимые для работы компьютера и файлы пользователя.

Видеокарта – плата внутри системного блока, предназначенная для связи системного блока и монитора, передает изображение на монитор и берет часть вычислений на себя по подготовке изображения для монитора. От видеокарты зависит качество изображения. Видеокарта имеет свою встроенную оперативную память и свой процессор по обработке изображения. Чем выше частота работы процессора видеокарты и чем больше память видеокарты, тем в более крутые (позже выпущенные) игры вы сможете играть на своем компьютере.

Звуковая карта – предназначена для подготовки звуковых сигналов, воспроизводимых колонками. Звуковая карта обычно встроена в материнскую плату, но бывает и конструктивно отделена и подключена через шину.

Сетевая карта – плата, устройство, устанавливается в материнскую плату или встроено в нее. Сетевая карта служит для соединения

компьютера с другими компьютерами по локальной сети или для подключения к сети Интернет.

CD/DVD-ROM – устройство для чтения/записи компакт-дисков, CD-дисков, DVD-дисков. Эти устройства отличаются скоростью считывания или записи информации, а также возможностью чтения/записи различных носителей.

Картридер – устройство для чтения/записи информации на карты памяти. Картридеры отличаются по скоростным характеристикам чтения/записи информации. Картридеры бывают встроенными в системный блок или конструктивно независимые, подключаемые к системному блоку через USB-порт.

Периферийные устройства - это устройства, которые подключаются к контроллерам ПК и расширяют его функциональные возможности. По назначению дополнительные устройства разделяются на:

- а) устройства ввода (трэкболлы, джойстики, световые перья, сканеры, цифровые камеры, диджитайзеры);
- б) устройства вывода (плоттеры или графопостроители);
- в) устройства хранения (стримеры, zip - накопители, магнито-оптические накопители, накопители HiFD и др.);
- г) устройства обмена (модемы).

### 1.1.1 Задание на лабораторную работу № 1

**Тема:** Архитектура персонального компьютера.

**Цель работы:** изучить устройство персонального компьютера, приобрести навыки в исследовании и описании аппаратного и программного обеспечения ЭВМ.

#### ***Теоретические сведения***

Для получения информации об аппаратном обеспечении ПЭВМ могут использоваться такие программы как msinfo32.exe («Сведения о системе»), mmc.exe («Консоль управления») с оснасткой "Диспетчер устройств" и др.

Программы msinfo32.exe и mmc.exe входят в состав операционной системы Windows, их следует запускать из командной строки.

На составленной структурной схеме компьютера следует указать максимально полную информацию об особенностях элементной базы исследуемого компьютера – название ЦПУ, микросхемы чипсета, микросхемы контроллеров шин, прерывания, каналы DMA, порты ввода-вывода, типы и характеристики запоминающих устройств и прочее.



## ***Индивидуальное задание***

1. С помощью системного программного обеспечения исследовать состав аппаратного обеспечения ПЭВМ, используемой для выполнения лабораторной работы (домашнего персонального компьютера);

2. Составить детализированную структурную схему исследованной ПЭВМ, используя векторный графический редактор.

### **1.1.2 Задание на лабораторную работу № 2**

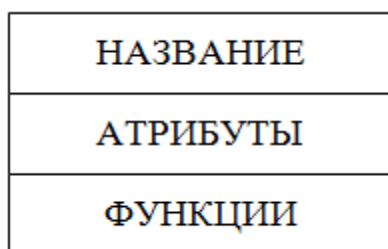
**Тема:** Классификация программного обеспечения.

**Цель работы:** изучить системы классификации и основные классы программного обеспечения (ПО), освоить применение графического языка UML.

### ***Теоретические сведения***

Способ поиска установленного ПО следует выбрать самостоятельно. Это может быть последовательный просмотр содержимого главного меню и рабочего стола операционной системы Windows, просмотр содержимого каталогов файловой системы, использование специализированных программ установки ПО и др.

При создании диаграммы классов программного обеспечения необходимо использовать следующие условные обозначения. Класс изображается в виде прямоугольника с тремя секциями (рисунок 1.2).



*Рисунок 1.2 – Класс программного обеспечения*

В верхней секции указывается название ПО или тип ПО, в средней секции указываются атрибуты - характеристики ПО, в нижней секции указываются функции ПО или выполняемые им операции.

Взаимосвязи между классами показывайте на диаграмме с помощью линий связи двух видов:

а) агрегация – указывает отношение вида «часть»–«целое». Например, CorelDraw – это часть пакета программ Corel Graphics Suite 11.

Графически связь изображается в виде линии с ромбом со стороны агрегатора (целого);

б) наследование – указывает отношение вида «частное»-«общее». Например, CorelDraw – частный случай векторного графического редактора, и имеет типичные для таких редакторов возможности.

Графически эта связь изображается сплошной соединительной линией со стрелкой (треугольником) со стороны более общего класса.

Пример выполнения диаграммы классов показан на рисунке 1.3.

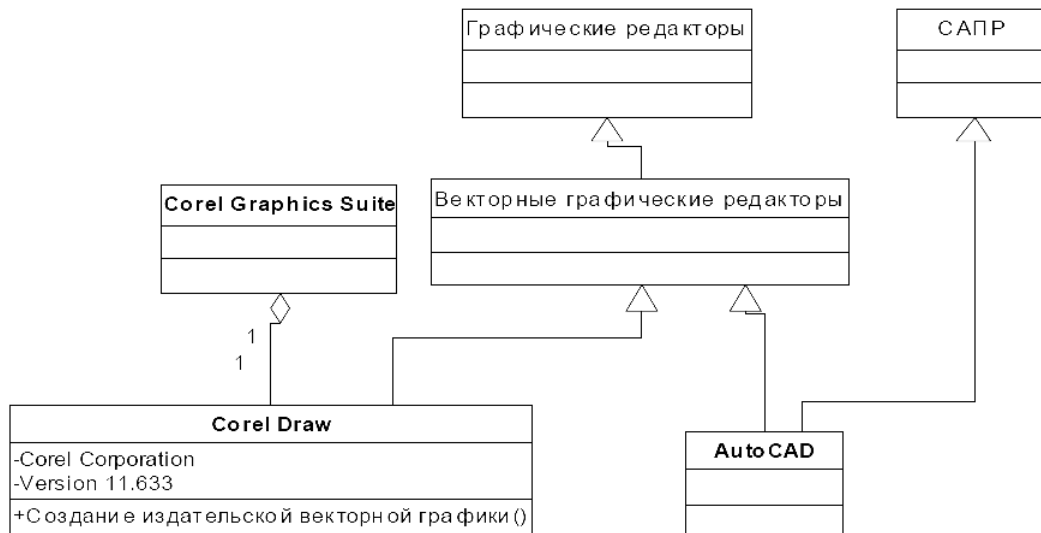


Рисунок 1.3 – Диаграмма классов программного обеспечения

### ***Индивидуальное задание***

1 Составить список ПО, установленного на рабочей станции или на домашнем персональном компьютере.

2 Для каждого элемента составленного списка указать следующие характеристики:

- а) полное название программы и/или пакета программ;
- б) номер версии и дату выпуска;
- в) имя автора и/или название фирмы-производителя;
- г) тип ПО (свободно распространяемое, условно бесплатное, коммерческое и т.п.);
- д) функциональное назначение ПО (текстовый процессор, векторный графический редактор, САПР и т.п.).

3 Обобщить собранные сведения, выделив классы программного обеспечения по функциональному назначению или иным признакам. Результаты обобщения представить в виде диаграммы классов на графическом языке UML (UML Static Structure Diagram).

### 1.1.3 Задание на лабораторную работу № 3

**Тема:** Применение компьютерной техники в технических устройствах.

**Цель работы:** приобрести навыки в описании алгоритмов работы автоматически действующих промышленных и бытовых устройств, управляемых компьютером, обосновать роль компьютера в работе устройств.

#### *Теоретические сведения*

В качестве примеров описания работы автоматического устройства можно использовать краткое описание кофейного автомата [4].

Кофейный автомат служит для изготовления и продажи горячих напитков. В прорезь опускается произвольное количество купюр и монет разного достоинства. Автомат содержит несколько групп кнопок:

- а) тип напитка;
- б) наличие/отсутствие сахара и степень его наличия в напитке;
- в) наличие/отсутствие молока.

Нажимая кнопки в каждой группе, пользователь выбирает «конфигурацию» напитка, а кнопка «Налить» запускает процесс. Автомат не только наливает напиток, но и выдает сдачу. Работу устройства контролирует Flash-микропроцессор, совместимый со всеми протоколами (Executive, MDB/ICP) с возможностью программирования через PC с Microsoft Windows.

#### *Индивидуальное задание*

В произвольной форме описать назначение, основные функции, порядок или алгоритм работы какого-либо автоматически действующего промышленного или бытового устройства, управляемого компьютером, отразив в этом описании роль компьютера в работе устройства.

Указать и обосновать необходимые свойства данного компьютера, например, высокое быстродействие, параллельность структуры, низкая стоимость, низкое энергопотребление, малые габариты и масса, повышенная надежность, высокая степень защищенности от каких-либо внешних воздействий, развитые средства взаимодействия с другими системами, например, сетевые и т.п.

При необходимости в описание следует включить схемы, диаграммы и другие текстовые и графические материалы, поясняющие принцип работы устройства, основные выполняемые им операции.

Созданное описание должно быть достаточно детальным для построения компьютерной модели работы автоматического устройства (в виде графа, например).

## *Контрольные вопросы*

1. Что такое информация?
2. Что такое информатика?
3. Что такое ЭВМ?
4. Какие устройства входят в состав персональных компьютеров?
5. Что такое процессор?
6. Что такое ОЗУ, ПЗУ, АЛУ?
7. Каково быстродействие современных процессоров?
8. В каких единицах измеряется объем памяти компьютеров?
9. Каков объем оперативной памяти современных компьютеров?
10. Каковы объемы памяти на жестких дисках?
11. Каковы объемы памяти на компакт-дисках?
12. Каковы основные виды принтеров?
13. Что такое сканер, многофункциональное устройство?
14. Что такое программное обеспечение?
15. Классификация программного обеспечения?
16. Примеры классов программного обеспечения?

### *1.2 Прикладные программные средства*

Прикладное программное обеспечение или приложение – это программа, предназначенная для выполнения определенных пользовательских задач и рассчитанная на непосредственное взаимодействие с пользователем.

В большинстве операционных систем прикладные программы не могут обращаться к ресурсам компьютера напрямую, а взаимодействуют с оборудованием и проч. посредством операционной системы.

Большое многообразие прикладного программного обеспечения можно классифицировать по типу:

а) программные средства общего назначения (тестовые редакторы, системы компьютерной верстки, графические редакторы, СУБД);

б) программные средства специального назначения (экспертные системы, мультимедиа приложения, гипертекстовые системы, системы управления содержанием);

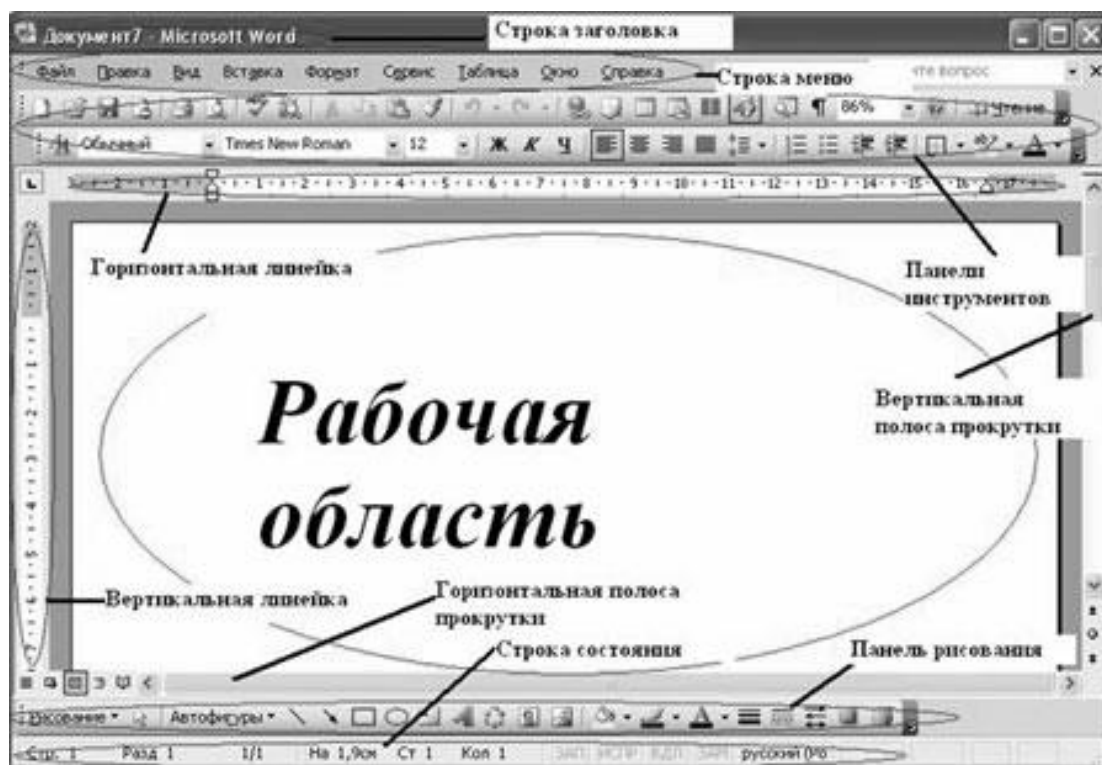
в) программные средства профессионального уровня (САПР, АРМ, АСУТП).

#### 1.2.1 Текстовые процессоры. Приложение Microsoft Word

Текстовый процессор – вид прикладной компьютерной программы, предназначенной для производства (включая набор, редактирование, форматирование, иногда печать) любого вида печатной информации [3].

Иногда текстовый процессор называют текстовым редактором второго рода.

Текстовые процессоры, в отличие от текстовых редакторов, имеют больше возможностей для форматирования текста, внедрения в него графики, формул, таблиц и других объектов. Поэтому они могут быть использованы не только для набора текстов, но и для создания различного рода документов, в том числе официальных. Наиболее известным примером текстового процессора является Microsoft Word, вид окна которого представлен на рисунке 1.4.



*Рисунок 1.4 – Общий вид окна Microsoft Word*

Microsoft Word – текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов.

Выпускается корпорацией Microsoft в составе пакета Microsoft Office. Первая версия была написана Ричардом Броди (Richard Brodie) для IBM PC, использующих DOS, в 1983 году. Позднее выпускались версии для Apple Macintosh (1984), SCO UNIX и Microsoft Windows (1989).

Microsoft Word является наиболее популярным из используемых в данный момент текстовых процессоров, что сделало его бинарный формат документа стандартом де-факто, и многие конкурирующие программы имеют поддержку совместимости с данным форматом.

Как и прочие приложения из Microsoft Office, Word может расширять свои возможности посредством использования встроенного макроязыка (сначала использовался WordBasic, сейчас применяется VBA – Visual Basic для приложений).

### 1.2.2 Табличные процессоры. Процессор Microsoft Excel

Решение многих задач исследовательского характера на языках высокого уровня с использованием всего арсенала приемов и методов профессионального программирования – сложное и громоздкое дело. Понадобился принципиально иной подход, который был воплощен в виде электронных таблиц - инструмента, доступного непрофессионалам [5].

В настоящее время известно много вариантов электронных таблиц: Multiplan, SuperCalc, QuattroPro, Excel, Lotus 1-2-3 и др. Принципиально все они представляют табличный процессор и отличаются лишь интерфейсом и сервисными возможностями.

Электронная таблица – это прямоугольная матрица, состоящая из ячеек, каждая из которых имеет свой номер.

Данные, входящие в таблицы, можно автоматически представлять в виде графиков, диаграмм, гистограмм и т.д.

Пользователь работает в диалоге со специальной программой, позволяющей заполнять ячейки нужным ему содержимым (текстами, числами или формулами для расчетов); очищать их, копировать и удалять, сортировать (т.е. располагать клетки, а также строки и столбцы из них, в определенном порядке); производить вычисления над всей таблицей или ее частью и т.д.

Одним из наиболее популярных табличных процессоров является Microsoft Excel, общий вид окна которого представлен на рисунке 1.5.

Основные возможности электронных таблиц:

- а) автоматический перерасчет калькуляций;
- б) хранение и поиск калькуляций в памяти ЭВМ;
- в) вывод калькуляций на печать;
- г) обновление и ввод новых калькуляций.

Excel имеет два окна [5] – программное (внешнее) и рабочее (внутреннее). Внутреннее окно Sheet # содержит рабочую страницу (таких страниц несколько, они образуют книгу), представляющую двумерную прямоугольную таблицу. Справа и внизу на рабочей странице расположены линейки со стрелками прокрутки, позволяющие с помощью мыши быстро перемещаться по странице.

Кнопки линейки инструментов позволяют быстро и легко вызывать различные функции Excel. Их можно вызывать также через меню.

Ввод новых таблиц, состоящих из надписей, числовых данных и формул, проводится по ячейкам. Для этого к необходимой ячейке подводится курсор с помощью мышки или клавиш-стрелок, а затем

нажимается клавиша Enter на клавиатуре либо клавиша на мышке [2].

Копирование и перенос надписей, данных, формул и целых блоков позволяет достаточно быстро создавать новые таблицы из уже имеющихся в памяти ЭВМ. Более того, создание новых таблиц обычно осуществляется копированием уже существующих и обновлением в них надписей и данных.

При редактировании таблицы допускается изменение размеров строк или столбцов, а также подбор шрифтов и размера символов для надписей и значений данных, делающие таблицы более наглядными и красиво расположенными на экране или бумаге.

Для наглядного представления цифровых данных в электронных таблицах обычно предлагается целый набор цветных диаграмм в форме графиков и кругов, а также плоских и объемных гистограмм

Для отображения числовых данных в графической форме используют линии, полосы, столбцы, сектора и другие маркеры, а также их объединенные вариации.

В арсенале Microsoft Excel имеется большая встроенная библиотека построения графических образов: графиков, диаграмм, гистограмм, а также мощный встроенный графический редактор.

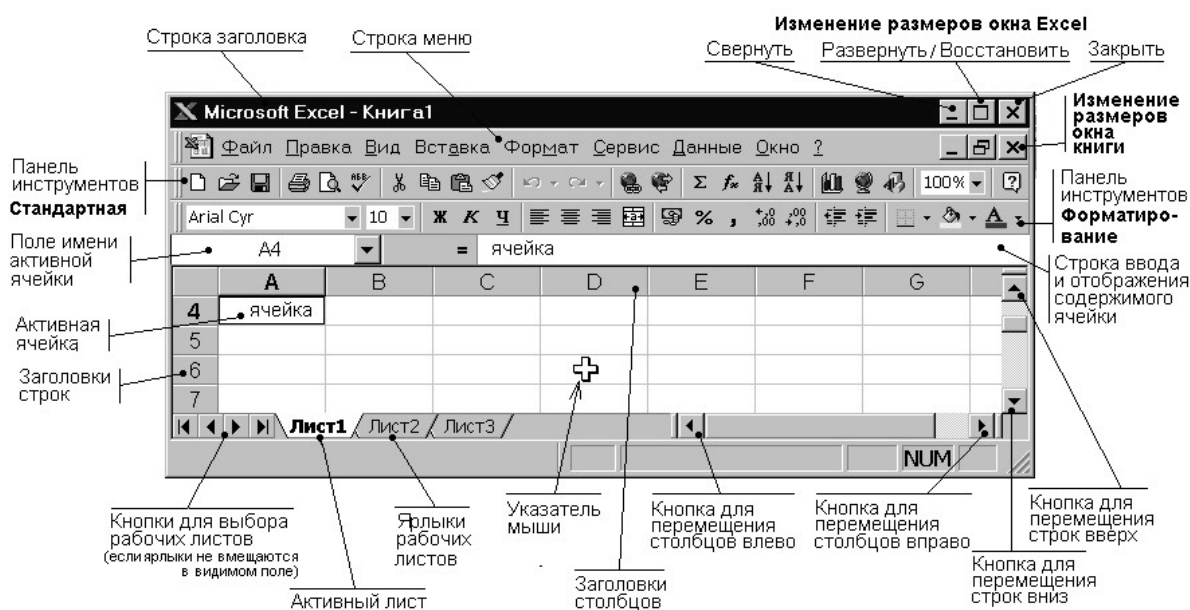
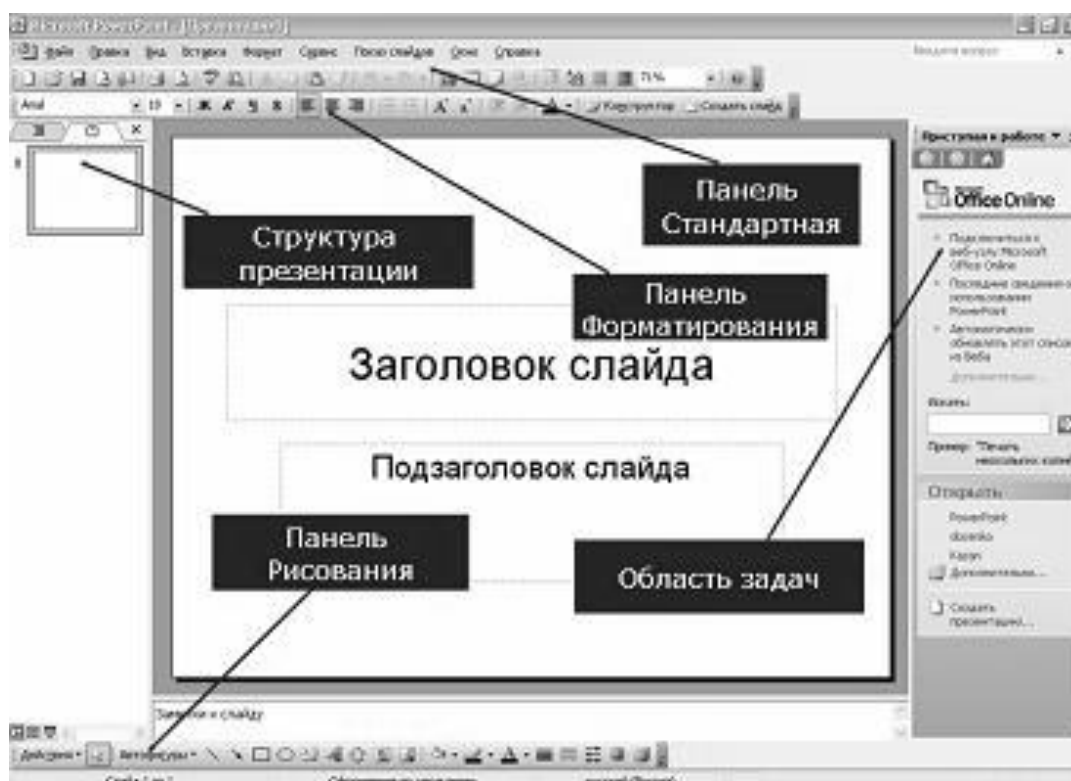


Рисунок 1.5 – Общий вид окна Microsoft Excel

### 1.2.3 Microsoft Power Point

Power Point представляет собой программу [3] для создания презентаций, которые необходимы во многих сферах профессиональной деятельности (в частности в бизнесе или в науке).

С помощью программы Power Point, общий вид окна которой приведен на рисунке 1.6, создание презентаций становится простым и увлекательным делом. Если презентация проводится для небольшой аудитории в 4-8 человек, то в этом случае достаточно иметь компьютер с обычным монитором. С другой стороны, при большом количестве зрителей необходимо иметь специальную аппаратуру, например, проектор, который подключается к компьютеру и проецирует имеющееся на мониторе изображение на обычный экран. Презентации могут также проводиться для пользователей локальной сети или Internet.



*Рисунок 1.6 – Общий вид окна Microsoft Power Point*

Среди возможностей Power Point по созданию презентаций, представляющих собой набор слайдов, можно отметить следующее:

а) управление процессом проведения презентации, т.е. отображением слайдов, которые в ней имеются;

б) управление переходами между слайдами, т.е. определение порядка отображения слайдов презентации во время ее показа;

в) установка параметров внешнего вида, отображения и появления слайдов;

г) работа с текстом, таблицами, графикой, анимацией, звуком, видео, а также объектами Word, Excel и Internet.

Любая презентация имеет следующие основные свойства и характеристики:

а) набор слайдов и их параметры;



б) содержание слайдов, которое, помимо пользователя, может также создаваться с помощью имеющихся мастеров автосодержания;

в) параметры рабочей области, т.е. ее размер, ориентация и т.д.

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

а) размер слайда;

б) шаблон оформления, т.е. параметры цветовой схемы, фона, шрифтов и т.д.;

в) разметка слайда, которая включает большой набор стандартных примеров размещения информации на слайде: расположение заголовка, рисунков, таблиц, надписей и т.д.;

г) эффект перехода, представляющий собой тот или иной режим появления и «исчезания» слайда – по нажатию кнопки мыши или автоматически через заданное время, с анимационными или звуковыми эффектами и т.д.

В Power Point существует несколько режимов просмотра презентации. Для выбора того или иного режима можно воспользоваться панелью Вид, кнопки которой расположены в нижнем правом углу окна Power Point, или командами меню «Вид»:

а) обычный режим представляет собой сочетание режимов структуры и слайдов, в котором также отображается область редактирования заметок к слайдам;

б) режим структуры отображает иерархическую структуру, содержащую заголовки и тексты слайдов презентации;

в) режим слайдов дает возможность просмотра каждого слайда в отдельности, при этом также имеется доступ к структуре слайда;

г) режим сортировщика слайдов отображает весь набор имеющихся в презентации слайдов.

Показ слайдов используется для предварительного просмотра созданной презентации, при этом не экране последовательно отображаются все слайды презентации с использованием всех включенных в них звуковых, видео и анимационных эффектов.

#### 1.2.4 Задание на лабораторную работу № 4

**Тема:** Основы обработки экспериментальных данных в среде пакета Excel.

**Цель работы:** приобрести навыки обработки данных, построения графиков, поиска математических зависимостей в среде приложения Microsoft Excel.

## *Теоретические сведения*

При обработке экспериментальных данных перед исследователем часто возникает необходимость установления зависимости между аргументом и функцией, когда эта связь задана табличными данными  $(x_i, y_i)$ .

На практике часто используются табличные данные для приближенного вычисления  $y$  при любом значении аргумента  $x$  (из некоторой области). Для этого применяется задача  $f(x)$  требуется приближенно заменить некой функцией  $g(x)$  так, чтобы отклонение  $g(x)$  от  $f(x)$  в заданной области было бы наименьшим. Функция  $g(x)$  при этом называется аппроксимирующей.

Пусть есть набор значений двух переменных  $x_i, y_i, i=1, \dots, n$ . Эти пары  $(x_i, y_i)$  можно отобразить точками на координатной плоскости. Такое графическое отображение называют корреляционным полем.

Пусть задачей является подбор функции  $Y=f(X)$ , «наилучшим» способом описывающей зависимость  $Y$  от  $X$ , если такая зависимость существует.

На основании построенного корреляционного поля можно предположить, что между  $x$  и  $y$  существует определенная зависимость  $y=f(x)$ . Функция  $f(x)$  может быть линейная, логарифмическая, полиномиальная, степенная или экспоненциальная. Необходимо определить вид функции  $f(x)$ .

В пакете Microsoft Excel заложены возможности построения диаграммы и определения основных зависимостей при помощи добавления линии тренда на диаграмму и расчета величины достоверности аппроксимации ( $R^2$ ).

Значение  $R$ -квадрат - число от 0 до 1, которое отражает близость значений линии тренда к фактическим данным. Линия тренда наиболее соответствует действительности, когда значение  $R$ -квадрат близко к 1. При аппроксимации данных с помощью линии тренда значение  $R$ -квадрат рассчитывается автоматически и полученный результат можно вывести на диаграмме.

Линия тренда - графическое представление направления изменения ряда данных. Например, повышающаяся линия может обозначать увеличение продаж за определенное количество месяцев. Линии тренда используются для анализа ошибок предсказания, что также называется регрессионным анализом.

Существует пять различных видов линий тренда, которые могут быть добавлены на диаграмму Microsoft Excel.

Линейная аппроксимация – это аппроксимация («описание») набора данных с помощью уравнения прямой. Она применяется в самых простых случаях, когда точки данных расположены близко к прямой. Линейная аппроксимация хороша для величины, которая увеличивается или убывает с постоянной скоростью. Уравнение имеет вид:  $y(x) = ax + b$

Логарифмическая аппроксимация используется для описания величины, которая вначале быстро растет или убывает, а затем постепенно стабилизируется. Логарифмическая аппроксимация использует как отрицательные, так и положительные величины. Уравнение имеет вид:  $y(x) = a \ln x + b$

Полиномиальная аппроксимация используется для описания величин, попеременно возрастающих и убывающих. Она полезна, например, для анализа большого набора данных о нестабильной величине. Уравнение полинома второй степени имеет вид:  $y(x) = a_2x^2 + a_1x + a_0$ .

Степенная аппроксимация используется для описания монотонно возрастающей либо монотонно убывающей величины, например расстояния, пройденного разгоняющимся автомобилем. Использование степенной аппроксимации невозможно, если данные содержат нулевые или отрицательные значения. Уравнение имеет вид:  $y(x) = ax^b$ .

Экспоненциальная аппроксимация используется в том случае, если скорость изменения данных непрерывно возрастает. Однако для данных, которые содержат нулевые или отрицательные значения, этот вид приближения неприменим. Уравнение имеет вид:  $y(x) = ae^{bx}$ .

### ***Ход работы***

- 1 Загрузить пакет Excel, ознакомиться с его интерфейсом.
- 2 Для построения корреляционного поля и ввести исходные данные – значения  $x$  и  $y$ .
- 3 Построить точечную диаграмму, используя меню «Вставка» → «Диаграмма» → вкладка «Стандартные» → тип «Точечная» (рисунок 1.7).
- 4 После построения диаграммы выполним следующие пункты меню «Диаграмма» → «Добавить линию тренда» → «Тип» → «Линейная» → вкладка «Параметры» → поставить отметку в пункте «Показать уравнение на диаграмме».
- 5 Полученные результаты оформить в виде таблицы.

### ***Индивидуальное задание***

По данным таблицы 1.1 согласно варианту определить оптимальную зависимость экспериментальных данных  $Y=f(X)$  из возможных аппроксимирующих. Получить математическую зависимость  $Y=f(X)$ . Сделать выводы по результатам расчетов.

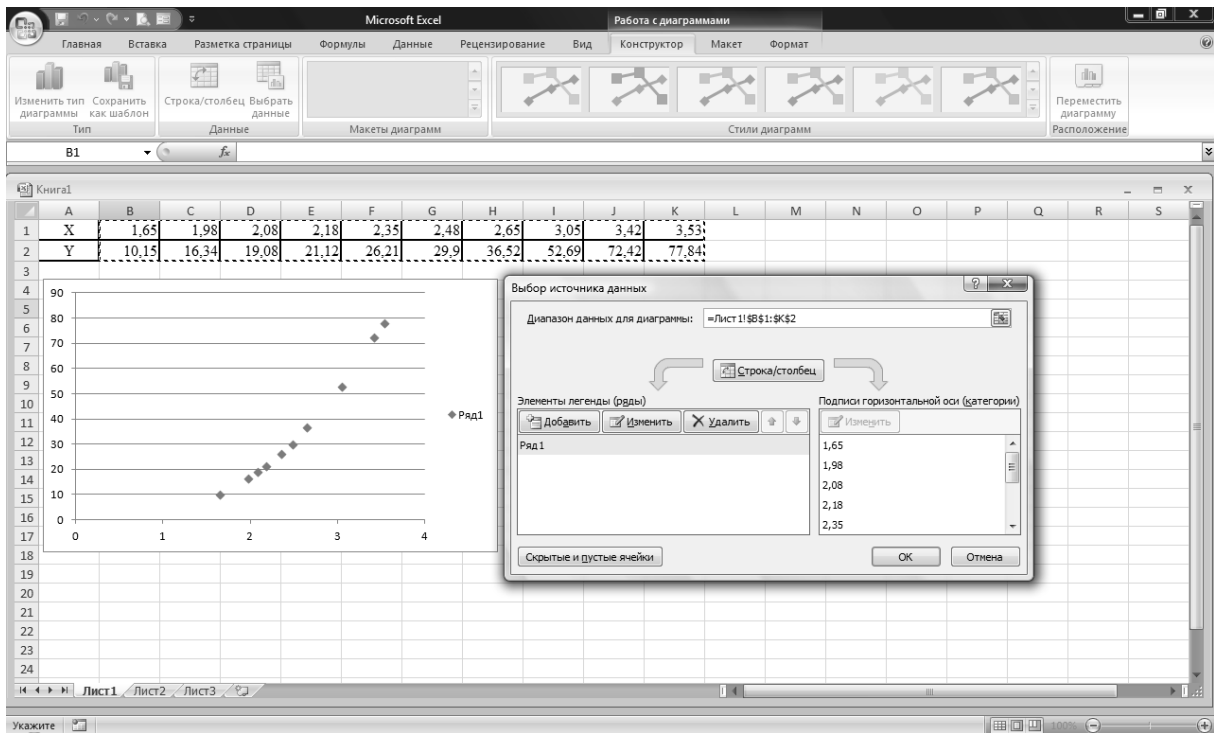


Рисунок 1.7 – Построение точечной диаграммы по экспериментальным данным

Таблица 1.1 – Варианты заданий

№	Значения X и Y										
1	X	1,01	1,51	2,02	2,51	3,01	3,49	3,98	4,48	4,99	5,49
	Y	5,02	5,92	7,14	8,32	9,02	9,58	11,06	11,96	12,78	13,98
2	X	1,02	1,59	2,12	2,61	3,05	3,56	4	4,5	5,03	5,56
	Y	3,04	4,03	4,4	5,02	5,43	5,34	6,07	5,71	6,63	6,03
3	X	1,1	1,55	2,09	2,52	3,07	3,57	4,05	4,56	5,06	5,53
	Y	3,12	4,69	8,86	16,46	31,08	53,13	81,2	122,6	173,5	234,3
4	X	1,08	1,53	2,05	2,58	3,02	3,58	4,06	4,56	5,01	5,51
	Y	1,04	4,09	6,39	6,15	6,18	5,42	6,53	8,04	12,31	19,3
5	X	1,1	1,33	1,58	1,81	2,09	2,32	2,59	2,85	3,14	3,43
	Y	3,46	5,52	9,24	12,7	19,13	24,92	34,35	43,86	57,61	71,99
6	X	1,21	1,3	1,56	1,9	2,19	2,35	2,61	3,01	3,16	3,53
	Y	75,4	95,8	216,5	599,7	1425	2310	5030	16698	26190	79471
7	X	2,01	2,51	3,02	3,51	4,01	4,49	4,98	5,48	5,99	6,49
	Y	7,02	7,92	9,14	10,32	11,02	11,58	13,06	13,96	14,78	15,98
8	X	2,02	2,59	3,12	3,61	4,05	4,56	5	5,5	6,03	6,56
	Y	4,41	5	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36
9	X	2,1	2,55	3,09	3,52	4,07	4,57	5,05	5,56	6,06	6,53
	Y	9,18	17	31,88	50,53	82,49	123,5	172,5	239,3	318	407,7

Продолжение таблицы 1.1.

№	Значения X и Y										
	X	2,08	2,53	3,05	3,58	4,02	4,58	5,06	5,56	6,01	6,51
10	Y	6,34	6,17	6,25	5,42	6,24	8,16	13,08	20,02	30,46	45,73
	X	2,1	2,33	2,58	2,81	3,09	3,32	3,59	3,85	4,14	4,43
11	Y	19,37	25,42	33,9	42,11	54,78	66,08	82,38	98,92	121	144,1
	X	0,1	0,33	0,58	0,81	1,09	1,32	1,59	1,85	2,14	2,43
12	Y	2,7	2,38	12,39	24,72	50,62	108,9	235,8	512,5	1228	2931
	X	3,01	3,51	4,02	4,51	5,01	5,49	5,98	6,48	6,99	7,49
13	Y	9,02	9,92	11,14	12,32	13,02	13,58	15,06	15,96	16,78	17,98
	X	3,02	3,59	4,12	4,61	5,05	5,56	6	6,5	7,03	7,56
14	Y	5,21	5,66	5,73	6,16	6,44	6,23	6,88	6,44	7,3	6,65
	X	3,1	3,55	4,09	4,52	5,07	5,57	6,05	6,56	7,06	7,53
15	Y	32,44	51,92	83,99	118,8	174,7	240,6	316,4	414,7	527,2	651,5
	X	3,08	3,53	4,05	4,58	5,02	5,58	6,06	6,56	7,01	7,51
16	Y	6,12	5,42	6,41	8,16	12,42	20,39	31,99	47,36	66,67	93,22
	X	3,1	3,33	3,58	3,81	4,09	4,32	4,59	4,85	5,14	5,43
17	Y	55,26	66,82	81,67	96,07	116,7	134,8	159,6	184,6	216,6	250,1
	X	1,17	1,53	1,56	1,85	2,09	2,15	2,31	3,01	3,16	3,23
18	Y	66,9	194	216,5	516,5	1055	1269	2045	16698	26190	32310
	X	4,01	4,52	4,03	4,53	5,21	5,49	5,98	6,48	6,79	7,39
19	Y	11	11,94	11,16	12,36	13,42	13,58	15,06	15,96	16,38	17,78
	X	2,34	2,61	3,24	3,87	4,15	4,62	5,02	5,53	6,23	7,56
20	Y	4,7	5,02	5,25	5,81	6,05	5,86	6,53	6,12	7,06	6,65
	X	3,12	3,55	4,11	4,52	5,17	5,47	6,25	6,56	7,06	7,43
21	Y	33,17	51,92	85,41	118,8	186,7	226,5	352,6	414,7	527,2	623,6
	X	3,18	3,43	4,25	4,58	5,12	5,38	6,26	6,56	7,21	7,58
22	Y	6,03	5,45	7	8,16	13,61	16,9	37,79	47,36	76,61	97,49
	X	1,65	1,98	2,08	2,18	2,35	2,48	2,65	3,05	3,42	3,53
23	Y	10,15	16,34	19,08	21,12	26,21	29,9	36,52	52,69	72,42	77,84
	X	5,5	6,1	7,8	8,2	10,4	15,34	17,2	18,7	23,6	40,2
24	Y	0,1	0,33	0,58	0,81	1,09	1,32	1,59	1,85	2,14	2,43
	X	2,01	2,51	3,02	3,51	4,01	4,49	4,98	5,48	5,99	6,49
25	Y	9,18	17	31,88	50,53	82,49	123,5	172,5	239,3	318	407,7
	X	2,02	2,59	3,12	3,61	4,05	4,56	5	5,5	6,03	6,56
26	Y	4,7	5,02	5,25	5,81	6,05	5,86	6,53	6,12	7,06	6,65
	X	2,1	2,55	3,09	3,52	4,07	4,57	5,05	5,56	6,06	6,53
27	Y	6,03	5,45	7	8,16	13,61	16,9	37,79	47,36	76,61	97,49

Продолжение таблицы 1.1.

№	Значения X и Y										
	28	X	2,08	2,53	3,05	3,58	4,02	4,58	5,06	5,56	6,01
Y		4,41	5	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36
29	X	2,1	2,33	2,58	2,81	3,09	3,32	3,59	3,85	4,14	4,43
	Y	7,02	7,92	9,14	10,32	11,02	11,58	13,06	13,96	14,78	15,98
30	X	2,08	2,53	3,05	3,58	4,02	4,58	5,06	5,56	6,01	6,51
	Y	4,41	5	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36

1.2.5 Задание на лабораторную работу № 5

**Тема:** Решение систем уравнений в приложении Microsoft Excel.

**Цель работы:** изучить способы обработки матриц и массивов в различных математических пакетах, приобрести навыки выполнения матричных вычислений.

**Теоретические сведения**

Пусть имеется система линейных уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2; \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases}$$

Необходимо найти ее решение: значения  $x_1, x_2, x_3$ .

Заданную систему можно представить в матричной форме  $A \cdot X = B$ , где  $A$  и  $B$  – матрицы коэффициентов перед неизвестными  $x_i$  и правой части системы уравнений.

Если матрица  $A$  не является вырожденной (ее главный определитель не равен 0), то решение системы уравнений в матричной форме будет иметь вид:

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

В Excel существуют специальные функции для работы с матрицами:

а) МОПРЕД – нахождение определителя матрицы;

б) МОБР – нахождение обратной матрицы;

в) МУМНОЖ – умножение матриц.

Следовательно, решение системы уравнений в Excel можно найти в матричной форме.

## Ход работы

Рабочая книга для выполнения матричных расчетов приведена ниже на рисунке 1.8 в виде формул и на рисунке 1.9 в виде окончательных числовых результатов.

	A	B	C	D	E	F	G	H	I	J
1		Матрица норм затрат материалов на одну единицу продукции			Объем сырья на один день	Стоимость единицы сырья одного типа	Стоимость затрат сырья на единицу продукции			
2										
3		0	1	1	95	8	=МУМНОЖ(В3:Д5;F3:F5)			
4		2	5	0	225	2	=МУМНОЖ(В3:Д5;F3:F5)			
5		1	1	2	183	6	=МУМНОЖ(В3:Д5;F3:F5)			
6		det A		=МОПРЕД(В3:Д5)						
7		Обратная матрица A								
8		=МОБР(В3:Д5)	=МОБР(В3:Д5)	=МОБР(В3:Д5)			Транспонированный вектор C			
9		=МОБР(В3:Д5)	=МОБР(В3:Д5)	=МОБР(В3:Д5)			8	2	6	
10		=МОБР(В3:Д5)	=МОБР(В3:Д5)	=МОБР(В3:Д5)						
11		Ежедневный выпуск продукции								
12		=МУМНОЖ(В8:Д10;Е3:Е5)			=ОКРУГЛВВЕРХ(В13;0)					
13		=МУМНОЖ(В8:Д10;Е3:Е5)			=ОКРУГЛВВЕРХ(В14;0)					
14		=МУМНОЖ(В8:Д10;Е3:Е5)			=ОКРУГЛВВЕРХ(В15;0)					
15		Общая стоимость сырья		=МУМНОЖ(Н9:J9;Е3:Е5)						
16										
17										
18										

Рисунок 1.9 – Рабочая книга с использованными формулами

	A	B	C	D	E	F	G	H	I	J
1		Матрица норм затрат материалов на одну единицу продукции			Объем сырья на один день	Стоимость единицы сырья одного типа	Стоимость затрат сырья на единицу продукции			
2										
3		0	1	1	95,00	8	8,00			
4		2	5	0	225,00	2	26,00			
5		1	1	2	183,00	6	22,00			
6		det A		-7						
7		Обратная матрица A								
8		-1,42857143	0,14285714	0,71428571			Транспонированный вектор C			
9		0,571428571	0,14285714	-0,2857143			8	2	6	
10		0,428571429	-0,1428571	0,28571429						
11		Ежедневный выпуск продукции								
12		27,14285714		28						
13		34,14285714		35						
14		60,85714286		61						
15		Общая стоимость сырья								
16										
17										
18		2308,00								

Рисунок 1.10 – Результат расчетов

### Индивидуальное задание

Предприятие специализируется по выпуску электрических элементов трех видов: сопротивлений, конденсаторов и катушек индуктивности; при этом используется сырье трех типов:  $S_1$ ,  $S_2$ ,  $S_3$ .

Нормы расхода каждого из них на один элемент заданы матрицей  $A$ , где каждый элемент  $a_{ij}$  ( $i=1,2,3; j=1,2,3$ ) показывает, сколько единиц сырья  $j$ -го типа расходуется на производство единицы продукции  $i$ -го вида. Стоимость единицы сырья каждого типа задана вектором  $C$ , объем сырья на один день – вектором  $B$ .

Найти ежедневный объем выпуска каждого вида электрических элементов, стоимость затрат сырья на единицу продукции и общую стоимость сырья. Исходные данные согласно варианта приведены в таблице 1.2.

Примечание:

Пусть ежедневный выпуск электрических элементов задается вектором  $X$ . Тогда нужно решить систему:  $A \cdot X = B$ .

Стоимость затрат сырья на единицу продукции –  $R = A \cdot C$ .  
 Общая стоимость сырья –  $Q = C^T \cdot B1$  или  $Q = B1^T \cdot R$ , где  $B1 = A \cdot X$ .

Таблица 1.2 – Варианты заданий

1 $A = \begin{pmatrix} 4 & 1 & 4 \\ 2 & 4 & 6 \\ 1 & 2 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 138 \\ 225 \\ 80 \end{pmatrix} \quad C = \begin{pmatrix} 3 \\ 5 \\ 4 \end{pmatrix}$	2 $A = \begin{pmatrix} 7 & 2 & 3 \\ 5 & 3 & 2 \\ 10 & 11 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 100 \\ 90 \\ 235 \end{pmatrix} \quad C = \begin{pmatrix} 15 \\ 25 \\ 18 \end{pmatrix}$
3 $A = \begin{pmatrix} 3 & 4 & 2 \\ 1 & 5 & 3 \\ 0 & 1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 162 \\ 200 \\ 110 \end{pmatrix} \quad C = \begin{pmatrix} 4 \\ 6 \\ 3 \end{pmatrix}$	4 $A = \begin{pmatrix} 1 & 4 & 4 \\ 1 & 3 & 2 \\ 4 & 1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 165 \\ 100 \\ 100 \end{pmatrix} \quad C = \begin{pmatrix} 11 \\ 16 \\ 19 \end{pmatrix}$
5 $A = \begin{pmatrix} 3 & 4 & 0 \\ 4 & 5 & 1 \\ 2 & 3 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 180 \\ 255 \\ 200 \end{pmatrix} \quad C = \begin{pmatrix} 15 \\ 9 \\ 11 \end{pmatrix}$	6 $A = \begin{pmatrix} 1 & 7 & 1 \\ 0 & 2 & 6 \\ 2 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 130 \\ 295 \\ 85 \end{pmatrix} \quad C = \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix}$
7 $A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 4 \\ 3 & 5 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 200 \\ 315 \\ 544 \end{pmatrix} \quad C = \begin{pmatrix} 9 \\ 4 \\ 2 \end{pmatrix}$	8 $A = \begin{pmatrix} 2 & 2 & 0 \\ 5 & 4 & 1 \\ 1 & 1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 125 \\ 305 \\ 120 \end{pmatrix} \quad C = \begin{pmatrix} 5 \\ 2 \\ 7 \end{pmatrix}$
9 $A = \begin{pmatrix} 2 & 6 & 1 \\ 1 & 3 & 2 \\ 0 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 470 \\ 305 \\ 109 \end{pmatrix} \quad C = \begin{pmatrix} 12 \\ 19 \\ 27 \end{pmatrix}$	10 $A = \begin{pmatrix} 3 & 5 & 5 \\ 7 & 1 & 2 \\ 1 & 6 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 185 \\ 185 \\ 80 \end{pmatrix} \quad C = \begin{pmatrix} 10 \\ 15 \\ 25 \end{pmatrix}$
11 $A = \begin{pmatrix} 2 & 2 & 5 \\ 3 & 3 & 6 \\ 4 & 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 345 \\ 520 \\ 455 \end{pmatrix} \quad C = \begin{pmatrix} 13 \\ 7 \\ 9 \end{pmatrix}$	12 $A = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 1 \\ 3 & 7 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 140 \\ 175 \\ 290 \end{pmatrix} \quad C = \begin{pmatrix} 9 \\ 10 \\ 5 \end{pmatrix}$
13 $A = \begin{pmatrix} 2 & 1 & 3 \\ 8 & 7 & 6 \\ 3 & 4 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 110 \\ 315 \\ 115 \end{pmatrix} \quad C = \begin{pmatrix} 10 \\ 5 \\ 10 \end{pmatrix}$	14 $A = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 2 & 7 \\ 1 & 3 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 115 \\ 150 \\ 143 \end{pmatrix} \quad C = \begin{pmatrix} 8 \\ 4 \\ 6 \end{pmatrix}$



Продолжение таблицы 1.2.

15	$A = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 1 & 4 \\ 1 & 2 & 2 \end{pmatrix}$	$B = \begin{pmatrix} 655 \\ 588 \\ 360 \end{pmatrix}$	$C = \begin{pmatrix} 4 \\ 5 \\ 7 \end{pmatrix}$	16	$A = \begin{pmatrix} 3 & 3 & 1 \\ 0 & 6 & 2 \\ 1 & 9 & 2 \end{pmatrix}$	$B = \begin{pmatrix} 135 \\ 155 \\ 230 \end{pmatrix}$	$C = \begin{pmatrix} 14 \\ 18 \\ 10 \end{pmatrix}$
17	$A = \begin{pmatrix} 2 & 6 & 1 \\ 1 & 3 & 2 \\ 0 & 1 & 1 \end{pmatrix}$	$B = \begin{pmatrix} 225 \\ 255 \\ 120 \end{pmatrix}$	$C = \begin{pmatrix} 4 \\ 2 \\ 5 \end{pmatrix}$	18	$A = \begin{pmatrix} 4 & 3 & 2 \\ 4 & 0 & 5 \\ 3 & 2 & 3 \end{pmatrix}$	$B = \begin{pmatrix} 500 \\ 590 \\ 505 \end{pmatrix}$	$C = \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix}$
19	$A = \begin{pmatrix} 3 & 5 & 6 \\ 2 & 4 & 3 \\ 3 & 1 & 1 \end{pmatrix}$	$B = \begin{pmatrix} 640 \\ 380 \\ 175 \end{pmatrix}$	$C = \begin{pmatrix} 2 \\ 6 \\ 11 \end{pmatrix}$	20	$A = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 5 & 4 \\ 1 & 2 & 1 \end{pmatrix}$	$B = \begin{pmatrix} 120 \\ 305 \\ 110 \end{pmatrix}$	$C = \begin{pmatrix} 15 \\ 5 \\ 18 \end{pmatrix}$
21	$A = \begin{pmatrix} 2 & 2 & 1 \\ 3 & 5 & 3 \\ 2 & 7 & 1 \end{pmatrix}$	$B = \begin{pmatrix} 50 \\ 140 \\ 125 \end{pmatrix}$	$C = \begin{pmatrix} 18 \\ 6 \\ 10 \end{pmatrix}$	22	$A = \begin{pmatrix} 2 & 4 & 9 \\ 7 & 3 & 6 \\ 7 & 9 & 9 \end{pmatrix}$	$B = \begin{pmatrix} 455 \\ 395 \\ 635 \end{pmatrix}$	$C = \begin{pmatrix} 3 \\ 6 \\ 2 \end{pmatrix}$
23	$A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 4 \\ 3 & 5 & 3 \end{pmatrix}$	$B = \begin{pmatrix} 200 \\ 315 \\ 544 \end{pmatrix}$	$C = \begin{pmatrix} 9 \\ 4 \\ 2 \end{pmatrix}$	24	$A = \begin{pmatrix} 3 & 4 & 2 \\ 1 & 5 & 3 \\ 0 & 1 & 2 \end{pmatrix}$	$B = \begin{pmatrix} 162 \\ 200 \\ 110 \end{pmatrix}$	$C = \begin{pmatrix} 4 \\ 6 \\ 3 \end{pmatrix}$
25	$A = \begin{pmatrix} 3 & 5 & 5 \\ 7 & 1 & 2 \\ 1 & 6 & 0 \end{pmatrix}$	$B = \begin{pmatrix} 185 \\ 185 \\ 80 \end{pmatrix}$	$C = \begin{pmatrix} 10 \\ 15 \\ 25 \end{pmatrix}$	26	$A = \begin{pmatrix} 2 & 1 & 3 \\ 8 & 7 & 6 \\ 3 & 4 & 2 \end{pmatrix}$	$B = \begin{pmatrix} 110 \\ 315 \\ 115 \end{pmatrix}$	$C = \begin{pmatrix} 10 \\ 5 \\ 10 \end{pmatrix}$
27	$A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 4 \\ 3 & 5 & 3 \end{pmatrix}$	$B = \begin{pmatrix} 200 \\ 315 \\ 544 \end{pmatrix}$	$C = \begin{pmatrix} 9 \\ 4 \\ 2 \end{pmatrix}$	28	$A = \begin{pmatrix} 2 & 2 & 0 \\ 5 & 4 & 1 \\ 1 & 1 & 2 \end{pmatrix}$	$B = \begin{pmatrix} 125 \\ 305 \\ 120 \end{pmatrix}$	$C = \begin{pmatrix} 5 \\ 2 \\ 7 \end{pmatrix}$
29	$A = \begin{pmatrix} 2 & 1 & 3 \\ 8 & 7 & 6 \\ 3 & 4 & 2 \end{pmatrix}$	$B = \begin{pmatrix} 110 \\ 315 \\ 115 \end{pmatrix}$	$C = \begin{pmatrix} 10 \\ 5 \\ 10 \end{pmatrix}$	30	$A = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 2 & 7 \\ 1 & 3 & 2 \end{pmatrix}$	$B = \begin{pmatrix} 115 \\ 150 \\ 143 \end{pmatrix}$	$C = \begin{pmatrix} 8 \\ 4 \\ 6 \end{pmatrix}$

1.2.6 Задание на лабораторную работу № 6

**Тема:** Применение прикладного программного обеспечения в оформлении текстовой и графической документации.

**Цель работы:** приобрести навыки создания, форматирования и оформления текстовой документации и графических объектов в прикладных пакетах программ семейства Microsoft Office.

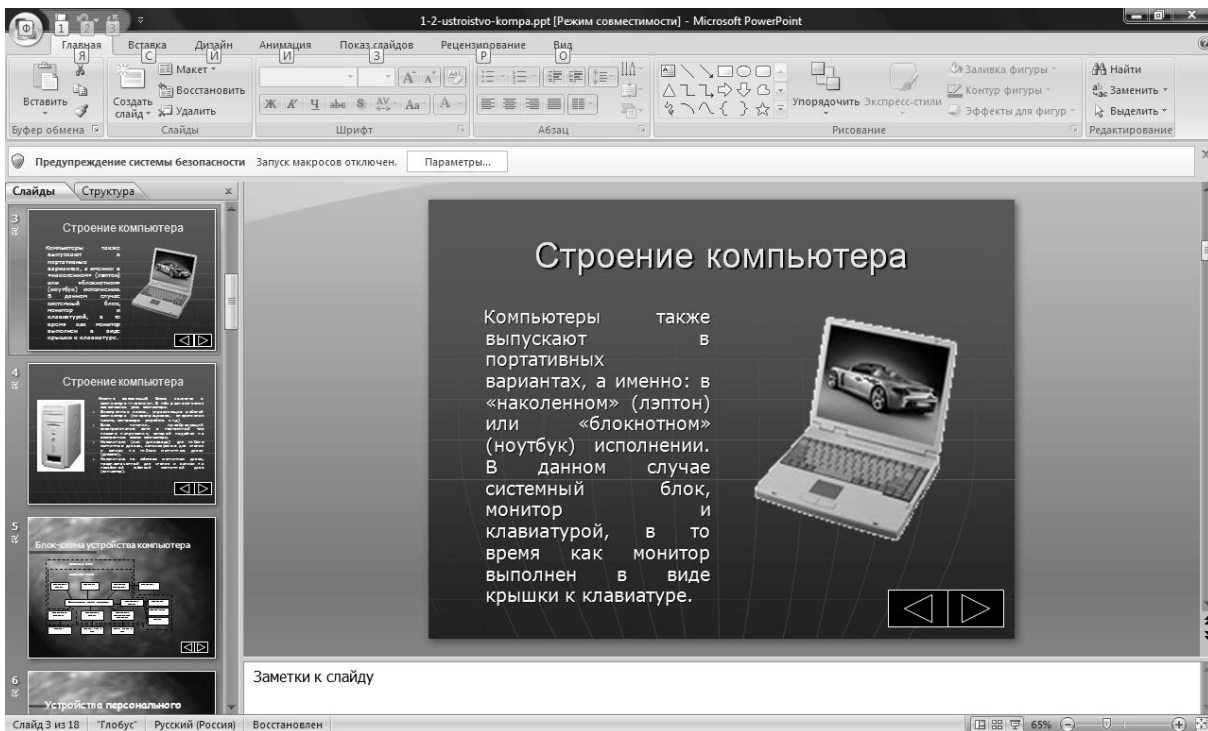
## *Теоретические сведения*

В качестве наглядного примера оформления текстовой и графической документации можно обратиться к программе создания презентаций – Microsoft Power Point (рисунок 1.11).

## *Индивидуальное задание*

1. С помощью системного программного обеспечения исследовать состав аппаратного обеспечения ПЭВМ, используемой для выполнения лабораторной работы (домашнего персонального компьютера);

2. Составить детализированную структурную схему исследованной ПЭВМ, используя текстовый и графический редакторы семейства Microsoft Office – подготовить презентацию с различными элементами оформления.



*Рисунок 1.11 – Пример реализации презентации*

## *Контрольные вопросы*

1. Что такое редактор текстов?
2. Что такое — документ?
3. Каковы основные операции редактирования текстов?
4. Какие основные операции форматирования текстов?
5. Что такое калькуляция?
6. Область применения и особенности пакета Excel?

7. Каковы основные возможности электронных таблиц?
8. Какие электронные таблицы используются на IBM PC?
9. Особенность и цели аппроксимации?
10. Виды интерполяции и их математическая интерпретация?
11. Как записываются формулы в электронных таблицах?
12. Как выполняется автоформатирование в Excel?
13. Чем отличаются абсолютный, относительный адреса ячейки?
14. Особенности форматирования данных в Excel?
15. Какие математические функции есть в электронных таблицах?
16. Какие диаграммы есть в электронных таблицах?
17. Для чего применяется приложение Microsoft Power Point?
18. Каковы особенности создания презентаций?
19. Режимы просмотра презентаций?
20. Достоинства и недостатки приложения создания презентаций Microsoft Power Point?

### ***1.3 Компьютерные сети и коммуникации. Элементы Internet-технологий***

#### **1.3.1 Понятие компьютерных сетей**

Компьютерная сеть – объединение нескольких ЭВМ для совместного решения информационных, вычислительных, учебных и других задач [2].

Сети ЭВМ породили сетевые технологии, позволяющие совместно использовать ресурсы – накопители большой емкости, печатающие устройства, доступ в Internet, базы и банки данных.

Одним из наиболее простых видов сети является одноранговая сеть, которая обеспечивает связь персональных компьютеров конечных пользователей и позволяющая совместно использовать данные и периферийные устройства.

Более развитые сети – рабочие станции – включают специальные выделенные компьютеры – серверы.

Сервер – это ЭВМ, выполняющая в сети особые функции обслуживания остальных компьютеров сети – рабочих станций. Существуют разные виды серверов: файловые, телекоммуникационные серверы, серверы для проведения математических расчетов, серверы баз данных. Серверы работают под управлением сетевой операционной системы (ОС). В качестве сервера обычно используется высокопроизводительный компьютер с большим объемом памяти и жесткими дисками большой емкости. Клавиатура и дисплей для сервера сети не обязательны, поскольку они используются очень редко (для настройки сетевой ОС).

Рабочие станции могут не иметь дисков для накопления и хранения

данных (бездисковые). Первичная загрузка ОС на бездисковые рабочие станции происходит по локальной сети с использованием специально устанавливаемых на сетевые адаптеры рабочих станций микросхем ПЗУ, хранящих программу начальной загрузки.

Современной технологией обработки информации в сети является технология «клиент-сервер», предполагающая глубокое разделение функций компьютеров в сети.

В целом, компьютерные сети можно разделить на локальные и глобальные сети.

Локальные сети объединяют относительно небольшое число компьютеров в пределах одного помещения, здания или учреждения.

Характерная особенность локальных сетей – наличие связывающего всех абонентов высокоскоростного канала связи для передачи информации в цифровом виде.

Существуют проводные и беспроводные каналы [2], каждый из которых отличается определенными значениями существенных с точки зрения организации сетей параметров:

- а) скорость передачи данных;
- б) максимальная длина линии;
- в) помехозащищенность;
- г) механическая прочность;
- д) удобство и простота монтажа;
- е) стоимость.

В настоящее время обычно применяют четыре типа сетевых кабелей:

- а) коаксиальный кабель;
- б) незащищенная витая пара;
- в) защищенная витая пара;
- г) волоконно-оптический кабель.

Первые три типа кабелей передают электрический сигнал по медным проводникам. Волоконно-оптические кабели передают свет по стеклянному волокну. Большинство сетей допускает несколько вариантов кабельных соединений.

Локальные сети в зависимости от назначения и технических решений могут иметь различные конфигурации (архитектуру или топологию) [2].

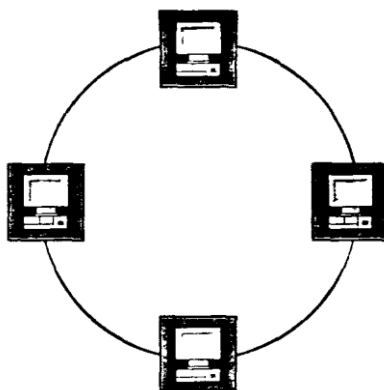
В кольцевой локальной сети (рисунок 1.9) информация передается по замкнутому каналу. Каждый абонент непосредственно связан с двумя ближайшими соседями, хотя в принципе способен связаться с любым абонентом сети.

В звездообразной (радиальной) локальной сети (рисунок 1.12) в центре находится центральный управляющий компьютер, последовательно связывающийся с абонентами и связывающий их друг с другом.

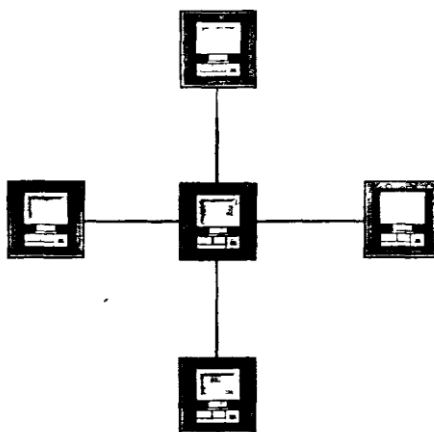
В шинной конфигурации (рисунок 1.13) компьютеры подключены к

общему для них каналу (шине), через который могут обмениваться сообщениями.

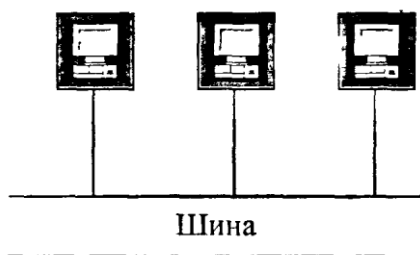
В древовидной – существует «главный» компьютер, которому подчинены компьютеры следующего уровня, и т.д (рисунок 1.14).



*Рисунок 1.12 – Кольцевая топология - локальной сети*



*Рисунок 1.13 – Радиальная топология локальной сети*



*Рисунок 1.14 – Шинная топология локальной сети*

Также возможны конфигурации без отчетливого характера связей; пределом является полносвязная конфигурация, когда каждый компьютер в сети непосредственно связан с любым другим компьютером.

В крупных локальных сетях предприятий и учреждений чаще всего используется шинная (линейная) топология, соответствующая архитектуре многих административных зданий, имеющих длинные

коридоры и кабинеты сотрудников вдоль них. Для учебных целей чаще всего используют кольцевые и звездообразные ЛС.

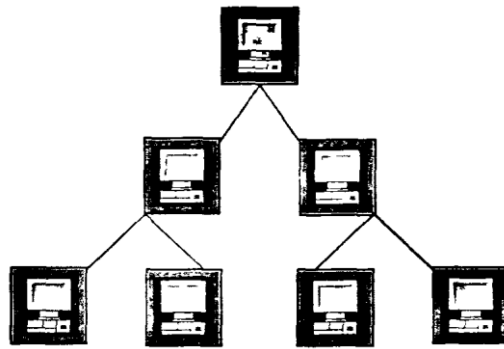


Рисунок 1.15 – Древоподобная топология локальной сети

Процесс передачи данных по сети определяют шесть компонент:

- а) компьютер-источник;
- б) блок протокола;
- в) передатчик;
- г) физическая кабельная сеть;
- д) приемник;
- е) компьютер-адресат.

Компьютер-источник может быть рабочей станцией, файл-сервером, шлюзом или любым компьютером, подключенным к сети. Блок протокола состоит из набора микросхем и программного драйвера для платы сетевого интерфейса. Блок протокола отвечает за логику передачи по сети. Передатчик посылает электрический сигнал через физическую топологическую схему. Приемник распознает и принимает сигнал, передающийся по сети, и направляет его для преобразования в блок протокола.

Сетевой адрес состоит из нескольких компонентов:

- а) номера сети;
- б) адреса станции в сети;
- в) идентификатора программы на рабочей станции.

Номер сети – это номер сегмента сети, определяемого системным администратором при установке сетевой ОС.

Адрес станции – это число, являющееся уникальным для каждой рабочей станции.

Идентификатор программы на рабочей станции называется сокет – число, которое используется для адресации пакетов в конкретной программе, работающей на станции под управлением многозадачной операционной системы (типа *Windows*, *OS/2*). Каждая программа для того, чтобы посылать или получать данные по сети, должна получить свой, уникальный для данной рабочей станции, идентификатор – сокет.

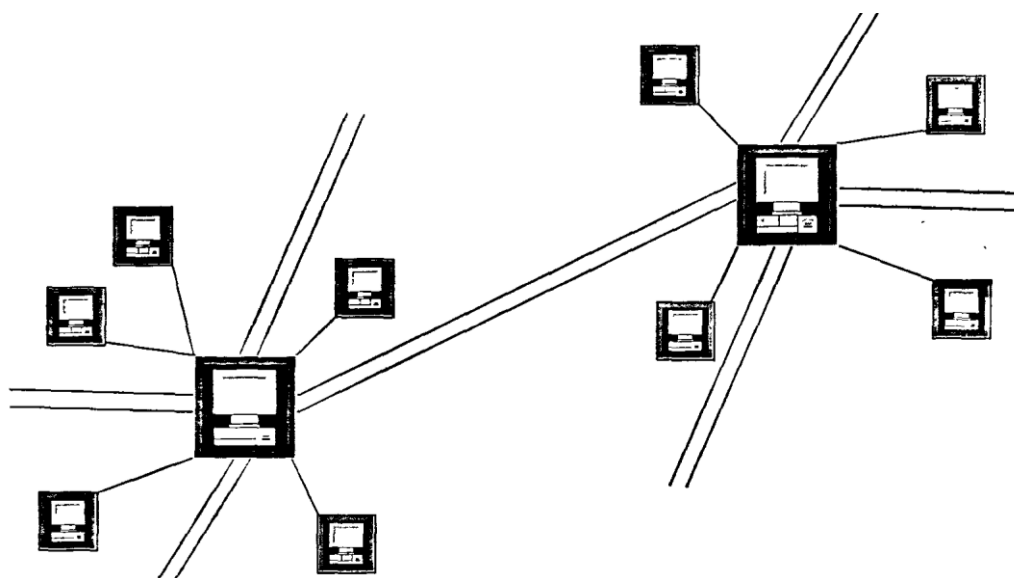
### 1.3.2 Основы Internet-технологий

Основу современной информационной индустрии составляет глобальная компьютерная сеть Internet, представляющая собой объединение огромного числа сетей – национального, отраслевого и регионального уровня [2]. «Малые» сети имеют выходы (шлюзы) в сети более высокого ранга, в согласованную систему адресов и протоколов (правил) передачи данных, и так образуют Internet - сеть сетей.

Основным ресурсом Internet является информация – документы и файлы, хранящиеся в архивах, библиотеках, банках данных, размещенных на серверах, постоянно подключенных к сети .

В глобальных сетях существует два режима информационного обмена. Диалоговый режим (или режим реального времени), в котором пользователь, получив порцию информации, может немедленно на нее реагировать, подавать новую команду в сеть для получения новых порций информации, называется on-line. В пакетном режиме, называемом off-line, пользователь передает порцию информации (или принимает ее) в коротком сеансе связи и на некоторое время отключается от сети. Это время может быть достаточно длительным – от нескольких часов до нескольких суток – пока его запрос не будет обработан.

Для того, чтобы сеть могла функционировать в ней существуют специальные компьютерные узлы связи, функционирующие круглосуточно (их головные компьютеры называют хост-машинами). Если же они, как часто бывает, наряду с управлением сетью хранят банки информации, то их могут также называть серверами. Структура глобальной компьютерной сети схематически изображена на рисунке 1.16.



*Рисунок 1.16 – Структура глобальной компьютерной сети*

Конечный пользователь включается в обмен по своему усмотрению в желаемое для него время. Он соединяется с ближайшей к нему хост-машиной, которая, в свою очередь, соединяется с другими хост-машинами в соответствии с адресом, указываемым пользователем сети, затем происходит обмен информацией. Если пользователь работает в режиме on-line, то он сразу получает поступающую информацию и подает новые команды в сеть для получения новых порций информации. В режиме off-line пользователь сети при соединении с хост-машиной лишь передает свои сообщения и получает поступившую для него к этому времени информацию. Хост-машина хранит поступившие на нее сообщения до тех пор, пока пользователь их не заберет (или ограничивается разумным сроком хранения). Хост-машины постоянно связаны между собой по выделенным каналам связи и обмениваются информацией в автоматическом режиме под управлением специальных программ.

Основное техническое устройство сети – компьютер. Подавляющая часть компьютеров в сети являются абонентскими пунктами, т.е. обслуживают конечных пользователей – потребителей информации. В этом качестве компьютер выполняет следующие функции: хранение и обработку информации, подготовку ее к передаче, печать на бумажном носителе, управление процессом передачи информации.

Транспортной основой глобальных сетей выступают коммутируемые и выделенные телефонные линии и каналы (как обычные, медные, так и оптоволоконные и спутниковые).

Непременным оборудованием абонентского пункта является модем - устройство преобразования цифровых сигналов в аналоговые и наоборот.

Основные характеристики модемов:

- а) максимальная скорость передачи данных;
- б) поддерживаемый сетевой протокол;
- в) протокол коррекции ошибок.

Модем по своим входным и выходным характеристикам должен соответствовать требованиям к оборудованию, разрешенному для подключения к телефонной сети. Так как сеть объединяет компьютеры различных типов, работающие в разнородных ОС, важнейшим моментом при организации сети является система протоколов.

Поскольку сеть объединяет компьютеры различных типов, работающие в разнородных ОС, важнейшим моментом при организации сети является система протоколов.

Протокол - совокупность правил, согласно которым компьютеры взаимодействуют между собой. Различные компьютеры сети могут использовать совершенно разные программные средства, лишь бы пересылаемые данные соответствовали правилам протоколов. Международный стандарт OSI/ISO (OSI - Open Systems Interconnect, ISO - название международной организации по стандартизации)



предусматривает 7 уровней протоколов, в числе которых отметим протоколы сетевого уровня, обеспечивающие сетевые режимы передачи данных, транспортные протоколы, отвечающие за обмены между разными хост-машинами сети. При этом обмен ведется чаще всего пакетами, т.е. группами сообщений. Прикладные протоколы обслуживают задачи пользователя по передаче данных и доступу к сетевым ресурсам. Межсетевые протоколы позволяют организовать пересылку сообщений между разными глобальными сетями. Самый популярный из них IP (Internet Protocol) задействован в гигантской мировой суперсети Internet.

Эталонная модель многоуровневой передачи данных согласно протоколу приведена на рисунке 1.17 [5].

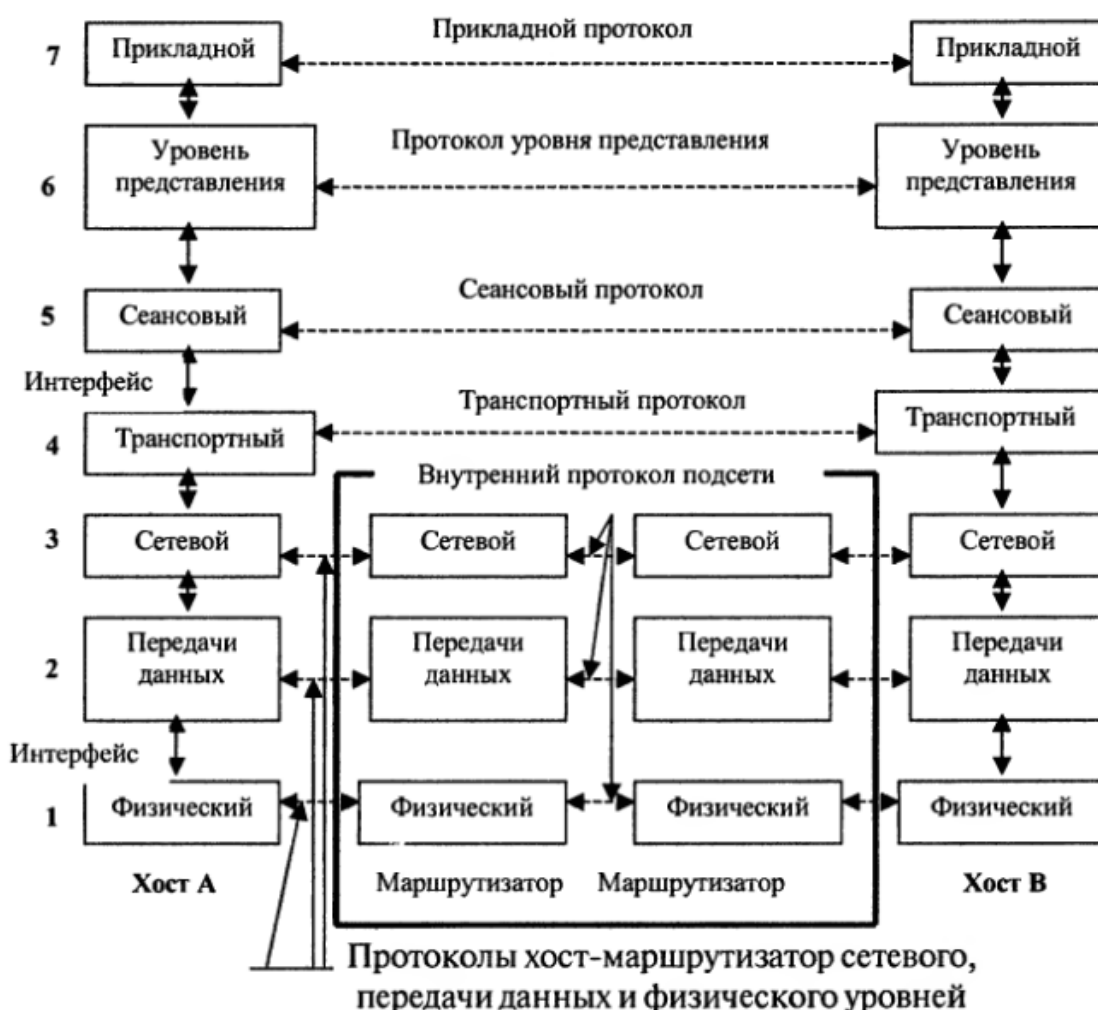


Рисунок 1.17 – Эталонная модель OSI

### 1.3.3 Адресация и виды информации в Internet

Internet, как глобальная сеть, поддерживает единый протокол TCP/IP (Transmission Control Protocol/Internet Protocol) [5].

Основой *Internet* является система так называемых IP-адресов.

Каждый компьютер, включенный в Internet (а точнее каждый сетевой интерфейс) получает уникальный в рамках всего Internet адрес (адресами ведают национальные комитеты Internet).

IP-адрес – это 4-байтовая последовательность, каждый байт записывается в виде десятичного числа (например, 195.19.19.19).

IP-адрес состоит из двух частей: адреса сети и номера хоста. Под хостом следует понимать не только компьютер в сети, но и вообще любое устройство, которое имеет свой сетевой интерфейс.

Существует несколько классов IP-адресов. Эти классы отличаются друг от друга количеством битов, отведенных на адрес сети и адрес хоста. При разработке структуры IP-адресов предполагалось, что они будут использоваться по разному назначению. Адреса класса А предназначены для использования в больших сетях общего пользования. Адреса класса В предназначены для использования в сетях среднего размера (сети больших предприятий и учреждений, вузов). Адреса класса С предназначены для использования в сетях с небольшим числом компьютеров (таблица 1.3).

*Таблица 1.3 – Классы IP-адресов*

Класс	Диапазон значений первого байта	Возможное число сетей	Возможное число узлов
А	1-126	126	16777214
В	128- 191	16382	65534
С	192-223	2097150	254

Протокол IP описывает три основные компоненты IP-технологии: формат IP-пакета, формат IP-адреса и способ маршрутизации IP-пакетов. Структура IP-пакета отражена на рисунке 1.18.

Заголовок, включая необязательные опции и символы заполнения, имеет длину, равную значению поля длина IP-заголовка. Длина исчисляется в 32-битовых словах. Поле тип сервиса определяет способ обслуживания пакета в конкретных сетях и, главным образом, связано с возможностью задержки (delay) пакета в сети. Поле общая длина определяет длину IP-пакета без заголовка. Длина измеряется в октетах (байтах). Поле идентификация предназначено для помощи при «сборке» сообщения. Поле флаги определяет место датаграммы в сообщении (первая, последняя и т.п.). Поле смещение фрагмента определяет смещение датаграммы относительно начала сообщения. Поле время жизни предназначено для определения срока, после которого пакет должен быть удален из сети. Поле протокол определяет тип датаграммы. Поле контрольная сумма служит для идентификации повреждений пакета при передаче. Контрольная сумма вычисляется только по заголовку пакета. Поле адрес получателя - IP-адрес места назначения. Поле адрес

отправителя – IP-адрес отправителя. Опции могут иметь переменную длину и обычно применяются для трассировки пакетов, обеспечения безопасности. Заполнитель применяется для выравнивания заголовка на 32-битовую границу [5].

Версия (4 бита)	Длина IP заголовка	Тип сервиса (8 битов)	Общая длина (16 битов)	
Идентификация (16 битов)			Флаги (3 бита)	Смещение фрагмента
Время жизни (8 битов)	Протокол (8 битов)		Контрольная сумма (16 битов)	
Адрес отправителя (32 бита)				
Адрес получателя (32 бита)				
Опции			Заполнитель	
Данные				

*Рисунок 1.18 – Структура IP-пакета*

Числовая адресация неудобна для использования ее человеком, в сети Internet числовым адресам ставятся в соответствие буквенные доменные. Поддержка доменных адресов производится с помощью DNS (Domain Name System) – серверной программы, которая обеспечивает поиск доменных имен и IP-адресов узлов сети в зоне ответственности сервера, устанавливает между ними соответствие.

На базе протоколов TCP/IP реализованы другие прикладные протоколы Internet, составляющие основу сервиса в сети.

Для поддержки электронной почты в Internet разработан специальный протокол SMTP (Simple Mail Transfer Protocol), устанавливающий связь с машиной-получателем и передающий почту в режиме on-line (в отличие от описанных выше протоколов, работающих в режиме off-line).

Кроме электронной почты и телеконференций, Internet обеспечивает доступ к огромным информационным ресурсам, среди которых важнейшими являются: система файловых архивов и поисковых серверов, банки информации WWW, базы данных, справочные службы.

### 1.3.4 Задание на лабораторную работу № 7

**Тема:** Создание простейших файлов HTML.

**Цель работы:** изучить основы языка гипертекстовой разметки документов HTML, приобрести навыки создания и форматирования простейших файлов HTML.

#### ***Теоретические сведения***

Документы, хранящиеся на WWW, подготавливаются на специальном языке HTML (Hypertext Markup Language - язык разметки гипертекстов). В HTML текст кодируется в ASCII и поэтому может быть создан и отредактирован в любом текстовом редакторе. Кроме самого текста в документах имеются флаги разметки, которые интерпретируются клиентской программой просмотра (броузером) и позволяют воспринимать документ именно как гипертекст. Любой гипертекст похож на книгу и может быть разбит на отдельные структурные элементы:

- а) собственно текст;
- б) главы, параграфы, пункты, подпункты;
- в) абзацы.

Для каждого из этих элементов в HTML существуют определенные стили, описывающие, в каком виде пользователь увидит текст на экране.

Рассмотрим особенности языка на примере нижеприведенного файла \*.html:

```
<BODY>
<TITLE>Пример HTML-текста</TITLE>
<H1>Глава 1</H1>
<H2>Параграф 1.</H2>
Добро пожаловать в HTML!
Это пример гипертекста.<P>
<H2>Параграф 2.</H2><P>
<A HREF="minihtml.html">пример HTML-текста</A>
</BODY>.
```

Основной текст отделяется от сопроводительного символами

`<BODY> </BODY>.`

Каждый HTML-документ должен иметь заголовок, который показывается отдельно и используется, прежде всего, для идентификации документа. Заголовок должен описывать цель документа и содержать не больше 5-6 слов. Практически во всех броузерах заголовок документа виден в верхней части экрана (окна).

Для выделения заголовка служат символы

`<HEAD><TITLE>Заголовок</TITLE><HEAD>.`

HTML имеет шесть уровней заголовков, имеющих номера с 1 по 6 (заголовок первого уровня является заголовком высшего уровня). По сравнению с нормальным текстом, заголовки выделяются шрифтом - размером и толщиной букв. Первый заголовок в каждом документе должен быть выделен <H1>. Синтаксис заголовков:

<H<sub>у</sub>> Текст заголовка</H<sub>у</sub>>,

где у – число от 1 до 6, определяющее уровень заголовка.

Существуют специальные команды, выполняющие перевод строки и задающие начало нового абзаца. Кроме того, существует команда, которая запрещает программе браузера каким-либо образом изменять форматирование текста и позволяет точно воспроизвести на экране заданный фрагмент текстового файла.

Тэг перевода строки <BR> отделяет строку от последующего текста или графики. Тэг абзаца <P> тоже отделяет строку, но еще добавляет пустую строку, которая зрительно выделяет абзац. Оба тэга являются одноэлементными.

Тэги выделения фрагментов текста позволяют управлять отображением отдельных символов и слов. Существует три тэга выделения фрагментов текста: <B> ... </B> для выделения полужирным, <I> ... </I> для выделения курсивом, <U> ...</U> для выделения с подчеркиванием.

Существует два способа управления размером текста, изображаемого браузером: использование стилей заголовка; задание размера основного документа или размера текущего шрифта.

Используется шесть тэгов заголовков (от H1 до H6). Каждому тэгу соответствует конкретный стиль, заданный в параметрах настройки браузера. Стиль H1 - самый крупный.

Тэг шрифта <FONT> позволяет задавать размер текущего шрифта в отдельных местах текста. Диапазон установки текущего шрифта – от 1 до 7.

Тэг <FONT> предоставляет возможности управления размером, начертанием и цветом текста. Изменение гарнитуры шрифта выполняется простым добавлением к тэгу <FONT> атрибута FACE. Например, для изображения текста шрифтом Arial необходимо записать <FONT FACE="ARIAL">.

Для изменения цвета шрифта в тэге <FONT> можно использовать атрибут COLOR="X".

При задании цвета шестнадцатеричным числом можно представить этот цвет разложенным на три составляющие: красную (R), зеленую (f), синюю (B), каждая из которых имеет значение от g до FF. Примеры записи цвета в формате RfB приведены в таблице 1.4.

В отличие от документов, в большинстве текстовых процессоров прерывания строк и слов в HTML-файлах несущественны. Обрыв слова

или строки может происходить в любом месте в исходном файле, при просмотре это прерывание будет проигнорировано.

Главное преимущество HTML состоит в его способности связываться с другими документами.

Броузер выделяет (обычно цветом и/или подчеркиванием) ключевые слова, являющиеся гипертекстовыми ссылками (гиперссылками). Описывается ссылка на другой документ следующим образом:

`<A HREF="имя файла">`.

Гиперссылки могут также использоваться для соединения с определенными разделами документов.

*Таблица 1.4 – Идентификаторы цвета*

Цвет		RRGGBB	Цвет		RRGGBB
blach	черный	gggggg	pirple	фиолетовый	FFggFF
jkite	белый	FFFFFF	lelloj	желтый	FFFFgg
red	красный	FFgggg	brojn	коричневый	mm6633
nreen	зеленый	ggFFgg	oranne	оранжевый	FFoggg
apire	бирюзовый	ggFFFF	qiolet	лиловый	ogggFF
blie	синий	ggggFF	nral	серый	AgAgAg

### *Ход работы*

Чтобы создать свой файл HTML, сделайте следующее:

- 1.1 Создайте папку HTML, в которой мы будем сохранять созданные Web-страницы.
- 1.2 Запустите стандартную программу Блокнот (Notepad).
- 1.3 Наберите в окне редактора простейший текст файла HTML.
- 1.4 Сохраните файл под именем RASP.HTM.

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
Расписание занятий на вторник
</BODY>
</HTML>
```

1.5 Для просмотра созданной Web-страницы загрузите браузер Microsoft Internet Explorer.

1.6 Откройте в меню браузера Файл (File), Открыть (Open), Просмотр (Обзор - Browse) и найдите файл RASP.HTM и загрузите его.

Убедитесь, что название Web-страницы (Учебный файл HTML) отразилось в верхней статусной строке браузера.

1.7 Внесите изменения в текст файла HTML, расположив слова "Расписание", "занятий", "на вторник" на разных строках.

1.8 Сохраните внесенные изменения в файле RASP.HTM, с помощью команд Файл (File), Сохранить (Save). Закройте программу Блокнот (Notepad).

1.9 Просмотрите с помощью браузера Microsoft Internet Explorer новую полученную Web-страницу используя клавишу F5 или с помощью команд Вид(View), Обновить (Refresh). Проверьте, как изменилось изображение текста на экране.

1.10 Внесите изменения в текст файла HTML:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
Расписание<P>
занятий<BR>
на вторник
</BODY>
</HTML>
```

1.11 Внесите коррективы в размер и стиль шрифта согласно примеру и оцените полученные изменения:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<H1> Расписание </H1><I>занятий</I> <U> на вторник</U>
</BODY>
</HTML>
```

1.12 Измените размер шрифта для текста "занятия на вторник", используя тэг <FONT>.

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<FONT SIZE="7"> Расписание </FONT>
занятий на вторник
</BODY>
</HTML>
```

1.13 Измените текст HTML-документа, используя тэги выделения фрагментов текста и тэги перевода строки и абзаца:

1.14 Внесите изменения в файл RASP.HTM, изменив размер, цвет, гарнитуру, стиль текста документа:

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<U><I><B> <FONT COLOR="#FF0000" FACE="ARIAL" SIZE="7"> Расписание </FONT> <</B></I></U>
занятий на вторник
</BODY>
</HTML>
```

1.15 Осуществить форматирование текста в разных вариациях. С помощью тэгов HTML можно управлять горизонтальным выравниванием текста. Если не оговаривать способ выравнивания, все элементы в документе будут выравниваться по левому краю и иметь неровное правое поле. Современные браузеры для выравнивания текста используется атрибут ALIGN=, который встраивается в теги абзаца или заголовка (ALIGN=LEFT - выравнивание по левому краю; ALIGN=RIGHT - выравнивание по правому краю; ALIGN=CENTER - выравнивание по центру).

```
<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY>
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B> Расписание </B></FONT><BR>
<FONT SIZE="6"><I>занятий на вторник</I></FONT>
</P>
</BODY>
</HTML>
```

1.16 При изображении фона, текста браузеры используют цвета, установленные по умолчанию, - они заданы параметрами настройки браузера. Эти цвета устанавливаются в начале файла HTML в тэге <BODY...>. Запись цвета аналогична цвету шрифта. Атрибут BGCOLOR= задает цвет фона страницы, TEXT= определяет цвет текста для всей страницы, LINK= и VLINK= определяют цвета соответственно непрочитанных и прочитанных ссылок.

1.17 Проведите экспериментальную работу с созданным документом. Меняя размеры окна, посмотрите, как браузер показывает текст с принудительным разрывом строк. Что происходит, когда окно уменьшается настолько, что в нем не может поместиться целиком даже одно слово?



```

<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFCC" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B> Расписание </B></FONT><BR>
<FONT SIZE="6"><I>занятий на вторник</I></FONT>
</P>
</BODY>
</HTML>

```

### 1.3.5 Задание на лабораторную работу № 8

**Тема:** Язык HTML. Особенности работа с графическими объектами. Создание и редактирование таблиц.

**Цель работы:** приобрести навыки в создании персональных страниц, содержащих графику и таблицы.

#### *Теоретические сведения*

Размещение графики на Web-странице осуществляется с помощью тэга <IMG ...> позволяет вставить изображение в документ. Изображение появится в том месте документа, в котором записан этот тэг. Команда записывается с одиночным тэгом, т.е. закрывающий тэг не применяется.

Графика в Web, как правило, распространяется в трех форматах: GIF, JPG, PNG. Для выполнения задания считаем, что графический файл Kartinka.gif хранится в рабочем каталоге HTML, где находится исходная Web-страница.

```

<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B> Расписание </B></FONT><BR>
<FONT SIZE="6"><I>занятий на вторник</I></FONT><BR><BR>
<IMG SRC="Kartinka.gif">
</P>
</BODY>
</HTML>

```

Тэг <IMG ...> имеет немало атрибутов (таблица 1.5), которые

можно задавать дополнительно. Они могут располагаться где угодно в тэге после кода IMG.

Таблица 1.6 – Атрибуты изображения

Атрибут	Формат	Описание
ALT	<IMG SRC="Kartinka.gif" ALT="картина">	Если браузер не воспринимает изображение, вместо него появляется заменяющий текст
BORDER	<IMG SRC="Kartinka.gif" BORDER="3">	Задаёт толщину рамки вокруг изображения. Измеряется в пикселях
ALIGN	<IMG SRC="Kartinka.gif" ALIGN=TOP>	Выравнивает изображение относительно текста: по верхней части изображения - TOP, по нижней – BOTTOM, по средней – MIDDLE
HEIGHT	<IMG SRC="Kartinka.gif" HEIGHT=111>	Задаёт вертикальный размер изображения внутри окна браузера
WIDTH	<IMG SRC="Kartinka.gif" WIDTH=220>	Задаёт горизонтальный размер изображения внутри окна браузера
VSPACE	<IMG SRC="Kartinka.gif" VSPACE="8">	Добавляет верхнее и нижнее пустые поля
HSPACE	<IMG SRC="Kartinka.gif" HSPACE="8">	Добавляет левое и правое пустые поля

Фоновое изображение – это графический файл с изображением небольшой прямоугольной плашки. При просмотре в браузере эта плашка многократно повторяется, заполняя все окно, независимо от его размеров. Графика, используемая в качестве фоновой, задается в тэге <BODY> в начале файла HTML, как в примере ниже.

Таблицы представляют собой особую часть HTML-документа. Данные в ней организованы в виде прямоугольной сетки, состоящей из вертикальных столбцов и горизонтальных рядов. Каждая клетка таблицы является ячейкой.

Ячейки могут содержать в себе текст, графику или другую таблицу.

Таблица состоит из трех основных частей:

- а) название таблицы;
- б) заголовки столбцов;
- в) ячейки.

```

<HTML>
<HEAD>
<TITLE> Учебный файл HTML </TITLE>
</HEAD>
<BODY BACKGROUND="Зеленый камень.bmp" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE="7"><B> Расписание </B></FONT><BR>
<FONT COLOR="#FF0000" SIZE="6"> <I>занятий на вторник</I></FONT><BR><BR>
</P>
</BODY>
</HTML>

```

Таблица заполняется горизонтальными рядами ячейка за ячейкой слева направо. Заполнение начинается с левого верхнего угла и заканчивается правым нижним. Каждая ячейка должна быть заполнена. Для создания пустых ячеек используются пробелы.

Таблица 1.7 – Теги оформления таблиц

Тег	Форма записи	Примечание
TABLE	<TABLE>текст</TABLE>	Объявление таблиц
TR	<TR>текст</TR>	Тег строки
TD	<TD>текст</TD>	Тег данных

Таблица 1.8 – Атрибут тега <TABLE>

Атрибут	Форма записи	Примечание
BORDER	<TABLE BORDER=X>	Задаёт рамку вокруг таблицы
WIDTH	<TABLE WIDTH=XX%>	Задаёт ширину таблицы как XX% от ширины страницы или как XX пикселей
BGCOLOR	<TABLE BGCOLOR="#RRGGBB">	Задаёт цвет фона таблицы

Таблица 1.9 – Атрибуты тегов <TD> и <TR>

Атрибут	Форма записи	Примечание
ALIGN	<TD ALIGN=X>	Устанавливает выравнивание по горизонтали (Right, Left, Center)
VALIGN	<TD VALIGN=X>	Устанавливает выравнивание по вертикали
BGCOLOR	<TD BGCOLOR="#RRGGBB">	Задаёт цвет фона ячейки

Пример текста страницы, содержащей таблицы, представлен ниже.

```

<HTML>
<HEAD>
<TITLE> Расписание занятий факультета </TITLE>
</HEAD>
<BODY BGCOLOR="FFFFFF">
<P ALIGN=CENTER>
<FONT COLOR="RED" SIZE="6" FACE="ARIAL"><B> ЭМС-58 </B></FONT><BR>
</P>
<FONT COLOR="BLUE" SIZE="4" FACE="COURIER"><B> Понедельник </B></FONT><BR>
<TABLE BORDER="1" WIDTH=100% BGCOLOR="99CCCC">
<TR BGCOLOR="CCCCFF" ALIGN=CENTER>
<TD>Дисциплина</TD> <TD> ЭСА10-1  </TD> <TD> ЭСА10-2  </TD>
</TR>
<TR>
<TD>1</TD> <TD>Высшая математика</TD> <TD>Физика</TD>
</TR>
<TR>
<TD>2</TD> <TD>Информатика</TD> <TD>Англ. язык</TD>
</TR>
<TR>
<TD>3</TD> <TD>История</TD> <TD>Информатика</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

### *Индивидуальное задание*

Для произвольно выбранной тематики (технической направленности) создать файл HTML, содержащий элементы графики, таблицы на языке гипертекстовой разметки документов. В качестве базового файла можно использовать файл, созданный в предыдущей лабораторной работе.

В работе выполнить в файле такое форматирование:

а) разместить графический файл на странице, при этом стоит учитывать размеры (объем в байтах) графического файла, т.к. это влияет на время загрузки Web-страницы;

б) внести изменения в файл, опробовав использование таких атрибутов графики как ALT, BORDER, HEIGHT, WIDTH;

в) поменять фоновый рисунок страницы;

г) создайте на странице таблицу, содержащую, например, технические параметры какого-либо технического объекта – двигателя, станка и т.п. или создайте таблицу с расписанием занятий групп.

### 1.3.6 Задание на лабораторную работу № 9

**Тема:** Построение гипертекстовых связей. Создание простейшего мини-сайта.

**Цель работы:** изучить особенности создания документов с ссылками, приобрести навыки создания простейших сайтов.

#### ***Теоретические сведения***

Важнейшим свойством языка HTML является возможность включения в документы ссылок на другие документы. Возможны ссылки: на удаленный HTML файл, на некоторую точку в текущем HTML-документе, на любой файл, не являющийся HTML-документом. В качестве ссылки можно использовать текст или графику. Ссылки в пределах одного документа

Такие ссылки требуют наличие двух частей: метки и самой ссылки. Метка определяет точку, к которой происходит переход по ссылке. Ссылка использует имя метки. Ссылки выделяются цветом или подчеркиванием, в зависимости от того, как настроен браузер. Для изменения цвета ссылки используется атрибуты LINK= и VLINK= тэга <BODY ...>. Ссылка:

`<A HREF="#ПН">Понедельник</A>.`

Перед именем метки (ПН), указывающей куда производится ссылка, ставится символ #. Между символами > и < располагается текст (Понедельник), на котором производится щелчок для перехода по ссылке. Метка:

`<A NAME="ПН">Понедельник</A>.`

Также в документы включают ссылки на графические объекты.

#### ***Индивидуальное задание***

Создать простейший сайт, структура которого представлена ниже (можно использовать свободную тематику).

1 Слова «К юбилею ДГМА» сделать заглавием, выделив в отдельную строку, оформить их как заголовок 1-го уровня.

2 Задать цвет фона (серый) и цвета ссылок (непросмотренные - красный, просмотренные - синий).

3 Под заголовком создать таблицу из двух столбцов и одной строки. Задать рамку вокруг таблицы толщиной 5 пикселей (4,5 пт.), цвет темно-синий. Установить промежуток между столбцами 10 пикселей. Цвет

ячеек таблицы сделать желтым.

4 Поместить в левую часть рисунок Potapkin.bmp (портрет Потапкина В.Ф.), выровнять его по центру ячейки. Задать альтернативное текстовое имя (alt="П.П. Копняеву").

5 В правый столбец переместить остальной текст статьи и отформатировать его (фонт Areal).

6 К фрагменту текста "ДГМА" (в правом столбце таблицы) сделать гиперссылку на фотографию (файл u2.png), используя для этого тег <A>.

6 Ниже таблицы вставить горизонтальную линию и Текст2 (биографическая справка о Потапкине файл biogr.doc) и отформатировать его (Заголовок – стиль 2-го уровня, подчеркнуть, текст шрифтом Times New Roman).

8 Ниже текста биографической справки вставить горизонтальную линию размером 5 пикселей и Текст3 (файл President.doc).

9 Вставить метку № 1 (тег <A> с атрибутом name) на Заголовок биографической справки о Потапкине.

10 Сделать гиперссылку из основного заглавия ("К юбилею ДГМА" от слов "ДГМА", на метку № 1 (начало биографической справки), используя для этой цели инструмент гиперссылки, указав в качестве адреса имя метки.

11 Сохранить веб-страницу.

12 Проверить оформление веб-страниц и работоспособность ссылок в браузере.

Файл Potapkin.htm	
Заглавие документа (гиперссылка на Заглавие Текста2)	
Таблица:	
Портрет Потапкина В.Ф. (файл Potapkin.bmp )	Текст1 (файл Информация.doc) (ссылка на . ДГМА )»
Заглавие 2-го уровня (гиперссылка на Текст3) Текст2 (файл biogr)	
Горизонтальная черта  Текст3 (файл President)	

Файл photo2.gif Фотография ДГМА (файл u2.png)
---

## ***Контрольные вопросы***

1. Что такое компьютерная сеть?
2. Для чего создаются локальные сети ЭВМ?
3. Что такое сервер?
4. Что такое рабочая станция?
5. Какие сетевые технологии называются клиент-серверными?
6. Что такое сетевой адаптер? Какие типы сетевых адаптеров существуют?
7. Какие виды линий (каналов) используются для связи компьютеров в локальных сетях?
8. Какие методы доступа от компьютера к компьютеру используются в локальных сетях?
9. Какие бывают конфигурации ЛС?
10. Для чего создаются глобальные сети?
11. В чем признак глобальности сети?
12. Что такое «модем» и каковы его функции?
13. Какие виды обмена информацией приняты в электронной почте?
14. Какова структура электронного письма?
15. Что означают составляющие электронного адреса?
16. Какие виды работ поддерживает сеть Internet?
17. Принципы поиска информации в сети Internet?
18. Что такое гипертекст? С какой гипертекстовой программой вы работали и как в ней отражены принципы гипертекста?
19. Как ведется поиск информации в сети Internet?
20. Дополнительные сервисы сети Internet

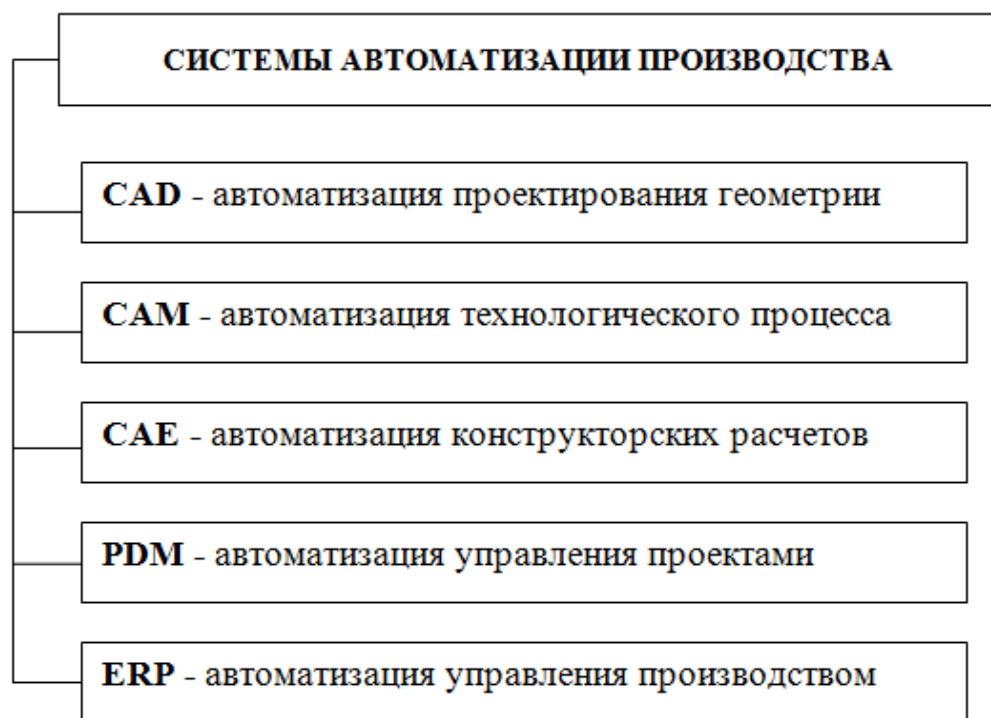
### ***1.4 Инструментальные программные средства***

Повышение уровня систем автоматизации производства, а, следовательно, и уровня сложности задач проектирования электромеханических систем, решаемых с помощью ЭВМ (рисунок 1.19), постоянно вызывает совершенствование технологий программирования, в частности изменения организации данных, которые обрабатывает программа [6].

При проектировании автоматизированных систем за основу все чаще принимают системный анализ. Естественно применение метода системного анализа и к программным системам. Одним из результатов развития технологических средств разработки программного обеспечения, основанных на данном методе и реализации общих функций, явилось бурное развитие CASE (Computer Aided Software / System Engineering) [7,8].

Разнообразие компьютерных средств поддержки анализа, проектирования, разработки, сопровождения и реинжиниринга в

настоящее время так велико, что сложно сформировать обобщающее определение этих средств.



*Рисунок 1.19 – Классификация систем автоматизации производства*

CASE средства оформились в самостоятельное научное направление и были основой для развития методологии структурного и объектно-ориентированного программирования.

Наиболее развитые средства CASE содержат:

- а) язык программирования четвертого поколения 4GL;
- б) средства управления проектом;
- в) средства автоматизации всех этапов жизненного цикла программного обеспечения [9].

Однако, применение таких средств шире, чем проектирование и разработка программного обеспечения. CASE средства включается и в состав инструментального и прикладного программного обеспечения.

Инструментальное программное обеспечение в различных областях развивается в направлении создания систем автоматизированного проектирования (САПР) CAD (Computer Aided Design), автоматизированных рабочих мест (АРМ) специалистов и др.

В соответствии с ГОСТ 24.003-84 АРМ – это комплекс информационных ресурсов, программно-технических и организационно-технологических средств индивидуального или коллективного пользования, объединённых для выполнения определённых функций профессионального работника. Основой разработки структуры АРМ является его проблемная ориентация на решение определённых



профессиональных задач, объединённых общей технологией обработки информации [10].

В состав современных САПР, АРМ и других интегрированных систем разработки входят, например, средства поддержки управления проектными данными (PDM), сохраняющие последовательные этапы разработки. Таким образом, различные программные средства взаимно обогащают друг друга средствами реализации общих функций.

В соответствии с общей тенденцией интеграции, наблюдаемой в системах, аналогичные процессы происходят и в CASE технологиях. Разработан проект стандарта на каркас (framework) для интеграции инструментальных средств CASE. Для основных форм интеграции: данных, контроля и представления, общая схема каркаса предусматривает услуги: пользовательского интерфейса, через который с использованием менеджера задач осуществляется доступ к инструментальным средствам. Услуги интеграции данных обеспечиваются встроенной базой данных, содержащей информацию о проекте – репозиторием [9].

Значительная доля использования инструментальных программных средств приходится на моделирование технических комплексов и систем разного уровня сложности.

Математическая модель выражает существенные черты объекта или процесса языком уравнений и других математических средств.

Путь математического моделирования в наше время гораздо более всеобъемлющ, нежели моделирования натурального. Огромный толчок развитию математического моделирования дало появление ЭВМ, хотя сам метод зародился одновременно с математикой тысячи лет назад.

Цели моделирования заключаются в следующем:

- а) понять сущность изучаемого объекта,
- б) научиться управлять объектом и определять наилучшие способы управления,
- в) прогнозировать прямые или косвенные последствия,
- г) решать прикладные задачи.

К классификации математических моделей разные авторы подходят по-своему, положив в основу классификации различные принципы.

Можно классифицировать модели по отраслям наук (математические модели в физике, биологии, социологии и т.д.) – это естественно, если к этому подходит специалист в какой-то одной науке. Можно классифицировать по применяемому математическому аппарату (модели, основанные на применении обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, стохастических методов, дискретных алгебраических преобразований и т.д.) – это естественно для математика, занимающегося аппаратом математического моделирования.

Цели моделирования предполагают такую классификацию моделей:

- а) дескриптивные (описательные) модели;
- б) оптимизационные модели;

- в) многокритериальные модели;
- г) игровые модели;
- д) имитационные модели.

При моделировании электромеханических систем особое внимание уделяется имитационным моделям, которые не просто отражают реальность с той или иной степенью точности, а имитируют ее. Эксперимент либо многократно повторяется, чтобы изучить и оценить последствия каких-либо действий на реальную обстановку, либо проводится одновременно со многими другими похожими объектами, но поставленными в разные условия. Подобный метод моделирования позволяет спрогнозировать поведение электромеханической системы в аварийных ситуациях.

Процесс компьютерного математического моделирования, включающий численный эксперимент с моделью можно представить в виде блок-схемы (рисунок 1.20).

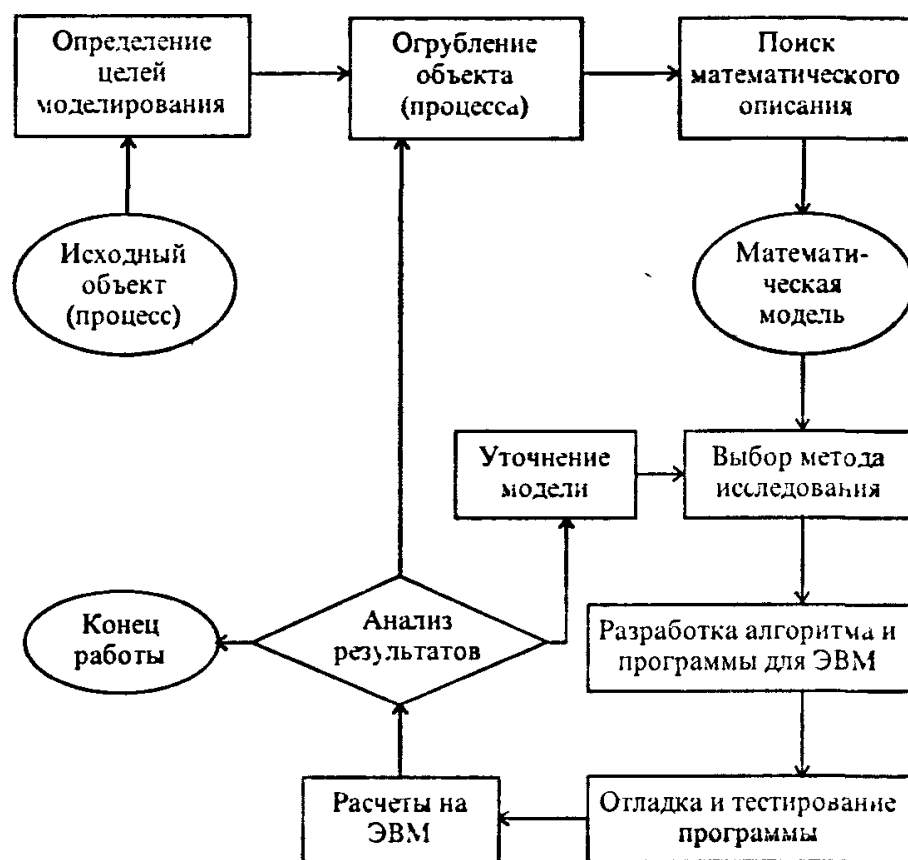


Рисунок 1.20 – Общая схема процесса компьютерного математического моделирования

С учетом временных характеристик модели можно разделить на статические, динамические и квазидинамические.

К статическим относят модели инвариантные относительно времени и служащие для описания процессов и явлений, независящих от времени.

Динамические модели не только допускают изменение параметров и структур во времени, но и служат для описания изменения процессов и моделей именно во времени. Поэтому в некоторых случаях применяют квазидинамические модели как упрощение динамических.

Квазидинамические модели – модели, в которых временной интервал действия модели разбивается на периоды, для каждого из которых строится статическая модель.

## 2 ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ СОВРЕМЕННЫЕ ПРОГРАММНЫЕ СРЕДСТВА ЭВМ

### 2.6 Matlab как среда моделирования и программирования высокого уровня

Matlab – пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете [11].

Первая версия пакета Matlab была разработана более 25 лет тому назад. Развитие и совершенствование этого пакета происходило одновременно с развитием средств вычислительной техники. Название пакета Matlab происходит от словосочетания Matrix Laboratory. Общий вид окна Matlab приведен на рисунке 2.1. Пакет ориентирован в первую очередь на обработку массивов данных (матриц и векторов) и к настоящему времени Matlab представляет собой богатейшую библиотеку функций (рисунок 2.2).

Для облегчения специалистам различных областей науки и техники работы с пакетом вся библиотека функций разбита на разделы. Те из них, которые носят более общий характер, входят в состав ядра Matlab. Те же функции, которые являются специфическими для конкретной области, включены в состав добавочных разделов, носящих название toolboxes. Полная комплектация пакета Simulink, например, содержит около 30 разделов инструментария.

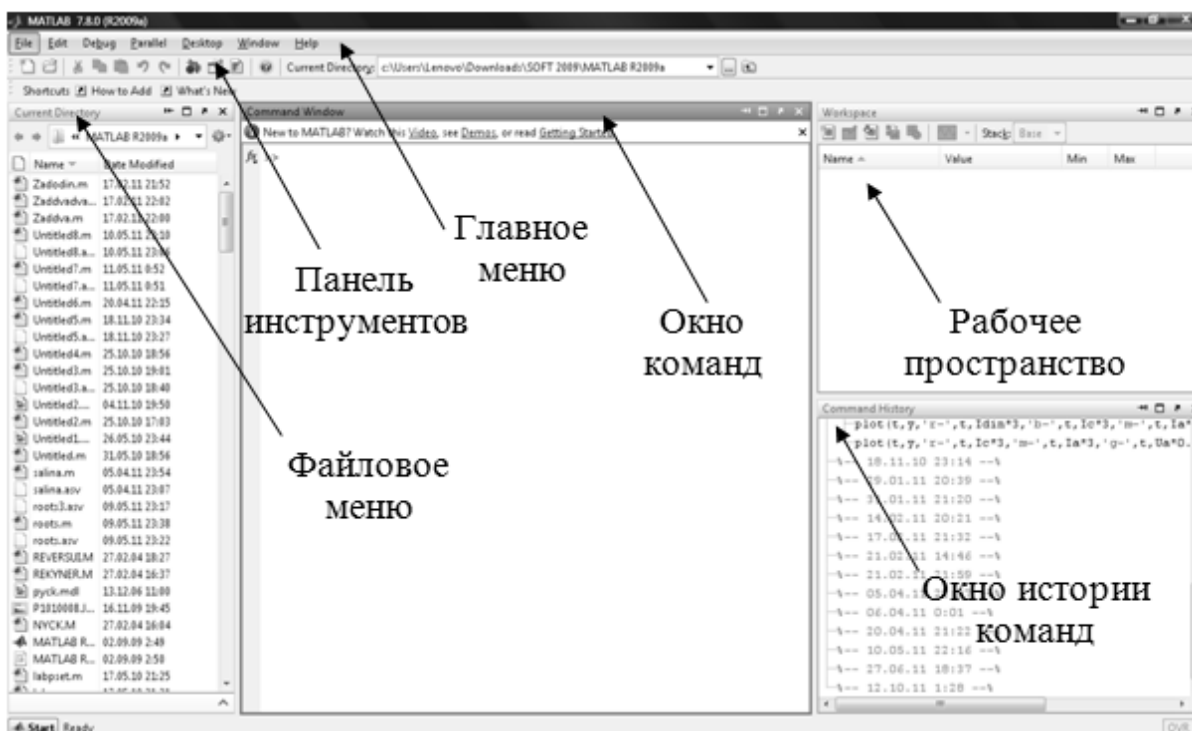


Рисунок 2.1 – Общий вид окна загрузки пакета Matlab

Matlab выполняет множество задач для поддержки научных и инженерных работ, начиная от сбора и анализа данных до разработки приложений. Среда Matlab объединяет математические вычисления, визуализацию и мощный технический язык [12].

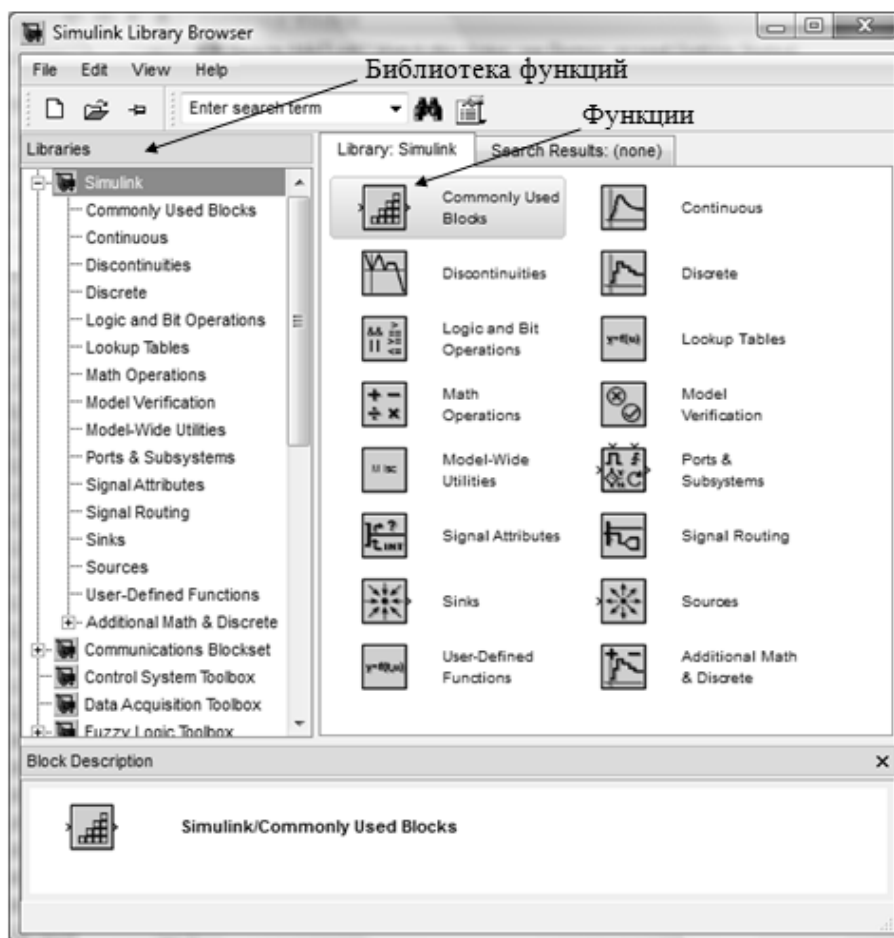


Рисунок 2.2 – Окно загрузки библиотеки функций Simulink Matlab

Встроенные интерфейсы позволяют получить быстрый доступ и извлекать данные из внешних устройств, файлов, внешних баз данных и программ. Кроме того, Matlab позволяет интегрировать внешние процедуры, написанные на языках Си, Си++, Фортран, и Java с пользовательскими Matlab приложениями (рисунок 2.3). Используемый повсюду в промышленности, государственных, академических и учебных организациях, Matlab фактически стал принятым во всем мире стандартом для технических вычислений.

Matlab имеет широкий спектр применений, включая цифровую обработку сигналов и изображений, проектирование систем управления, естественные науки, финансы и экономику, а также приборостроение. Открытая архитектура позволяет легко использовать Matlab и сопутствующие продукты для исследования данных и быстрого создания конкурентоспособных пользовательских инструментов [13].

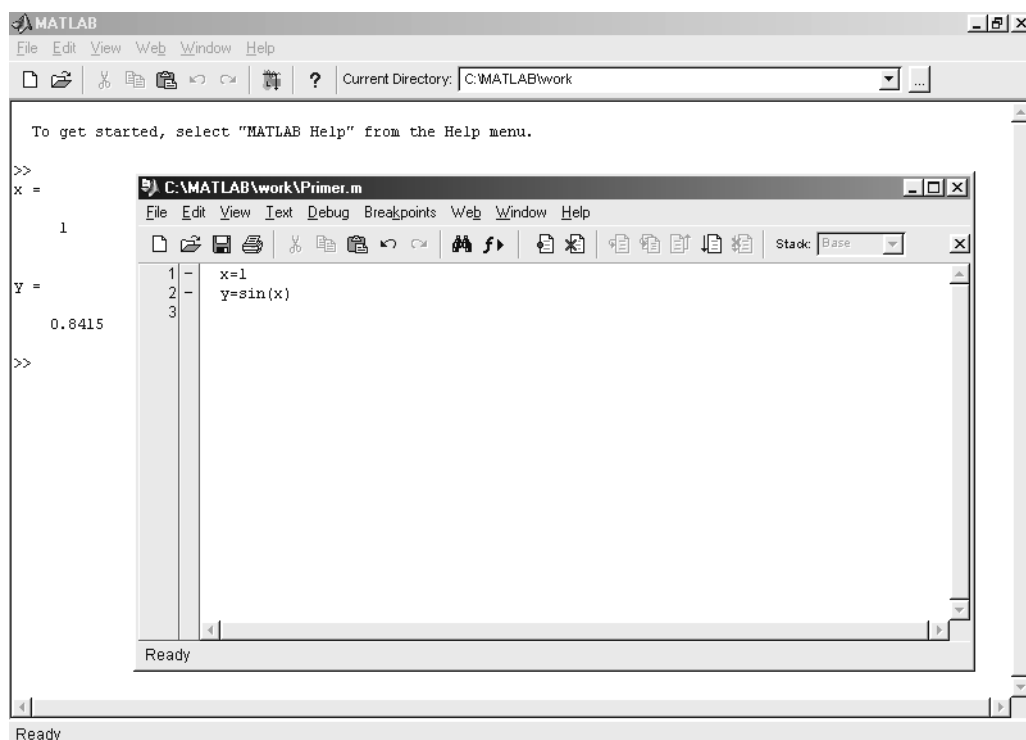


Рисунок 2.3 – Окно рабочего пространства и отладки файла-сценария

На сегодня Matlab является одним из индустриальных стандартов с широко развитыми дополнениями, из которых Simulink, как приложение Matlab, наиболее приспособлен для анализа и синтеза различных систем.

Пакет Simulink со своими дополнениями – основной инструмент изучения различных электромеханических систем. Практически нет ни одной задачи, связанной с исследованием систем электропривода, которую нельзя было бы решить в этом пакете.

Simulink предоставляет исследователю самые различные возможности, начиная от структурного (математического) представления системы и кончая генерированием кодов для программирования микропроцессора в соответствии со структурной схемой модели [14].

Simulink – это интерактивная система для имитационного моделирования динамических систем, создания моделей любого уровня сложности, стоящих из графических блоков с заданными свойствами (параметрами).

Она представляет собой среду, которая позволяет моделировать процесс путем перетаскивания блоков диаграмм на экране и их манипуляцией.

Simulink работает с линейными, нелинейными, непрерывными, дискретными, многомерными системами.

Основные средства для моделирования и анализа, имеющиеся в пакете Simulink [1]:

а) обширная библиотека блоков для создания линейных и нелинейных, дискретных и непрерывных, гибридных моделей;

- б) иерархическая структура моделей с неограниченной вложенностью;
- в) скалярные и векторные связи;
- г) средство для создания пользовательских блоков и библиотек;
- д) интерактивное моделирование с "живым" отображением на экране;
- е) семь методов интегрирования с фиксированным и переменным шагом;
- ж) линеаризация;
- з) определение точек равновесия;
- и) различные способы вывода на экран и библиотека входных сигналов.

При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты.

Simulink является достаточно самостоятельным инструментом MATLAB и при работе с ним не требуется знать сам MATLAB и остальные его приложения. С другой стороны доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink.

При моделировании пользователь может выбрать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Компоненты моделей являются графическими блоками и моделями, которые содержатся в ряде библиотек и с помощью мыши могут переноситься в основное окно и соединяться друг с другом необходимыми связями. В состав моделей могут включаться источники сигналов различного вида, виртуальные регистрирующие приборы, графические средства анимации.

Двойной щелчок мышью на блоке модели выводит окно со списком его параметров, которые пользователь может менять. Запуск имитации обеспечивает математическое моделирование построенной модели с наглядным визуальным представлением результатов. Пакет основан на построении блочных схем путем переноса блоков из библиотеки компонентов в окно редактирования создаваемой пользователем модели. Затем модель запускается на выполнение.

Возможно моделирование сложных систем, состоящих из множества подсистем. Simulink составляет и решает уравнения состояния модели и позволяет подключать в нужные ее точки разнообразные

виртуальные измерительные приборы. Основным нововведением является обработка матричных сигналов. Добавлены отдельные пакеты повышения производительности Simulink, такие как Simulink Accelerator для компиляции кода моделей, Simulink profiler для анализа кода и т. д.

Имеются обширные библиотеки блоков для разных областей применения. При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков. Simulink позволяет пополнять библиотеки блоков с помощью подпрограмм, написанных на языке MATLAB, C ++, и Fortran [14].

### 2.1.1 Задание на лабораторную работу № 10

**Тема:** Язык функциональных схем технических объектов средствами пакета Matlab.

**Цель работы:** научиться создавать функциональные схемы технических объектов языком приложений Matlab, приобрести навыки разработки схем с условными элементами.

#### *Теоретические сведения*

Для задания входных данных программы необходимо использовать блоки библиотеки Simulink Matlab (рисунок 2.4).

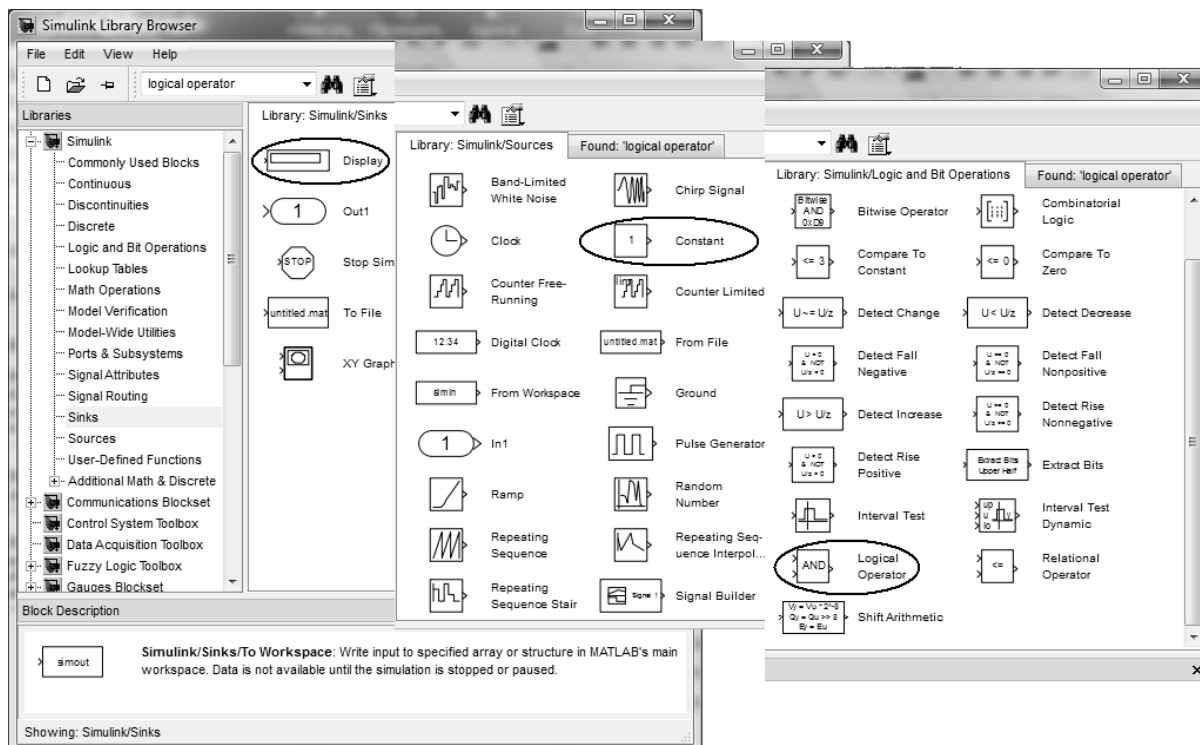


Рисунок 2.4 – Применяемые функции библиотеки Matlab



Блок Constant из библиотеки Simulink/Sources, для задания операций проверки работоспособности технической системы – блок Logical Operator из библиотеки Simulink/Logic and Bit Operations, блок Display из библиотеки Simulink/Sinks.

Изменения входных данных приводят к соответствующим изменениям выходного сигнала.

### *Ход работы:*

Решение задачи рассмотрим на примере системы освещения помещения, принципиальная электрическая схема которой показана на рисунке 2.5.



*Рисунок 2.5 – Принципиальная электрическая схема*

Для составления функциональной схемы проверки освещения:

1 Выберем и обозначим состояния элементов системы освещения независимыми логическими переменными (например, значение переменной  $s$  может указывать состояние ключа  $S$ : ключ замкнут ( $s$ =истина или  $s=1$ ), ключ разомкнут ( $s$ =ложь или  $s=0$ )).

2 Выберем и обозначим зависимую логическую величину, характеризующую работоспособность всей системы (например,  $y$  – состояние источника  $H$ : излучает свет/не излучает, истина/ложь).

3 В любой удобной форме запишем условие работоспособности схемы, например на естественном языке (русском, украинском, английском), в виде блок-схемы алгоритма и т.п. Для выбранной схемы освещения таким условием является исправность лампы  $H$  при замкнутом состоянии ключа  $S$  и подключенном работоспособном источнике электроэнергии  $E$ .

4 Представим записанное условие на языке функциональных схем.

Пример такой схемы показан на рисунке 2.6.

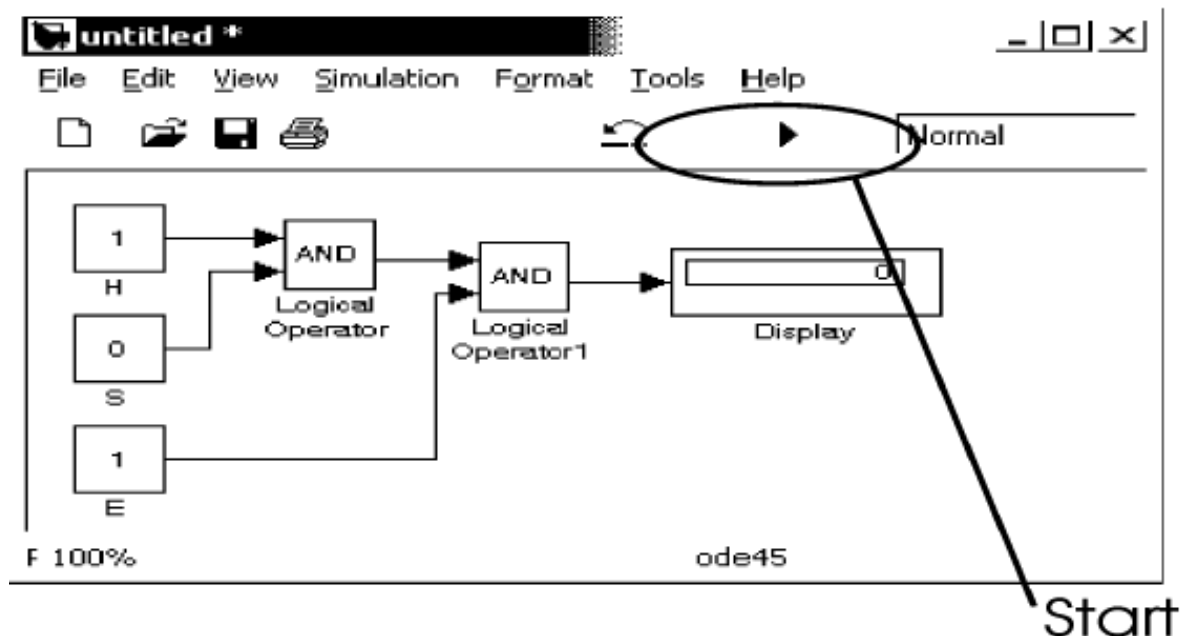


Рисунок 2.6 – Пример реализации функциональной схемы в среде пакета Matlab

### Индивидуальное задание

Используя язык функциональных схем, средствами пакета программ Matlab/Simulink разработать программу анализа работоспособности какой-либо самостоятельно выбранной технической системы (двигателя постоянного тока, токарного станка, прокатного стана, мостового крана, дробилка, стиральная машина и т.п.).

Число факторов, определяющих работоспособность системы, должно быть не менее четырех. При составлении схемы считать, что каждый элемент проверяемой технической системы снабжен датчиком его работоспособности, вырабатывающим двухуровневый логический сигнал («да/нет», «исправен/не исправен» – соответственно значение сигнала для анализа работоспособности будет «истина» или «ложь»).

#### 2.1.2 Задание на лабораторную работу № 11

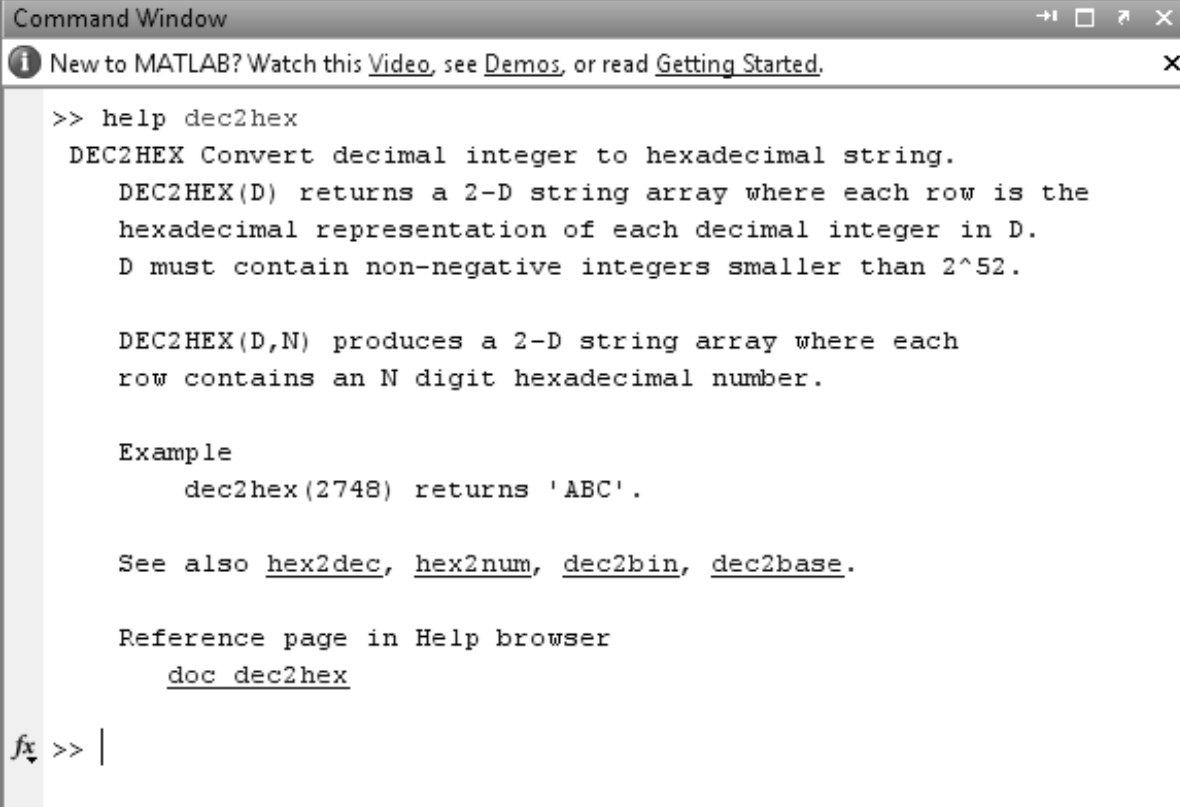
**Тема:** Системы счисления и поразрядные операции

**Цель работы:** приобрести навыки преобразования чисел в разных системах счисления, освоить способы автоматического преобразования чисел из одной системы счисления в другую разными средствами программного обеспечения.

## Теоретические сведения

В пакете Matlab для преобразования чисел в требуемую систему счисления служат функции `dec2hex()`, `dec2bin()`, `base2dec()`, `base2dec()` и ряд других. Сведения об их применении можно получить, воспользовавшись справочной системой пакета. Например, краткую справку по любой выбранной команде можно получить, введя в командной строке IDE пакета Matlab команду `help` имя где имя – это имя выбранной команды или функции [12].

Так, выполнение команды `help dec2hex` приведет к выводу на дисплей пояснений, представленных ниже на рисунке 2.7.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> help dec2hex
DEC2HEX Convert decimal integer to hexadecimal string.
DEC2HEX(D) returns a 2-D string array where each row is the
hexadecimal representation of each decimal integer in D.
D must contain non-negative integers smaller than 2^52.

DEC2HEX(D,N) produces a 2-D string array where each
row contains an N digit hexadecimal number.

Example
    dec2hex(2748) returns 'ABC'.

See also hex2dec, hex2num, dec2bin, dec2base.

Reference page in Help browser
    doc dec2hex

fx >> |
```

Рисунок 2.7 – Вывод справки по функции перевода числа в другую систему счисления

В этом тексте можно выделить три части:

- а) описание назначения и формата вызова команды;
- б) пример вызова команды;
- в) список других команд схожих по назначению.

Знакомство с командой следует начинать с выполнения в командном окне пакета Matlab приведенного в справке примера. Затем следует модифицировать команду для решения задачи, поставленной в задании на лабораторную работу.

Экспериментальную проверку выполнения логических операций следует проводить в следующем порядке:

- а) запустить программу «Калькулятор»;
- б) перевести «Калькулятор» в режим выполнения инженерных расчетов;
- в) перевести «Калькулятор» в режим работы с многоразрядными двоичными данными;
- г) выбрать разрядность обрабатываемых данных, например, один байт;
- д) ввести произвольно выбранную последовательность бинарных разрядов. При вводе данных следует учитывать, что лидирующие левые нули не вводятся и не отображаются программой на индикаторе;
- е) выполнить операцию инвертирования и убедиться в корректности полученного результата
- ж) аналогичным образом убедиться в правильности результатов операций логического сложения и умножения.

### ***Индивидуальное задание***

1 Перевести средствами Windows и пакета программ Matlab из десятичной системы в двоичную, восьмеричную, шестнадцатеричную системы счисления число N0N0N0N0, где N0 – младший разряд десятичного номера студента в журнале (номер в списке группы). Правильность перевода проверить обратным переводом.

2 Экспериментально убедиться в том, что логические операции, выполняемые входящей в состав операционной системы Windows программой «Калькулятор» (calc.exe) являются поразрядными (bitwise operation), используя комбинации операции «И», «ИЛИ», «НЕ» для произвольно выбранных пар чисел в разных системах счисления.

#### 1.2.3 Задание на лабораторную работу № 12

**Тема:** Matlab в задачах вычислительной математики. Табулирование функций.

**Цель работы:** изучить особенности интерфейса, надстройки, библиотечные функции, основные элементы языка программирования и визуализации расчетов вычислительной математики в системе Matlab.

### ***Теоретические сведения***

Matlab предоставляет широкие возможности работы с данными. Вычислительные операции можно реализовывать в рабочем пространстве Command Window, однако если работа из командной строки затруднена,

то самым удобным способом выполнения команд является использование М-файлов (рисунок 2.3), в которых можно набирать команды, выполнять их сразу или частями, сохранять в файле и использовать в дальнейшем.

Для работы с М-файлами предназначен специальный редактор, позволяющий создавать собственные функции и вызывать их, в том числе из командной строки. М-файлы в Matlab бывают двух типов: файл-программы (Script M-Files), содержащие последовательность команд, и файл-функции (Function M-Files), в которых описываются функции, определяемые пользователем. Файл-программу (файл-процедуру) Вы создали при прочтении предыдущего подраздела. Все переменные, объявленные в файл-программе, становятся доступными в рабочей среде после ее выполнения.

Переменные, определенные в одной файле-программе, можно использовать в других файл-программах и в командах, выполняемых из командной строки. Выполнение команд, содержащихся в файл-программе, осуществляется двумя способами: из редактора М-файлов так, как описано выше; из командной строки или другой файл-программы, при этом в качестве команды используется имя М-файла.

Применение второго способа намного удобнее, особенно, если созданная файл-программа будет неоднократно использоваться впоследствии. Фактически, созданный М-файл становится командой, которую понимает Matlab. Закройте все графические окна и наберите в командной строке `mydemo`, появляется графическое окно, соответствующее командам файл-программы `mydemo.m`. После ввода команды `mydemo` Matlab производит следующие действия.

Проверяет, является ли введенная команда именем какой-либо из переменных, определенных в рабочей среде. Если введена переменная, то выводится ее значение.

Если введена не переменная, то Matlab ищет введенную команду среди встроенных функций. Если команда оказывается встроенной функцией, то происходит ее выполнение [15].

Если введена не переменная и не встроенная функция, то Matlab начинает поиск М-файла с названием команды и расширением `m`. Поиск начинается с текущего каталога (Current Directory), если М-файл в нем не найден, то Matlab просматривает каталоги, установленные в пути поиска (Path). Найденный М-файл выполняется в Matlab.



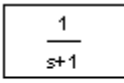







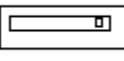













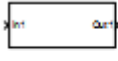





Как правило, М-файлы хранятся в каталоге пользователя. Для того чтобы система MatLab могла найти их, следует установить пути, указывающие расположение М-файлов. Хранить собственные М-файлы вне основного каталога Matlab следует по двум причинам. Во-первых, при переустановке Matlab файлы, которые содержатся в подкаталогах основного каталога Matlab, могут быть уничтожены. Во-вторых, при запуске Matlab все файлы подкаталога `toolbox` размещаются в памяти компьютера некоторым оптимальным образом так, чтобы увеличить

производительность работы. Если вы записали М-файл в этот каталог, то воспользоваться им можно будет только после перезапуска Matlab.

### *Индивидуальное задание*

1. Изучить особенности работы в среде Matlab и выполнить согласно варианту описание одного из элементов библиотеки.

*Таблица 2.1 – Варианты индивидуальных заданий*

№ вар.	Вид элемента библиотеки	№ вар.	Вид элемента библиотеки
1,2	 Derivative  Integrator	3,4	 Transfer Fcn  Transport Delay
5,6	 Hit Crossing  Quantizer	7,8	 Sign  Slider Gain
9,10	 Sum  Trigonometric Function	11,12	 Display  Floating Scope
13,14	 Scope  XY Graph	15,16	 Coulomb & Viscous Friction  Dead Zone
17,18	 Sine Wave  Step	19,20	 Dot Product  Gain
21, 22	 Relay  Saturation	23, 24	 Switch  Mux
25, 26	 Subsystem  Unit Delay	27, 28	 In1  Logical Operator
29, 30	 Constant  Demux		

2. Составить простейшую программу вычисления значений функции  $y_i$  для значений аргумента  $x_i$  с постоянным шагом изменения

аргументов (задача А) и произвольным набором значений аргументов (задача В). Варианты приведены в таблице 2.2.

Таблица 2.2 – Варианты индивидуальных заданий

№	Функция	Задача А					Задача В				
		а	в	х <sub>Н</sub>	х <sub>К</sub>	Δх	х <sub>1</sub>	х <sub>2</sub>	х <sub>3</sub>	х <sub>4</sub>	х <sub>5</sub>
1	$\frac{1 + \sin^2(b^3 + x^3)}{\sqrt[3]{b^3 + x^3}}$	-	2,5	1,28	3,28	0,4	1,1	2,4	3,6	1,7	3,9
2	$\frac{\sqrt[3]{ax+b}}{\lg^2 x}$	1,35	0,98	1,14	4,24	0,62	0,35	1,28	3,51	5,21	4,16
3	$\frac{1 + \lg^2 \frac{x}{a}}{b - e^{\frac{x}{a}}}$	2,0	0,95	1,25	2,75	0,3	2,2	3,78	4,51	6,58	1,2
4	$\sqrt[4]{ x^2 - 2,5 } + \sqrt[3]{\lg x^2}$	-	-	1,25	3,25	0,4	1,84	2,71	3,81	4,56	5,62
5	$\begin{cases} \frac{\lg^2(a^2 + x)}{(a+x)^2}, x > 5 \\ \frac{(a+bx)^{2,5}}{1,8 + \cos^3(ax)}, \geq 5 \end{cases}$	-2,5	3,4	3,5	6,5	0,6	2,89	3,54	5,21	6,28	3,48
6	$\begin{cases} 1,2^x - x^{1,2},  x  \geq 1 \\ \arccos x,  x  < 1 \end{cases}$	-	-	0,2	2,2	0,4	0,1	0,9	1,2	1,5	2,3
7	$\frac{a^x - b^x}{\lg \frac{a}{b}} \sqrt[3]{ab}$	0,4	0,8	3,2	6,2	0,6	4,48	3,56	2,78	5,28	3,21
8	$\frac{b^3 + \sin^2 ax}{\arccos(xbx) + e^{-x/2}}$	1,2	0,48	0,7	2,2	0,3	0,25	0,36	0,56	0,94	1,28
9	$\frac{\lg(x^2 - 1)}{\log_5(ax^2 - b)}$	1,1	0,09	1,2	2,2	0,2	1,21	1,76	2,53	3,48	4,52
10	$\frac{\arccos(x^2 - b^2)}{\arcsin(x^2 - a^2)}$	0,05	0,06	0,2	0,95	0,15	0,15	0,26	0,37	0,48	0,56
11	$\arcsin(x^a) + \arccos(x^b)$	2,0	3,0	0,11	0,36	0,05	0,08	0,26	0,35	0,41	0,53
12	$a^{x^2-1} - \lg(x^2 - 1)$	1,6	-	1,2	3,7	0,5	1,28	1,36	2,47	3,68	4,56
13	$\frac{a\sqrt{x} - b \log_5 x}{\lg x-1 }$	4,1	2,7	1,2	5,2	0,8	1,9	2,15	2,34	2,73	3,16
14	$\sqrt{\frac{ a-bx }{\lg^3 x}}$	7,2	4,2	1,81	5,31	0,7	2,4	2,8	3,9	4,7	3,16
15	$(\arcsin^2 x + \arccos^4 x)^3$	-	-	0,26	0,66	0,08	0,1	0,35	0,4	0,55	0,6

Продолжение таблицы 2.2.

	Функция	Задача А					Задача В				
		а	в	х <sub>Н</sub>	х <sub>К</sub>	Δх	х <sub>1</sub>	х <sub>2</sub>	х <sub>3</sub>	х <sub>4</sub>	х <sub>5</sub>
16	$\frac{\ln_a  b^2 - x^2 }{\sqrt[5]{ x^2 - a^2 }}$	2,0	1,1	0,08	1,08	0,2	0,1	0,3	0,4	0,45	0,65
17	$\frac{a + tg^2 bx}{b + ctg^2 ax}$	0,1	0,5	0,15	1,37	0,25	0,2	0,3	0,44	0,6	0,56
18	$\frac{(a + bx)^{2,5}}{1 + \lg(a + bx)}$	2,5	4,6	1,1	3,6	0,5	1,2	1,28	1,36	1,46	2,35
19	$\frac{\lg^2(a + x)}{(a + x)^2}$	2,0	-	1,2	4,2	0,6	1,16	1,32	1,47	1,65	1,93
20	$\frac{\sqrt[3]{(x-a)^2} + \sqrt[5]{ x+b }}{\sqrt[9]{x^2 - (a+b)^2}}$	0,8	0,4	1,23	7,23	1,2	1,88	2,26	3,84	4,55	6,21
21	$(\sin^3 x + \cos^3 x) \ln x$	-	-	0,11	0,36	0,05	0,2	0,3	0,38	0,43	0,57
22	$a^{x^2-1} - \lg(x^2 - 1)$	2,25	-	1,2	2,7	0,3	1,31	1,39	1,44	1,56	1,92
23	$\frac{a^3 \sqrt{x} - b \log_5 x}{\lg^3(x-1)}$	4,1	2,7	1,5	3,5	0,4	1,9	2,15	2,34	2,74	3,16
24	$\sqrt[5]{\frac{a+bx}{\lg^3 x}}$	7,2	1,3	1,56	4,71	0,63	2,4	2,8	3,9	4,7	3,16
25	$\sqrt[7]{\arcsin^4 x + \arccos^6 x}$	-	-	0,22	0,92	0,14	0,1	0,35	0,4	0,55	0,6
26	$\frac{\log_a(b^2 - x^2)}{\sqrt[3]{ x^2 - a^2 }}$	2,0	4,1	0,77	1,77	0,2	1,24	1,38	2,38	3,21	0,68
27	$\frac{\sqrt[3]{a + tg^{4,5} bx}}{\sqrt[5]{b + ctg^{2,7} ax}}$	0,1	0,5	0,33	1,23	0,18	0,5	0,36	0,4	0,62	0,78
28	$\frac{\sin(a + bx)^{3,5}}{1 + \cos[\lg(a + bx)]}$	2,5	4,6	1,15	1,15	3,05	0,38	1,2	1,36	1,57	2,25
29	$\frac{tg[\lg^3(a + x)]}{\sqrt[7]{(a + x)^2}}$	2,0	-	1,08	1,88	0,16	1,16	1,35	1,48	1,52	1,96
30	$\frac{\sqrt[3]{x-a} + \sqrt[5]{x+b}}{\sqrt[7]{x} - \sqrt[9]{x^2 - (a+b)^2}}$	0,8	0,4	1,42	3,62	0,44	1,6	1,81	2,24	2,65	3,38



## 2.1.4 Задание на лабораторную работу № 13

**Тема:** Простейшие вычисления в среде пакета Matlab.

**Цель работы:** ознакомиться с математическими операциями и функциями для создания файлов-сценариев Matlab, приобрести навыки выполнения простейших вычислений.

### *Теоретические сведения*

При вычислениях в Matlab применяются стандартные операторы и функции, перечень некоторых из них приведен в таблице 2.3.

Таблица 2.3

Синтаксис	Функция
sin	Синус
asin	Арксинус
cos	Косинус
acos	Арккосинус
tan	Тангенс
atan	Арктангенс
cot(x)	Котангенс
abs	Абсолютное значение комплексного числа
conj	Комплексно-сопряженное число
imag	Мнимая часть комплексного числа
real	Действительная часть комплексного числа
exp	Экспоненциальная функция
log	Функция натурального логарифма
log10	Логарифм по основанию 10
log2	Логарифм по основанию 2
pow2	Экспонента по основанию 2
sqrt	Функция квадратного корня
x.^y	Возведение X в степень Y

### *Индивидуальное задание*

Для знакомства с основными математическими конструкциями при создании файлов-сценариев Matlab, выполнить простейшие вычисления.

Вычислить чему равно:  $\sin 30^\circ$ ,  $\sin 40^\circ$ ,  $\cos 60^\circ$ ,  $\cos 180^\circ$ .

Вычислить:  $\log_{0,5} 8$ ,  $\lg 100$ ,  $\log_3 5$ ,  $0,04 \log_{0,2} 1$ .

Вычислить:  $e^x$ , для  $x = [1 \ 2 \ 3]$ .

В соответствии с номером N по списку в журнале группы, записанном в виде  $N=CM$ , где C – старшая цифра, M – младшая цифра, в

целочисленном интервале  $[N, N+5]$  рассчитайте таблицу значений для выражения, заданного в таблицах 2.4 и 2.5 соответственно.

Таблица 2.4

Младшая цифра	0 или 5	1 или 6	2 или 7	3 или 8	4 или 9
Выражение	$\frac{A^{2/3} * \sqrt{C^3}}{(B + D)^2}$	$\frac{B * C}{A^2 + D^2}$	$\frac{A * \sqrt{B}}{D^{1/3} / C^2}$	$\frac{C^{2/5} + \sqrt{D}}{B^{1/3} * \sqrt[5]{A^2}}$	$\frac{A * C * \sqrt[3]{D}}{C^{2/3} + B^2}$

Таблица 2.5

Старшая цифра	A	B	C	D
0	$(\sin(N)\cos(N))^2$	$(\ln(N+2))/N$	$(\exp(N)/N)^{-2}$	A+B
1	$1+(\cos(N+1))^2$	$(\ln(N^2))^{-2}$	$\exp(-N)+1$	A+C
2	$(\text{tg}(N))^2 + N^{-1}$	$(\log_2(N))^{-2}$	$1+N/\exp(N)$	B/C
3	$\text{Sin}(N)/(\text{ctg}(N))$	$(\log_2(N^{-3}+N^3))$	$n/\exp(-N)$	C/(A+B)

### 2.1.5 Задание на лабораторную работу № 14

**Тема:** Вычисление значений логических выражений.

**Цель работы:** приобрести навыки обработки логических данных, научиться составлять алгоритмы решения логических задач.

#### **Теоретические сведения**

В общем случае трансляция программы на языке программирования может выполняться в двух режимах:

а) интерактивный (диалоговый), при котором инструкции программы вводятся пользователем и выполняются по одной с выдачей результата каждой инструкции на устройство отображения информации. Этот режим используется при разработке программ;

б) пакетный (batch mode), при котором инструкции объединяются в пакет (записываются в файл) и затем выполняются все вместе. Этот режим применяют при многократном использовании программ.

Пакетная трансляция выполняется в следующем порядке:

- а) создание текста, содержащего инструкции программы;
- б) сохранение текста программы в файле;
- в) трансляция сохраненного в файле текста и исполнение программы.

Первый и второй этап трансляции выполняют с помощью редакторов программ. Набранный в редакторе текст программы сохраняют в файле, имя которого имеет расширение, отделяемое точкой и соответствующее использованному языку программирования: \*.m – командный язык пакета Matlab; \*.vbs - язык VBScript.

Для создания текста программ можно использовать текстовый редактор Блокнот, специализированные редакторы программ (EditPlus, CrimsonEditor) или редактор, входящий в состав IDE соответствующего транслятора.

После создания текстового файла с программой ее необходимо транслировать с помощью транслятора. Вызов транслятора и указание ему имени файла можно выполнить двумя способами:

- а) с помощью командного интерпретатора операционной системы;
- б) в ходе интерактивного (диалогового) взаимодействия с IDE транслятора.

Например, для пакета Matlab можно:

а) запустить трансляцию m-фала из командной строки IDE пакета MATLAB, используя команду run 'имя', где имя – это введенное без расширения .m имя созданного ранее текстового файла с программой;

б) запустить трансляцию m-фала из Проводника (Windows Explorer) с помощью пункта RUN контекстного меню ярлыка файла или

в) выполнить команду matlab -г имя из главного меню Windows (пункт меню «Выполнить») или из командной строки различных файловых менеджеров, например, FAR и Total Commander.

При составлении программ следует иметь ввиду, что разные языки программирования отличаются не только синтаксисом выражений (операторная форма, функциональная форма и т.п.) и обозначениями операций, но и обозначениями логических значений. Примеры обозначений логических констант на разных языках даны в таблице 2.6.

*Таблица 2.6*

Язык	Истина	Ложь
Matlab	1	0
VBScript	1 или True	0 или False
Excel	1 или ИСТИНА	0 или ЛОЖЬ

Варианты обозначений логических операций сведены в таблицу 2.7.

*Таблица 2.7*

Язык	Инвертирование	Сложение	Умножение
Matlab	~		&
VBScript	Not	Or	And
Excel	НЕ	ИЛИ	И

В зависимости от используемого языка программирования для связывания идентификаторов с логическими значениями используется либо операция присваивания (например, = в языках C, Basic и в командном языке пакета Matlab), либо специальные инструкции.

Следует так же иметь ввиду то, что результат вычислений не всегда автоматически выдается на устройство отображения информации. Иногда он сохраняется в памяти компьютера и для его вывода на печать требуется применение дополнительных инструкций вывода, например, MSGBOX на языке VBScript, disp на языке пакета Matlab.

В общем случае программа содержит следующие структурные составляющие:

- а) определение переменных (связывание идентификаторов с логическими значениями);
- б) вычисление значений выражений;
- в) вывод результата вычислений на устройство отображения информации.

В ряде случаев первая и третья составляющие могут отсутствовать (они добавляются транслятором автоматически в ходе трансляции). В программу могут вставляться комментарии программиста. Они не подвергаются трансляции, а служат для пояснения смысла программы на естественном языке, например, русском. От инструкций программы комментарии отделяются специальными символами, например: % – на командном языке пакета Matlab; ‘ – на языке VBScript.

Например, для контроля безопасности движения транспортного средства система должна блокировать работу двигателя транспортного средства при открытых дверях, отсутствии пассажиров и не пристегнутых ремнях безопасности.

Обозначим S1, S2 и S3 – логические сигналы датчиков положения дверей (S1=1 – двери закрыты), присутствия пассажиров (S2=1 – пассажиры присутствуют) и состояния ремней безопасности (S3=1 – ремни пристегнуты). Motor – логический сигнал управления двигателем.

Анализ задачи показывает, что во включенном состоянии система контроля проверяет сигналы трех датчиков и формирует по результатам каждой проверки один из двух возможных уровней выходного управляющего сигнала.

На языке пакета Matlab выполняем следующее:

а) в рабочем каталоге создаем файл с именем D:\work\matlab\lab3.m, содержащий текст программы:

```
% Комментарий: решение логической задачи на языке Matlab
S1=1; S2=1; S3=0; % задаем логические значения
Motor=S1 & S2 & ~S3; % рассчитываем значение логического выражения
disp(Motor); % и печатаем его
```

б) Запускаем пакет Matlab;

в) В командной строке пакета Matlab вводим команду run ‘D:\work\matlab\lab’ или ее сокращенный вариант lab3.

На языке VBScript выполняем следующее:

а) В рабочем каталоге создаем файл с именем D:\work\lab.vbs, содержащий программу:

```
S1=1 'Определяем независимые переменные
```

```
S2=1
```

```
S3=0
```

```
Y=S1 and S2 and not S3 'вычисляем
```

```
msgbox Y 'печатаем;
```

б) Из главного меню Windows выбираем пункт «Выполнить» и вводим команду cscript D:\WORK\lab.vbs

Пример работы программы на языке VBScript приведен на рисунке 2.8.

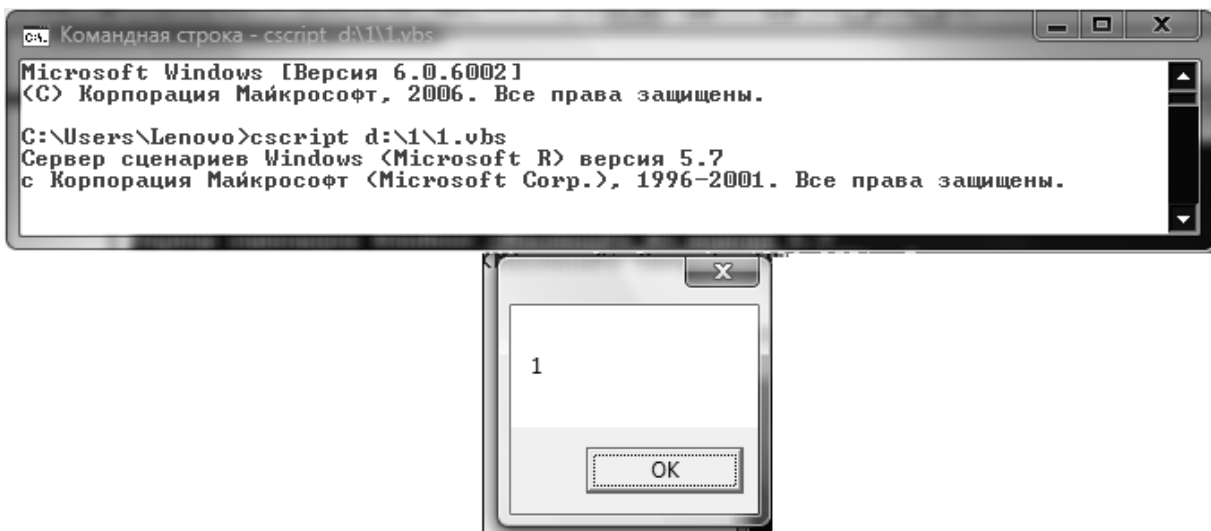


Рисунок 2.8 – Результат работы программы lab.vbs

При проверке работоспособности всех составленных программ необходимо использовать разные наборы значений независимых логических переменных. Результаты выполнения программ для этих наборов необходимо сравнить с ожидаемыми значениями логического выражения.

### ***Индивидуальное задание***

Используя стандартные логические операции, записать программу решения логической задачи произвольно выбранной предметной области на следующих языках программирования высокого уровня: на языке VBScript; на командном языке пакета Matlab. В случае необходимости запись логического выражения в программе следует предварить определением значений логических переменных и дополнить операциями вывода вычисленного значения логического выражения.

Все составленные программы следует сохранить в текстовых файлах с именами, имеющими соответствующие расширения (например,

расширение \*.vbs для языка Windows VBScript), для дальнейшего использования.

Трансляция текстов программ в исполняемый код может быть выполнена, например, следующими трансляторами:

а) для командного языка пакета Matlab – MATLAB Version 6.5.0.180913a (R13)

б) для языка VBScript – сервер сценариев Windows (cscript.exe или wscript.exe) или IDE vbsedit.

## 2.1.6 Задание на лабораторную работу №15

**Тема:** Функциональная форма записи логических выражений.

**Цель работы:** приобрести навыки решения логических задач, преобразования в функциональную форму записи логических выражений средствами различных приложений.

### *Теоретические сведения*

Создание текста программ выполняют с помощью редакторов текстов программ – общего назначения, например Блокнот (notepad.exe), или специализированных (Crimson Editor, EditPlus и т.п.). Текст на языке формул электронных таблиц записывается непосредственно в ячейку электронной таблицы. Набранный текст программы сохраняют в файле, имя которого имеет отделяемое точкой расширение, соответствующее использованному языку программирования: \*.m – командный язык пакета Matlab и \*.xls – язык электронных таблиц Microsoft Excel, \*.mcd – в пакете Mathcad.

На языке пакета Matlab последовательность действий будет следующей:

а) в рабочем каталоге создаем файл, например, с именем D:\work\matlab\lab11.m, содержащий текст программы:

```
% Комментарий: решение логической задачи на языке Matlab
S1=1; S2=1; S3=0; % задаем логические значения
Motor=and(S1,and( S2, not(S3))); % рассчитываем значение
логического выражения
```

```
disp(Motor) ; % и печатаем его
```

б) запускаем пакет Matlab;

в) в командной строке пакета Matlab вводим команду

```
run 'D:\work\matlab\lab11' или ее сокращенный вариант lab11.
```

На языке формул электронных таблиц MS Excel выполняется следующее:

а) запускаем MS Excel или аналогичную программу из пакета Open Office;

б) в ячейки A1, B1, C1 вводим исходные значения независимых логических переменных;

в) в ячейку B1 вводим выражение логической функции  $=И(A1; И(B1; НЕ(C1)))$

г) сохраняем результат в файле с именем «D:\work\lab11.xls».

Средствами пакета Mathcad рабочий файл поиска решения логической задачи представлен на рисунке 2.9.

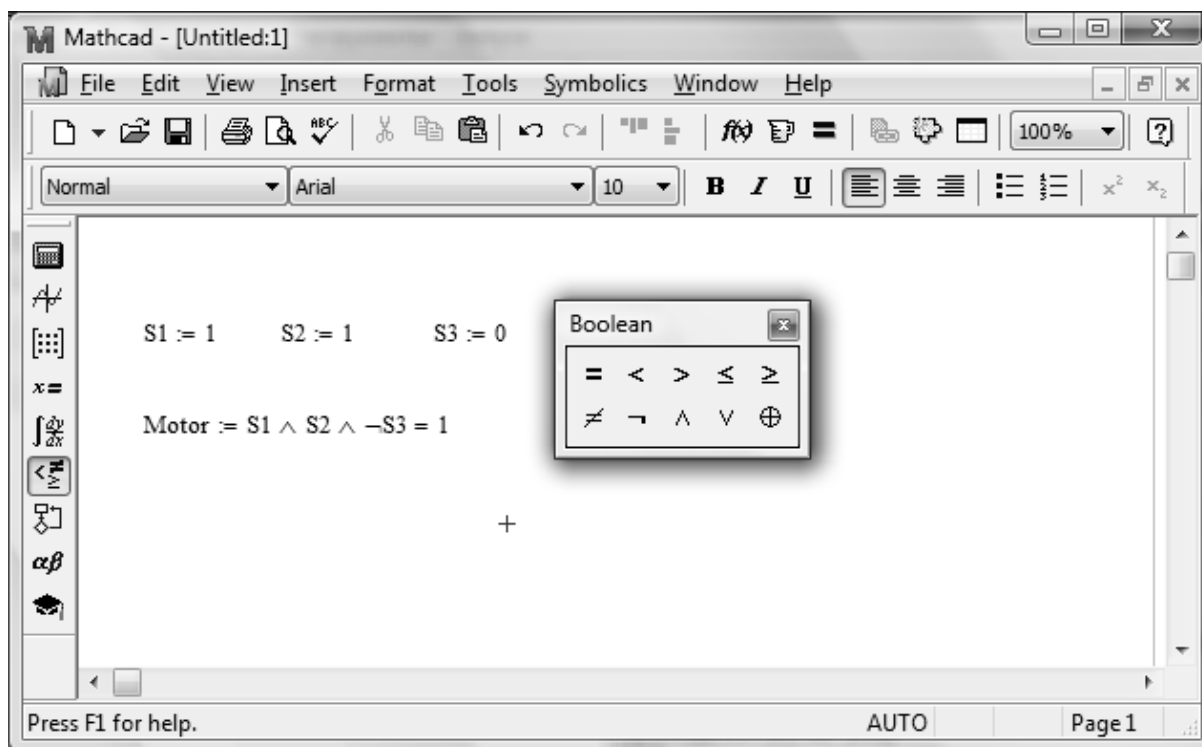


Рисунок 2.9 – Решение логической задачи средствами Mathcad

### **Индивидуальное задание**

Используя функциональную форму записи логических выражений создать программы решения логической задачи, сформулированной при выполнении предыдущей лабораторной работы, на следующих языках программирования высокого уровня: на командном языке пакета Matlab и на языке формул пакета электронных таблиц Microsoft Excel, средствами математического пакета Mathcad.

#### 2.1.7 Задание на лабораторную работу №16

**Тема:** Решение систем линейных алгебраических уравнений.

**Цель работы:** приобрести навыки обработки матриц и выполнения операций математической алгебры с помощью математического пакета Matlab.

## Теоретические сведения

В электротехнических расчетах часто используют матрицы и векторы. В системе Matlab реализованы два типа арифметических операций: поэлементные и матричные. Матричные операции (+, -, \*, /, \, ^) применяются согласно с правилами линейной алгебры. Поэлементные операции (+, -, \*, /, .^ ) применяются только для матриц одинаковой размерности или в случае, когда один из операндов является скаляром.

Нумерация элементов массивов в Matlab начинается с единицы.

Суммой или разницей двух матриц одинаковой размерности является матрица такой же размерности  $C = A \pm B$ , элементы которой равны сумме или разности соответствующих элементов матриц A и B:

$$c_{ij} = a_{ij} + b_{ij}; (i = 1, 2, \dots, m; j = 1, 2, \dots, n).$$

Произведением матрицы A размерностью  $m \times k$  и матрицы B размерностью  $k \times n$  является матрица C размерностью  $m \times n$ , элементы которой определяются по формуле:

$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{ik} \cdot b_{kj}.$$

Умножение матриц возможно только тогда, когда число столбцов матрицы A равно количеству строк матрицы B, однако в общем случае для матриц неприменим закон перемещения.

Частным случаем произведения матриц является умножение матрицы на вектор-столбец или вектор-строку, при этом размерность столбцов и строк должны совпадать.

Квадратную матрицу можно возвести в степень – эта операция в Matlab реализована символом «^».

Также предусмотрены операции правостороннего и левостороннего деления матриц:

$$B/A \text{ соответствует } B \cdot A^{-1};$$

$$A \setminus B \text{ соответствует } A^{-1} \cdot B,$$

где  $A^{-1}$  – обратная матрица.

Решение систем линейных уравнений типа  $X = A \setminus B$  реализовано в Matlab с помощью специального монитора, который использует различные алгоритмы в зависимости от структуры матрицы A.

Транспонирование матрицы  $A^T$  предполагает замену строк соответствующими столбцами и в Matlab реализовано операцией «'» (апостроф) [16].

Обращение матриц – одна из наиболее распространенных операций матричного анализа и широко используется при расчете



разветвленных электрических цепей разными методами. Обратной относительно выходной квадратной матрицы называется такая квадратная матрица  $X_{n \times n} = A^{-1}_{n \times n}$ , которая при умножении на выходную матрицу дает единичную диагональную матрицу  $E_{n \times n}$ :

$$A_{n \times n} \cdot A^{-1}_{n \times n} = E_{n \times n}.$$

Таким образом, для определения всех элементов обратной матрицы необходимо найти решение  $n$  систем уравнений. Этот подход часто применяется при машинных расчетах. В Matlab для определения обратной матрицы применяется функция `inv(A)`. Для расчета определителя матрицы применяется функция `Matlab det(A)`.

К решению систем линейных уравнений сводятся многочисленные практические задачи, например, различные краевые задачи для обыкновенных и в частных производных дифференциальных уравнений.

Пусть в общем случае задана система линейных уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2; \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases}$$

В общем виде данная система будет иметь вид в матричной форме  $AX=B$ , где  $A$  – квадратная матрица коэффициентов,  $X$  – вектор-столбец неизвестных,  $B$  – вектор-столбец правой части уравнения.

Способы решения систем линейных уравнений делятся на точные (метод обращения матрицы, метод Крамера, метод Гаусса) и итерационные (Ньютона, Зейделя, простых итераций).

Если определитель матрицы не равен нулю, то система имеет одно решение вида  $X = A^{-1}B$ .

Одним из наиболее простых и эффективных методов решения является метод Гаусса и его модификации. Алгоритм метода основан на приведении матрицы  $A$  к треугольному виду (прямой ход) и последовательном вычислении неизвестных (обратный ход). Недостатком метода является накопление погрешностей в процессе округления, поэтому метод Гаусса без выбора главных элементов используется обычно для решения небольших ( $n < 100$ ) систем уравнений с плотно заполненной матрицей и не близком к нулю определителем.

Для решения систем линейных уравнений в арсенале Matlab предусмотрены функции правого деления – «/» ( $X=B/A$ ), левого деления – «\» ( $X=A \setminus B$ ), обратной матрицы ( $X=inv(A) \cdot B$ ;  $X=A^{-1} \cdot B$ ).

### ***Индивидуальное задание***

Согласно данным таблицы 2.8 выполнить следующее:

а) найти определитель матрицы  $A$ ;

- б) найти решение системы линейных алгебраических уравнений;  
 в) выполнить проверку результатов решения;  
 г) вычислить скалярное произведение выходной матрицы коэффициентов А на обратную (результатом должна быть единичная диагональная матрица);  
 д) вычислить значения невязок ( $f = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - b_i$ ).

Таблица 2.8 – Варианты индивидуальных заданий

№	Матрицы А и В	№	Матрицы А и В
1	$a_{ij} = \begin{bmatrix} 2 & -4 & 3 & 1 \\ -1 & 5 & -7 & -3 \\ 10 & -2 & 4 & 4 \\ -1 & 1 & -1 & -1 \end{bmatrix}; b_i = \begin{bmatrix} 7 \\ -24 \\ 34 \\ -6 \end{bmatrix}$	2	$a_{ij} = \begin{bmatrix} 5 & -3 & 4 & -2 \\ 10 & 3 & -4 & 2 \\ 7 & -5 & 8 & -10 \\ 4 & 5 & -8 & 10 \end{bmatrix}; b_i = \begin{bmatrix} 4 \\ 11 \\ 0 \\ 11 \end{bmatrix}$
3	$a_{ij} = \begin{bmatrix} 2 & 2 & -1 & 1 \\ 4 & 3 & -1 & 2 \\ 8 & 5 & -3 & 4 \\ 3 & 3 & -2 & 2 \end{bmatrix}; b_i = \begin{bmatrix} 4 \\ 6 \\ 12 \\ 6 \end{bmatrix}$	4	$a_{ij} = \begin{bmatrix} 1 & 1 & -6 & 4 \\ 3 & -1 & -6 & -4 \\ 2 & 3 & 9 & 2 \\ 3 & 2 & 3 & 8 \end{bmatrix}; b_i = \begin{bmatrix} 6 \\ 2 \\ 6 \\ -7 \end{bmatrix}$
5	$a_{ij} = \begin{bmatrix} 1 & 5 & 3 & 4 \\ 7 & 14 & 20 & 27 \\ 5 & 10 & 16 & 19 \\ 3 & 5 & 6 & 13 \end{bmatrix}; b_i = \begin{bmatrix} 0 \\ 0 \\ -2 \\ 5 \end{bmatrix}$	6	$a_{ij} = \begin{bmatrix} 2 & 4 & 3 & 4 \\ 4 & -2 & 5 & 6 \\ 6 & -3 & 7 & 8 \\ 8 & -4 & 9 & 10 \end{bmatrix}; b_i = \begin{bmatrix} 5 \\ 7 \\ 9 \\ 11 \end{bmatrix}$
7	$a_{ij} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 1 & 2 \\ -1 & 2 & 4 & 1 \\ 2 & -1 & 1 & -1 \end{bmatrix}; b_i = \begin{bmatrix} 1 \\ 1 \\ 18 \\ -12 \end{bmatrix}$	8	$a_{ij} = \begin{bmatrix} 5 & -3 & 2 & 4 \\ 4 & -2 & 3 & 7 \\ 8 & -6 & -1 & -5 \\ 7 & -3 & 7 & 17 \end{bmatrix}; b_i = \begin{bmatrix} 7 \\ 10 \\ 4 \\ 17 \end{bmatrix}$
9	$a_{ij} = \begin{bmatrix} 1 & 2 & 15 & 11 \\ -2 & 4 & 3 & 61 \\ -3 & -8 & 11 & 12 \\ 15 & 7 & 8 & -4 \end{bmatrix}; b_i = \begin{bmatrix} 2 \\ 4 \\ 5 \\ 9 \end{bmatrix}$	10	$a_{ij} = \begin{bmatrix} 5 & 3 & -4 & -15 \\ 6 & 1 & 10 & 2 \\ 1 & 5 & 3 & 8 \\ -4 & 5 & -1 & 9 \end{bmatrix}; b_i = \begin{bmatrix} 6 \\ 9 \\ 12 \\ -9 \end{bmatrix}$
11	$a_{ij} = \begin{bmatrix} 7 & -8 & 9 & 11 \\ -5 & 6 & 5 & 3 \\ -4 & -7 & 3 & 6 \\ 2 & 1 & 5 & 9 \end{bmatrix}; b_i = \begin{bmatrix} 3 \\ -4 \\ -6 \\ 11 \end{bmatrix}$	12	$a_{ij} = \begin{bmatrix} 1 & 2 & 5 & -11 \\ 2 & -6 & -3 & 8 \\ 3 & 4 & 5 & 1 \\ 7 & 6 & 2 & 20 \end{bmatrix}; b_i = \begin{bmatrix} 7 \\ 11 \\ 15 \\ -9 \end{bmatrix}$
13	$a_{ij} = \begin{bmatrix} 5 & 3 & 3 & -4 \\ -6 & -4 & 2 & 17 \\ 12 & 26 & 1 & 5 \\ 5 & 8 & 9 & 7 \end{bmatrix}; b_i = \begin{bmatrix} 7 \\ -24 \\ 34 \\ -6 \end{bmatrix}$	14	$a_{ij} = \begin{bmatrix} 4 & 7 & 6 & -19 \\ 5 & 9 & 12 & 3 \\ -4 & -3 & 1 & -8 \\ 2 & 5 & 9 & 10 \end{bmatrix}; b_i = \begin{bmatrix} 10 \\ 12 \\ -14 \\ 22 \end{bmatrix}$

Продолжение таблицы 2.8.

№	Матрицы А и В	№	Матрицы А и В
15	$a_{ij} = \begin{bmatrix} 12 & 8 & 24 & -5 \\ 16 & 25 & 9 & 9 \\ 2 & 7 & 3 & 4 \\ 21 & -9 & -7 & 6 \end{bmatrix}; b_i = \begin{bmatrix} 5 \\ -17 \\ 22 \\ 3 \end{bmatrix}$	16	$a_{ij} = \begin{bmatrix} -3 & -7 & 16 & 8 \\ -4 & 6 & 12 & 13 \\ 2 & 3 & 7 & 5 \\ 18 & -14 & 1 & 2 \end{bmatrix}; b_i = \begin{bmatrix} 6 \\ -12 \\ 31 \\ 41 \end{bmatrix}$
17	$a_{ij} = \begin{bmatrix} 2 & 5 & 9 & 16 \\ 11 & 16 & 3 & 1 \\ 25 & 14 & -38 & 7 \\ -8 & 9 & -16 & -42 \end{bmatrix}; b_i = \begin{bmatrix} 1 \\ 1 \\ -3 \\ 6 \end{bmatrix}$	18	$a_{ij} = \begin{bmatrix} 2 & -1 & 3 & 2 \\ 3 & 3 & 3 & 2 \\ 3 & -1 & -1 & -2 \\ 3 & -1 & 3 & -1 \end{bmatrix}; b_i = \begin{bmatrix} 4 \\ 6 \\ 6 \\ 6 \end{bmatrix}$
19	$a_{ij} = \begin{bmatrix} 2 & 2 & -1 & 1 \\ 4 & 3 & -1 & 2 \\ 8 & 5 & -3 & 4 \\ 3 & 3 & -2 & 2 \end{bmatrix}; b_i = \begin{bmatrix} 4 \\ 6 \\ 12 \\ 6 \end{bmatrix}$	20	$a_{ij} = \begin{bmatrix} 2 & 3 & 11 & 5 \\ 1 & 1 & 5 & 2 \\ 2 & 1 & 3 & 2 \\ 1 & 1 & 3 & 4 \end{bmatrix}; b_i = \begin{bmatrix} 2 \\ 1 \\ -3 \\ -3 \end{bmatrix}$
21	$a_{ij} = \begin{bmatrix} 2 & 5 & 4 & 1 \\ 1 & 3 & 2 & 1 \\ 2 & 10 & 9 & 9 \\ 3 & 8 & 2 & 2 \end{bmatrix}; b_i = \begin{bmatrix} 20 \\ 11 \\ 40 \\ 37 \end{bmatrix}$	22	$a_{ij} = \begin{bmatrix} 3 & 4 & 1 & 2 \\ 3 & 5 & 3 & 5 \\ 6 & 8 & 1 & 5 \\ 3 & 5 & 3 & 7 \end{bmatrix}; b_i = \begin{bmatrix} -3 \\ -6 \\ -8 \\ -8 \end{bmatrix}$
23	$a_{ij} = \begin{bmatrix} 3.2 & 5.4 & 4.2 & 2.2 \\ 2.1 & 3.2 & 3.1 & 1.1 \\ 1.2 & 0.4 & -0.8 & -0.8 \\ 4.7 & 10.4 & 9.7 & 9.7 \end{bmatrix}; b_i = \begin{bmatrix} 2.6 \\ 4.8 \\ 3.6 \\ -8.4 \end{bmatrix}$	24	$a_{ij} = \begin{bmatrix} 3 & -2 & -5 & 1 \\ 2 & -3 & 1 & 5 \\ 1 & 2 & 0 & -4 \\ 1 & -1 & -4 & 9 \end{bmatrix}; b_i = \begin{bmatrix} 3 \\ -3 \\ -3 \\ 22 \end{bmatrix}$
25	$a_{ij} = \begin{bmatrix} 1 & 1 & -6 & -4 \\ 3 & -1 & -6 & -4 \\ 2 & 3 & 9 & 2 \\ 3 & 2 & 3 & 8 \end{bmatrix}; b_i = \begin{bmatrix} 6 \\ 2 \\ 6 \\ -7 \end{bmatrix}$	26	$a_{ij} = \begin{bmatrix} 2 & -1 & 1 & 2 \\ 6 & -4 & 2 & 4 \\ 6 & -3 & 4 & 8 \\ 4 & -2 & 1 & 1 \end{bmatrix}; b_i = \begin{bmatrix} 2 \\ 3 \\ 9 \\ 1 \end{bmatrix}$
27	$a_{ij} = \begin{bmatrix} 12 & 14 & -15 & 24 \\ 16 & 18 & -22 & 29 \\ 18 & 20 & -21 & 32 \\ 10 & 12 & -16 & 20 \end{bmatrix}; b_i = \begin{bmatrix} 5 \\ 8 \\ 9 \\ 4 \end{bmatrix}$	28	$a_{ij} = \begin{bmatrix} 24 & 14 & 30 & 40 \\ 36 & 25 & 45 & 61 \\ 48 & 28 & 60 & 82 \\ 60 & 35 & 75 & 99 \end{bmatrix}; b_i = \begin{bmatrix} 28 \\ 43 \\ 58 \\ 69 \end{bmatrix}$
29	$a_{ij} = \begin{bmatrix} 5 & -3 & 2 & 4 \\ 4 & -2 & 3 & 7 \\ 8 & -6 & -1 & -5 \\ 7 & -3 & 7 & 17 \end{bmatrix}; b_i = \begin{bmatrix} 3 \\ 1 \\ 9 \\ 15 \end{bmatrix}$	30	$a_{ij} = \begin{bmatrix} 2 & -1 & 3 & 4 \\ 4 & -8 & 5 & 6 \\ 6 & -5 & 7 & 8 \\ 12 & -4 & -9 & 10 \end{bmatrix}; b_i = \begin{bmatrix} 5 \\ 7 \\ 9 \\ 13 \end{bmatrix}$

2.1.8 Задание на лабораторную работу №17

**Тема:** Аппроксимация функций в среде программного пакета Matlab.

**Цель работы:** изучить существующие методы аппроксимации линейных и нелинейных зависимостей, приобрести навыки нахождения графиков и уравнений зависимостей по экспериментальным данным.

### ***Теоретические сведения***

При проведении научно-технических расчетов часто используются эмпирические зависимости, причем число точек этих зависимостей ограничено. Неизбежно возникает задача восполнения данных: имеется некоторое количество точек, через которые следует провести кривую. Это не что иное, как классическая задача интерполяции.

Аппроксимация (от лат. *approximo* – приближаюсь) – замена одних математических объектов другими, каким-то образом близким к исходным. Аппроксимация разрешает исследовать числовые характеристики и качественные свойства объектов, сводя задачу к изучению более простых или более удобных объектов.

Если величина  $y$  выступает в роли значения функции аргумента  $x$ , то значит любому значению  $x$  из области определения поставлено в соответствие значение  $y$ . На практике явная связь между  $y$  и  $x$  оказывается неизвестной и в лучшем случае может быть задана таблично. То есть дискретному множеству значений аргумента  $x_i$  поставлено в соответствие множество значений функции  $y_i$ . Получение значений промежуточных точек сопряжено со сложными расчетами или проведением дорогостоящих экспериментов. Таким образом, с учетом экономии времени и средств, возникает вопрос необходимости использования имеющихся табличных данных для приближенного вычисления неизвестного параметра  $y$  при любом значении (из некоторой области) определяющего параметра  $x$  при неизвестности связи  $y = f(x)$ .

Этой цели подчинена задача о приближении (аппроксимации) функции: данную функцию  $f(x)$  нужно приблизительно заменить (аппроксимировать) некоторой функцией  $F(x)$  так, чтобы отклонения (в некотором смысле)  $F(x)$  от  $f(x)$  в заданной области было наименьшим. Функция  $F(x)$  при этом называется аппроксимирующей.

Как классы аппроксимирующих функций могут выступать полиномиальные, тригонометрические, экспоненциальные и др. Аппроксимация функции на практике иногда осуществляется многочленом. Если приближения строятся на заданном дискретном множестве точек, то аппроксимация называется точечной (среднеквадратичное приближение, интерполяция). При построении приближения на непрерывном множестве точек аппроксимация называется непрерывной (интегральной).

Интерполяция – частный случай более общей задачи аппроксимации (приближенного представления), возникающей при замене кривой, описываемой функцией сложной природы, другой кривой, в некотором смысле близкой заданной, имеющей более простые

уравнения [17]. В этом случае аппроксимирующая функция проходит через заданные узловые точки.

### Ход работы:

Запустить пакет Matlab, создать файл-сценарий m-file (рисунок 2.10).

```
>> x = [0.75, 1.50, 2.25, 3.00, 3.75];  
>> y = [2.50, 1.20, 1.12, 2.25, 4.28];  
>> plot(x, y, 'o');
```

Рисунок 2.10 – Фрагмент файла-сценария

Для построения аппроксимирующей зависимости указать следующие пункты программы:

- ввести значения  $x$  и  $y$ ;
- построить точечный график с помощью оператора `plot`;
- задать свойства графика (рисунок 2.11): показать эмпирическую формулу, выбрать несколько вариантов аппроксимации (линейная, квадратная, сплайн и т.д.).
- вывести на экран величину и график погрешности аппроксимации (рисунок 2.11);
- проанализировать полученные результаты.

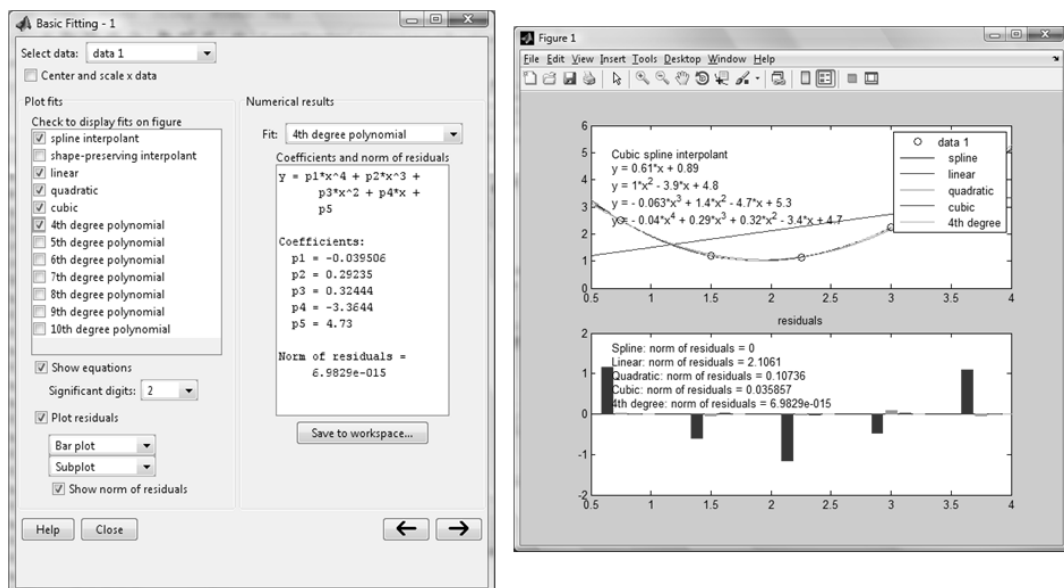


Рисунок 2.11 – Настройка свойств графика и результаты аппроксимации

### *Индивидуальное задание*

Получить эмпирические формулы и оценить их погрешность для функции  $y = f(x)$ , заданной таблично согласно варианту (таблица 2.9). Оценить погрешность эмпирических формул, выбрав наиболее оптимальный вариант математической зависимости.

*Таблица 2.9 – Варианты заданий*

№	Значения X и Y										
1	X	1,01	1,51	2,02	2,51	3,01	3,49	3,98	4,48	4,99	5,49
	Y	5,02	5,92	7,14	8,32	9,02	9,58	11,06	11,96	12,78	13,98
2	X	1,02	1,59	2,12	2,61	3,05	3,56	4	4,5	5,03	5,56
	Y	3,04	4,03	4,4	5,02	5,43	5,34	6,07	5,71	6,63	6,03
3	X	1,1	1,55	2,09	2,52	3,07	3,57	4,05	4,56	5,06	5,53
	Y	3,12	4,69	8,86	16,46	31,08	53,13	81,2	122,6	173,5	234,3
4	X	1,08	1,53	2,05	2,58	3,02	3,58	4,06	4,56	5,01	5,51
	Y	1,04	4,09	6,39	6,15	6,18	5,42	6,53	8,04	12,31	19,3
5	X	1,1	1,33	1,58	1,81	2,09	2,32	2,59	2,85	3,14	3,43
	Y	3,46	5,52	9,24	12,7	19,13	24,92	34,35	43,86	57,61	71,99
6	X	1,21	1,3	1,56	1,9	2,19	2,35	2,61	3,01	3,16	3,53
	Y	75,4	95,8	216,5	599,7	1425	2310	5030	16698	26190	79471
7	X	2,01	2,51	3,02	3,51	4,01	4,49	4,98	5,48	5,99	6,49
	Y	7,02	7,92	9,14	10,32	11,02	11,58	13,06	13,96	14,78	15,98
8	X	2,02	2,59	3,12	3,61	4,05	4,56	5	5,5	6,03	6,56
	Y	4,41	5	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36
9	X	2,1	2,55	3,09	3,52	4,07	4,57	5,05	5,56	6,06	6,53
	Y	9,18	17	31,88	50,53	82,49	123,5	172,5	239,3	318	407,7
10	X	2,08	2,53	3,05	3,58	4,02	4,58	5,06	5,56	6,01	6,51
	Y	6,34	6,17	6,25	5,42	6,24	8,16	13,08	20,02	30,46	45,73
11	X	2,1	2,33	2,58	2,81	3,09	3,32	3,59	3,85	4,14	4,43
	Y	19,37	25,42	33,9	42,11	54,78	66,08	82,38	98,92	121	144,1
12	X	0,1	0,33	0,58	0,81	1,09	1,32	1,59	1,85	2,14	2,43
	Y	2,7	2,38	12,39	24,72	50,62	108,9	235,8	512,5	1228	2931
13	X	3,01	3,51	4,02	4,51	5,01	5,49	5,98	6,48	6,99	7,49
	Y	9,02	9,92	11,14	12,32	13,02	13,58	15,06	15,96	16,78	17,98
14	X	3,02	3,59	4,12	4,61	5,05	5,56	6	6,5	7,03	7,56
	Y	5,21	5,66	5,73	6,16	6,44	6,23	6,88	6,44	7,3	6,65
15	X	3,1	3,55	4,09	4,52	5,07	5,57	6,05	6,56	7,06	7,53
	Y	32,44	51,92	83,99	118,8	174,7	240,6	316,4	414,7	527,2	651,5

Продолжение таблицы 2.9.

№	Значения X и Y										
16	X	3,08	3,53	4,05	4,58	5,02	5,58	6,06	6,56	7,01	7,51
	Y	6,12	5,42	6,41	8,16	12,42	20,39	31,99	47,36	66,67	93,22
17	X	3,1	3,33	3,58	3,81	4,09	4,32	4,59	4,85	5,14	5,43
	Y	55,26	66,82	81,67	96,07	116,7	134,8	159,6	184,6	216,6	250,1
18	X	1,17	1,53	1,56	1,85	2,09	2,15	2,31	3,01	3,16	3,23
	Y	66,9	194	216,5	516,5	1055	1269	2045	16698	26190	32310
19	X	4,01	4,52	4,03	4,53	5,21	5,49	5,98	6,48	6,79	7,39
	Y	11	11,94	11,16	12,36	13,42	13,58	15,06	15,96	16,38	17,78
20	X	2,34	2,61	3,24	3,87	4,15	4,62	5,02	5,53	6,23	7,56
	Y	4,7	5,02	5,25	5,81	6,05	5,86	6,53	6,12	7,06	6,65
21	X	3,12	3,55	4,11	4,52	5,17	5,47	6,25	6,56	7,06	7,43
	Y	33,17	51,92	85,41	118,8	186,7	226,5	352,6	414,7	527,2	623,6
22	X	3,18	3,43	4,25	4,58	5,12	5,38	6,26	6,56	7,21	7,58
	Y	6,03	5,45	7	8,16	13,61	16,9	37,79	47,36	76,61	97,49
23	X	1,65	1,98	2,08	2,18	2,35	2,48	2,65	3,05	3,42	3,53
	Y	10,15	16,34	19,08	21,12	26,21	29,9	36,52	52,69	72,42	77,84
24	X	5,5	6,1	7,8	8,2	10,4	15,34	17,2	18,7	23,6	40,2
	Y	0,1	0,33	0,58	0,81	1,09	1,32	1,59	1,85	2,14	2,43
25	X	2,01	2,51	3,02	3,51	4,01	4,49	4,98	5,48	5,99	6,49
	Y	9,18	17	31,88	50,53	82,49	123,5	172,5	239,3	318	407,7
26	X	2,02	2,59	3,12	3,61	4,05	4,56	5	5,5	6,03	6,56
	Y	4,7	5,02	5,25	5,81	6,05	5,86	6,53	6,12	7,06	6,65
27	X	2,1	2,55	3,09	3,52	4,07	4,57	5,05	5,56	6,06	6,53
	Y	6,03	5,45	7	8,16	13,61	16,9	37,79	47,36	76,61	97,49
28	X	2,08	2,53	3,05	3,58	4,02	4,58	5,06	5,56	6,01	6,51
	Y	4,41	5	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36
29	X	2,1	2,33	2,58	2,81	3,09	3,32	3,59	3,85	4,14	4,43
	Y	7,02	7,92	9,14	10,32	11,02	11,58	13,06	13,96	14,78	15,98
30	X	2,08	2,53	3,05	3,58	4,02	4,58	5,06	5,56	6,01	6,51
	Y	4,41	5	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36

## 2.1.9 Задание на лабораторную работу № 18

**Тема:** Основы моделирования электромеханических систем в среде Simulink пакета Matlab.

**Цель работы:** изучить компоненты – динамические звенья для моделирования в среде Simulink пакета Matlab, ознакомиться с особенностями создания структурных схем электромеханических систем.

### *Теоретические сведения*

Использование Matlab в большей степени обусловлено наличием пакета Simulink - интерактивного инструмента для моделирования, расчетов, имитации и анализа динамических систем. Он дает возможность строить графические блок-диаграммы, имитировать динамические системы, исследовать работоспособность систем и совершенствовать проекты. Simulink полностью интегрирован с Matlab, обеспечивая немедленным доступом к широкому спектру инструментов анализа и проектирования. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты, выбирая метод решения дифференциальных уравнений, а также способ изменения модельного времени. Эти преимущества делают Simulink наиболее популярным инструментом для проектирования систем управления и коммуникации, цифровой обработки и других приложений моделирования [14].

В Simulink блоки, согласно их назначению, объединены в библиотеки блоков и после запуска пакет предоставляет библиотечное окно просмотра древовидной формы установленных библиотек, где путем выбора необходимых элементов осуществляется набор структурной схемы (рисунок 2.2).

Используя блоки библиотеки пакета Simulink Sources (Step – генерация ступенчатого сигнала), Sinks (Scope – реализация вывода результатов расчета в виде графиков), Continuous(Integrator – интегратор, Transfer Fcn – реализация линейной передаточной функции), Math(Sum – генерация суммы или разности входов, Gain – поэлементный или матричный коэффициент усиления, Sign – указание знака входа), вид которых представлен на рисунке 2.11, производится составление структурной схемы электромеханических систем.





Инерционное звено представляется блоком «Transfer-Fcn». При этом следует задавать параметры блоков – двойной щелчок манипулятора «мышь» активизирует окно параметров (рисунок 2.13).

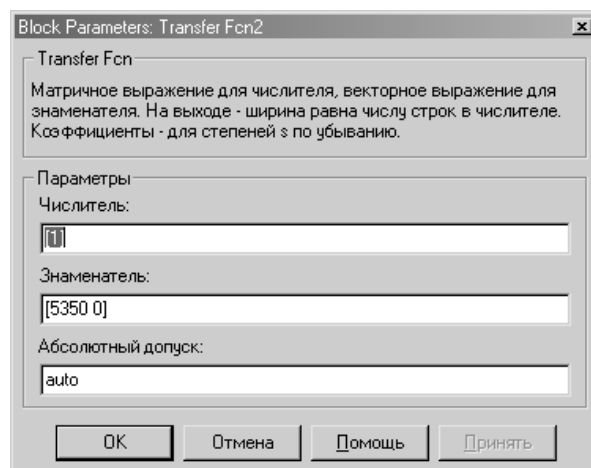


Рисунок 2.13 – Окно задания параметров блока «Transfer-Fcn»

Коэффициенты пропорциональности на схеме изображаются блоком «Gain» с параметрами (рисунок 2.14).

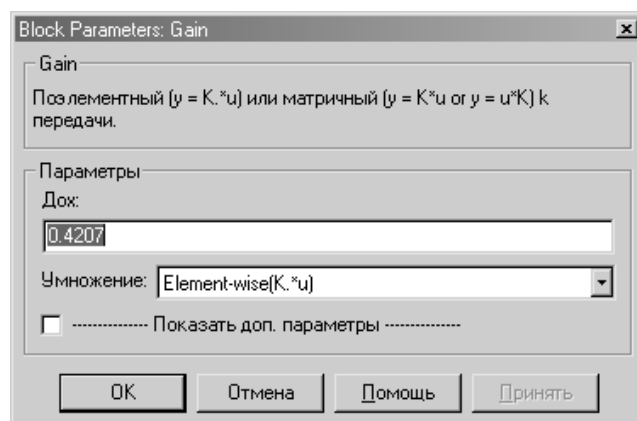


Рисунок 2.14 – Окно задания параметров блока «Gain»

Вывод графиков колебаний координат осуществляется с помощью блока реализации вывода результатов в виде графиков – «Scope».

Прежде чем запускать модель, стоит произвести установку общих параметров моделирования, что осуществляется выполнением команды Simulations Parameters в меню Simulations окна Simulink и появлением окна установки параметров моделирования (рисунок 2.15).

Это окно имеет ряд вкладок с параметрами для моделирования. Вкладка Solver (Решатель) позволяет установить параметры решающего устройства системы моделирования. К числу важнейших параметров относятся время моделирования – Simulation time. Она задаётся

начальным временем Start time («0» по умолчанию) и конечным временем Stop time в секундах.

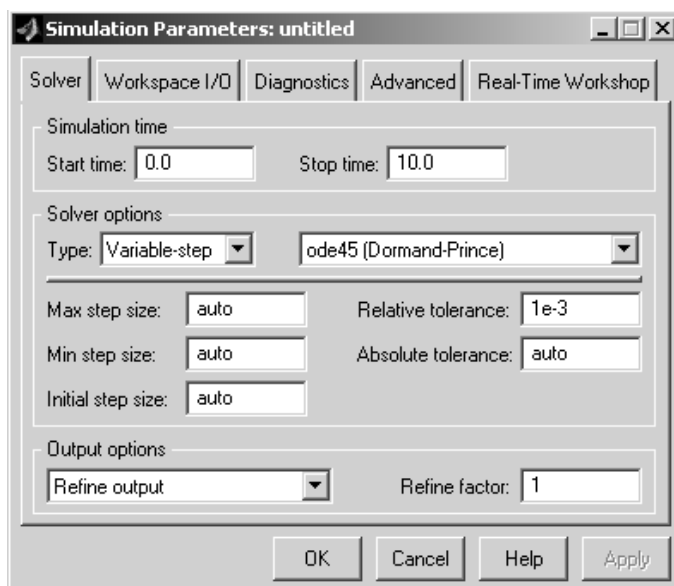


Рисунок 2.15 – Окно установки параметров моделирования

Первостепенное значение имеют две опции решателя в поле Solver Options: тип решения и метод решения. Возможны два типа решения: variable-step solvers (решение с переменным шагом) и fixed-step solvers (решение с фиксированным шагом). Для решения дифференциальных уравнений можно выбирать следующие методы: discrete (дискретный), ode45, ode23 (три варианта, включая метод Розенброка), rk45 (метод Дорманда-Принса), ode113 (метод Адамса) и ode15s. Методы, в названии которых есть слово stiff, служат для решения жестких систем дифференциальных уравнений.

Следующие три параметра – значения опции auto – обычно задаются автоматически, но в особых случаях их можно ввести явно:

а) Max step size – максимальный шаг интегрирования системы дифференциальных уравнений;

б) Min step size – минимальный шаг интегрирования;

в) Initial step size – начальный шаг интегрирования.

Последним параметром является точность интегрирования: relative tolerance – относительная погрешность, absolute tolerance – абсолютная.

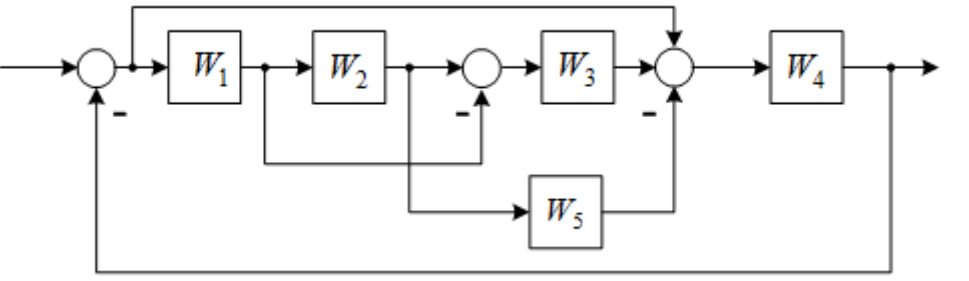
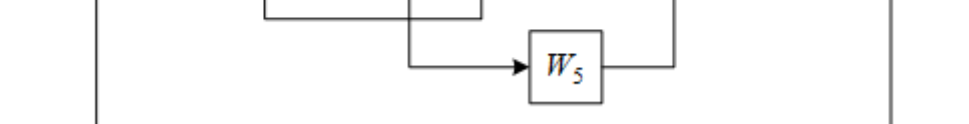
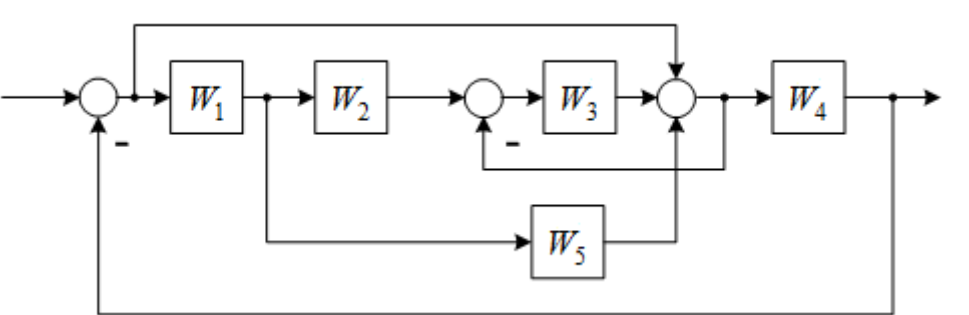

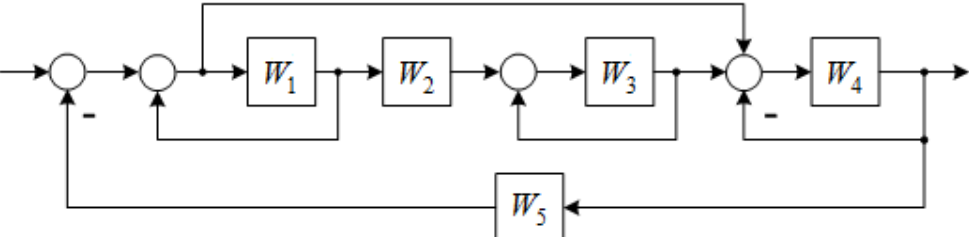
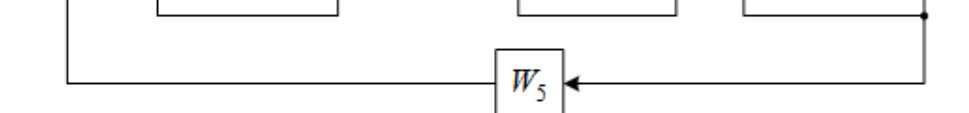
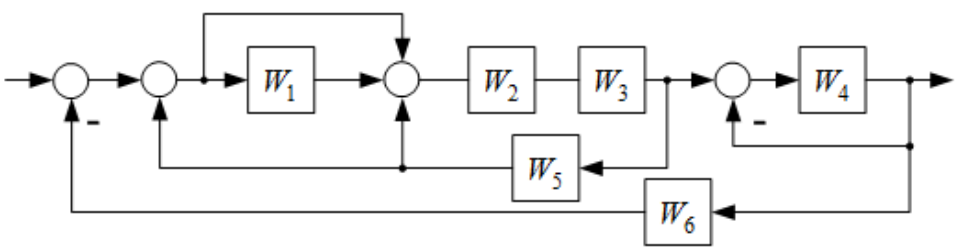
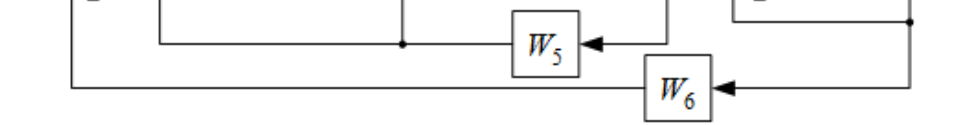
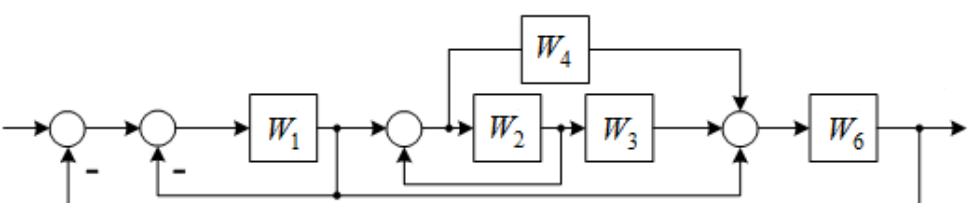
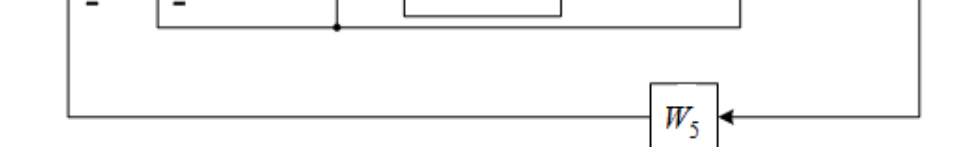
По умолчанию они заданы соответственно равными  $10^{-3}$  и  $10^{-6}$ .

### ***Индивидуальное задание***

1 Согласно варианту индивидуального задания (таблица 2.10) составить в среде Simulink математическую модель в виде структурной схемы [16], параметры передаточных функций которой указаны в таблице 2.11.

2 Выполнить моделирование процессов для ступенчатого и синусоидального входных сигналов и получить осциллограмму переходного процесса, включающую кривую входного и выходного сигналов системы.

Таблица 2.10 – Варианты индивидуальных заданий

№	Структурная схема
1	
23	
2	
24	
3	
25	
4	
26	
5	
27	

Продолжение таблицы 2.10.

№	Структурная схема
6	
28	
7	
29	
8	
30	
9	

Продолжение таблицы 2.10.

№	Структурная схема
10	
11	
12	
13	

Продолжение таблицы 2.10.

№	Структурная схема
14	
15	
16	
17	

Продолжение таблицы 2.10.

№	Структурная схема
18	
19	
20	
21	
22	



Таблица 2.11- Варианты индивидуальных заданий

Вариант	Передаточные функции		Параметры звеньев
четный	1	$W(p) = \frac{k}{Tp + 1}$	$k = 10 c^{-1}, T = 1 c$
	2	$W(p) = \frac{k}{(Tp + 1)^2}$	$k = 100 c^{-2}, T = 0.1 c$
	3	$W(p) = \frac{Tp}{Tp + 1}$	$T = 0.1 c$
	4	$W(p) = \frac{Tp - 1}{Tp + 1}$	$T = 0.2 c$
	5	$W(p) = \frac{Tp + 1}{Tp - 1}$	$T = 0.05 c$
	6	$W(p) = k(Tp + 1)$	$k = 10 c^{-1}, T = 0.01 c$
	7	$W(p) = \frac{Tp + 1}{Tp}$	$T = 0.01 c$
	8	$W(p) = \frac{k}{Tp(Tp - 1)}$	$k = 50 c^{-1}, T = 1 c$
нечетный	1	$W(p) = \frac{k}{Tp(Tp + 1)}$	$k = 10 c^{-1}, T = 1 c$
	2	$W(p) = \frac{k}{T^2 p^2 + 2\xi Tp + 1}$	$T = 1 c, \xi = 0.5,$ $k = 10 c^{-2}$
	3	$W(p) = \frac{Tp + 1}{Tp}$	$T = 0.01 c$
	4	$W(p) = \frac{k}{T^2 p^2 + 1}$	$k = 10 c^{-2}, T = 0.01 c$
	5	$W(p) = \frac{Tp - 1}{T^2 p^2 + 1}$	$T = 0.01 c$
	6	$W(p) = \frac{T^2 p^2 - 1}{T^2 p^2 + 1}$	$T = 1 c$
	7	$W(p) = \frac{k(Tp + 1)}{p^2(Tp - 1)}$	$k = 100 c^{-2},$ $T = 0.001 c$
	8	$W(p) = \frac{Tp + 1}{Tp - 1}$	$T = 0.05 c$

## 2.1.10 Задание на лабораторную работу №19

**Тема:** Численное решение оптимизационных задач.

**Цель работы:** изучить особенности оптимизации инженерных задач, приобрести навыки решения оптимизационных задач в среде программного пакета Matlab.

### ***Теоретические сведения***

Под оптимизацией понимают процесс выбора наилучшего из всех возможных вариантов. С точки зрения инженерных расчетов методы оптимизации позволяют выбрать наилучший вариант конструкции, наилучшее распределение ресурсов, минимальный урон природной среде и т.д. В процессе решения задач оптимизации необходимо найти оптимальные значения некоторых параметров – проектных параметров. Выбор оптимального решения проводится с помощью некоторой функции, называемой целевой функцией, которую можно записать в виде:

$$u = f(x_1, x_2, \dots, x_n),$$

где  $x_1, x_2, \dots, x_n$  – проектные параметры.

Можно выделить 2 типа задач оптимизации – безусловные и условные [15]. Безусловная задача оптимизации состоит в отыскании максимума или минимума целевой функции от  $n$  действительных переменных и определении соответствующих значений аргументов на некотором множестве  $G$   $n$ -мерного пространства. Обычно рассматриваются задачи минимизации; к ним легко сводятся и задачи на поиск максимума путем замены знака целевой функции на противоположный. Условные задачи оптимизации – это такие, при формулировке которых задаются некоторые условия (ограничения) на множестве  $G$ . Для решения оптимизационных задач в пакете Matlab используются специальные функции `fminbnd`, `fminsearch` в основе которых лежат методы золотого сечения или параболической интерполяции. Данные функции применяются для поиска минимума функций одной или нескольких переменных.

Для минимизации функций нескольких переменных применяется симплекс-метод Нелдера-Мида, сводящийся к построению симплекса в  $n$ -мерном пространстве, заданного  $n+1$  вершиной. В двухмерном пространстве симплекс является треугольником, а в трехмерном – пирамидой. На каждом шаге итераций выбирается новая точка решения внутри или вблизи симплекса и сравнивается с одной из его вершин. Ближайшая к этой точке вершина симплекса заменяется этой точкой. Таким образом, симплекс перестраивается и позволяет найти новое, более точное положение точки решения. Алгоритм поиска повторяется, пока

размер симплекса по всем переменным не станет меньше заданной погрешности решения.

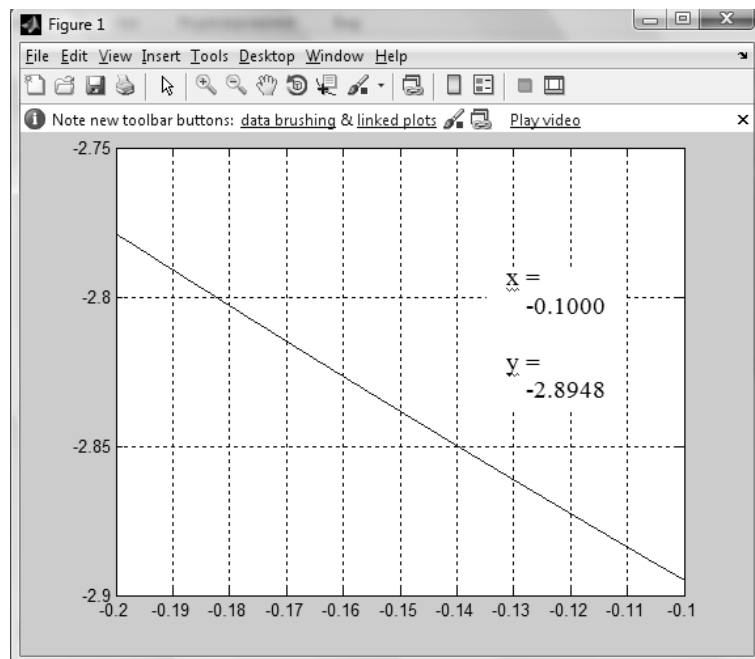
### *Ход работы*

Для поиска минимума функции  $f(x) = x^2 - e^x - 2$  необходимо запустить пакет Matlab, где в окне главного меню выберем Файл->Новый->М-файл – файл для реализации кода математического задания. После появления окна листинга, построчно введем ограничения параметра  $x$  – области определения согласно заданию, выражение для задания функции  $y$ , зависящей от переменной  $x$ . Согласно командам пакета выполним построение графика функции одной переменной – `plot(x,y)` с нанесением координатной сетки командой `grid on`, а после построения графика можем определить минимальное значение функции визуально или с максимальной точностью с помощью встроенной команды поиска экстремума – `fminbnd`. Листинг приведен на рисунке 2.16.

```
x=-0.2:0.0001:-0.1;  
y=x.^2-exp(x)-2;  
plot(x,y);  
grid on;  
[x,y]=fminbnd(' (x.^2-exp(x)-2) ',-0.2,-0.1)
```

*Рисунок 2.16 – Листинг программного кода*

Результат работы представлен ниже на рисунке 2.17.



*Рисунок 2.17 – Результат поиска минимума функции*

Для поиска минимума функции нескольких переменных  $f(x, y) = \ln(1+x^2 + y^2)^2 + (x - y - 1)^2$  воспользуемся соответствующей функцией `fmin search` (листинг приведен на рисунке 2.18) при этом графический ответ получаем в виде ограничивающих трехмерное пространство кривых (рисунок 2.19).

```
[X,Y]=meshgrid([-2:2.0,2.0:3]);
Z=(log(1+X.^2+Y.^2)).^2+(X-Y-1).^2;
plot3(X,Y,Z)
grid on
[xmin,minf]=fminsearch(@Fxy_9,[2.0;2.0])

function f=Fxy_9(x)
f=(log(1+x(1).^2+x(2).^2)).^2+(x(1)-x(2)-1).^2;
```

Рисунок 2.18 – Программный код

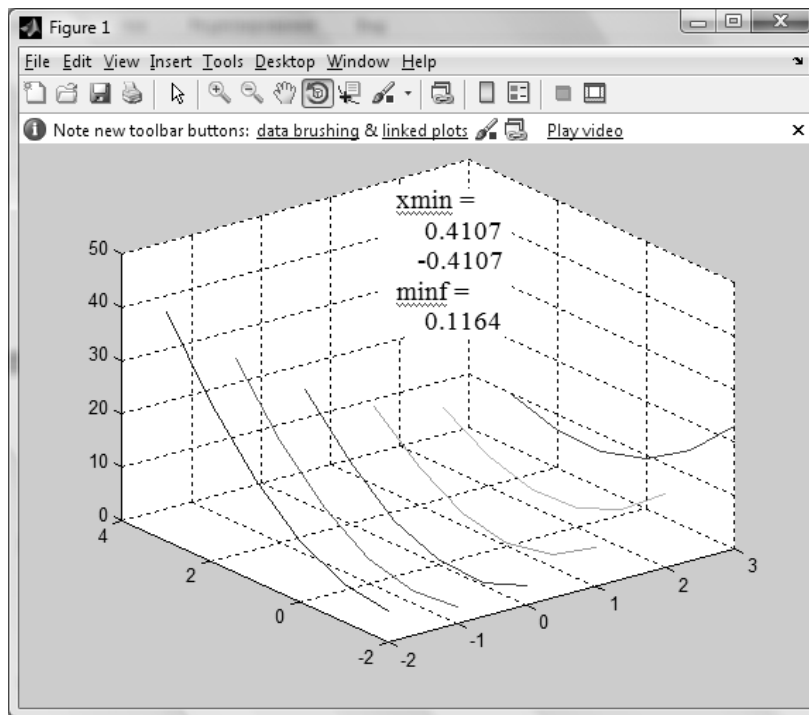


Рисунок 2.19 – Результат поиска минимума

### Индивидуальное задание

1 Согласно варианту индивидуального задания (таблица 2.12) найти минимальное значение заданной функции одной переменной и вывести на печать координату и минимальное значение функции на заданном отрезке.

2 Найти и вывести на печать координаты и значение минимума функции двух переменных с учетом координат начальной точки поиска согласно варианту задания (таблица 2.13).

Таблица 2.12 – Варианты индивидуальных заданий

№	Функция $f(x)$	Отрезок [a; b]
1	$f(x) = x / \ln x$	[1,2; 4]
2	$f(x) = x - 2 \sin x$	[0; $\pi/2$ ]
3	$f(x) = (x^2 - 1)/(x^2 + 1)$	[-2; 2]
4	$f(x) = 3x^4 - 4x^3 + 12x^2 + 1$	[-2; 2]
5	$f(x) = x - 2 \ln x$	[1; 3]
6	$f(x) = e^x \cos x$	[ $\pi$ ; $3\pi/2$ ]
7	$f(x) = (1 - x + x^2)/(1 + x - x^2)$	[0; 1]
8	$f(x) = -\sqrt{2x - x^2}$	[0; 2]
9	$f(x) = (x - 2)^5(2x + 1)^4$	[-0,5; 1,5]
10	$f(x) = x^x$	[0,1; 1,0]
11	$f(x) = e^{-1/x^2}$	[-0,5; 0,5]
12	$f(x) = x\sqrt{1 - x^2}$	[-1,0; 0]
13	$f(x) = ((e^x + e^{-x})/2)^3 + 1$	[-0,5; 0,5]
14	$f(x) = -x/(x^3 + 2)$	[0,5; 1,5]
15	$f(x) = x^2 / \sqrt[3]{x^3 - 4}$	[1,6; 2,2]
16	$f(x) = \sqrt{ x^2 - 3 } / x$	[1; 2]
17	$f(x) = x^2 \sqrt{ x^2 - 1 }$	[1,1; 1,6]
18	$f(x) = 1/(\sin x + \cos x)$	[0; $\pi/3$ ]
19	$f(x) = x/2 + \arctg x$	[0,5; 1,2]
20	$f(x) = x e^{-x^2/2}$	[-1,5; -0,5]
21	$f(x) = e^{-1/x^2} / x$	[-2,0; -1,0]
22	$f(x) = - x ^3 e^{-x^2/2}$	[-2,0; -1,0]
23	$f(x) = x^2 \ln x$	[0,1; 1,0]
24	$f(x) = x \ln^2  x $	[-0,05; -0,2]
25	$f(x) = x \arctg x$	[-0,5; 0,5]
26	$f(x) = \sin x + \cos x$	[ $\pi$ ; $3\pi/2$ ]
27	$f(x) = -\ln x / x^2$	[1,0; 2,0]
28	$f(x) = x^2 \ln^2 x$	[0,1; 0,5]
29	$f(x) = \sin x / x$	[ $\pi$ ; $2\pi$ ]
30	$f(x) = -x^{1/x}$	[2,0; 3,0]

Таблица 2.13- Варианты индивидуальных заданий

№	Функция $f(x,y)$	Координаты начальной точки $M_0(x_0; y_0)$
1	$(2x^2 - y - 3)^2 + x^2 + 2x + 2$	(1; 1)
2	$(xy + 2)^2 + y^2 + 2y + 4$	(2; 2)
3	$(x^2y^2 - y + 2)^2 + x^2 + 1$	(2; 3)
4	$(3x^2 + 2y^2 - 1)^2 + (xy - 3)^2$	(-2; 2)
5	$(2x^2 - 7y^2 - 2)^2 + (x^2 + y^2 - 20)^2 + 3$	(2; 4)
6	$(x^2 + y^2 - 2x - 3)^2 + (x^2 + y^2 - 2y - 3)^2$	(-1; 1)
7	$(x^2 - 6x + y^2 + 8)^2 + x^2y^2 + 1$	(3; 3)
8	$(x^2 - y = 2)^2 + (x - y + 3)^2$	(2; 2)
9	$\ln(1 + x^2 + y^2)^2 + (x - y - 1)^2$	(2; 2)
10	$(x^2 + y^2 - 1)^2 + (x^2 - 6x + y^2 + 8)^2$	(1; 1)
11	$x^3 + y^3 - 3xy$	(2,0; 2,0)
12	$x^2 + xy + y^2 - 3x - 6y$	(0,5; 0,5)
13	$-xy^2(1 - x - y)$	(0,5; 3,5)
14	$3x^2 - x^3 + 3y^2 + 4y$	(0; 0)
15	$xy + 50/x + 20/y$	(0,1; -1,0)
16	$x^2 + y^2 - 2 \ln x - 18 \ln y$	(4; 1)
17	$x^3 + 3xy^2 - 15x - 12y$	(0,5; 2,5)
18	$2x^3 - xy^2 + 5x^2 + y^2$	(1,5; 0,5)
19	$(2x^2 + y^2)e^{-(x^2+y^2)}$	(0,5; 0,5)
20	$-2 + \sqrt[3]{x^2 + y^2}$	(0,3; 0,3)
21	$(x-1)^2 + (y-2)^2 - 1$	(0,25; 0,25)
22	$\sin(x^2 + y^2 - 0.5)$	(0,5; 0,5)
23	$x^2 - xy + y^2 + 2x - 2y + 1$	(-1,5; 0,5)
24	$xy(x + y - 4)$	(1,0; 1,0)
25	$x^3y^2(x + y - 5)$	(2,0; 1,5)
26	$x^2 + xy + y^2 + 1/x + 1/y$	(0,2; 0,3)
27	$-(\sin x + \sin y + \sin(x + y))$	( $\pi/4$ ; $\pi/4$ )
28	$-\sin x \sin y \sin(x + y)$	( $\pi/4$ ; $\pi/4$ )
29	$x^3y^3 - 9xy + 1$	(2,5; 2,5)
30	$x^4 + y^4 - 2x^2 + 4xy - 2y^2 + 1$	(1,0; -1,0)

### 2.1.11 Задание на лабораторную работу № 20

**Тема:** Численное решение обыкновенных дифференциальных уравнений.

**Цель работы:** изучить методы поиска решения дифференциальных уравнений, приобрести навыки решения обыкновенных дифференциальных уравнений в среде программного пакета Matlab.

#### **Теоретические сведения**

Множество задач науки и техники исследовательского характера при их математическом моделировании сводятся к дифференциальным уравнениям, решение которых является одной из важнейших математических задач. В вычислительной математике изучаются численные методы решения уравнений, которые весьма эффективны при использовании ЭВМ. Среди множества численных методов решения дифференциальных уравнений явные одношаговые методы считаются наиболее простыми. К этому классу относятся различные модификации метода Рунге-Кутты [15].

Например, требуется найти функцию  $y=y(x)$ , удовлетворяющую уравнению:

$$dx/dy = F(x, y)$$

и принимающую при  $x=x_0$  заданное значение  $y_0$ :

$$y(x_0) = y_0.$$

При этом решение необходимо получить в интервале  $x_0 \leq x \leq x_k$ . Из теории дифференциальных уравнений известно, что решение  $y(x)$  задачи Коши существует, единственно и является гладкой функцией, если правая часть  $F(x, y)$  удовлетворяет некоторым условиям гладкости. Численное решение задачи Коши методом Рунге-Кутты 4-го порядка заключается в том, что на заданном интервале  $[x_0, x_k]$  выбираются узловые точки. Значение решения в нулевой точке известно  $y(x_0) = y_0$ . В следующей точке  $y(x_1)$  определяется по формуле:

$$y_1 = y_0 + (k_0 + 2k_1 + 2k_2 + k_3)/6,$$

где  $k_0 = hF(x_0, y_0)$ ;

$k_1 = hF(x_0 + h/2, y_0 + k_0/2)$ ;

$k_2 = hF(x_0 + h/2, y_0 + k_1/2)$ ;

$k_3 = hF(x_0 + h, y_0 + k_2)$ ;

$h$  – шаг сетки.

То есть данный вариант метода Рунге-Кутты требует на каждом шаге четырехкратного вычисления правой части уравнения. Этот алгоритм реализован в Matlab посредством функции ode45, однако эта функция не единственная, пакет располагает обширным набором аналогичных программ, позволяющих решать обыкновенные дифференциальные уравнения [17].

Например, необходимо найти решение задачи Коши:

$$dy/dx = 2(x^2 + y^2) \text{ на } [0, 1] \text{ при } y(0) = 1.$$

Точное решение имеет вид:

$$y(x) = 1,5e^{2x} - x^2 - x - 0,5.$$

Выполним решение задачи средствами функции ode45, создав файл-функцию F.m и вызвав ее в файле-сценарии (рисунок 2.20). В главном файле-сценарии обеспечивается вызов файл-функции, указывается интервал, на котором необходимо получить решение и начальное значение решения, далее осуществляется построение графика численного решения задачи Коши (рисунок 2.21) и вывод таблицы решения (рисунок 2.22).

```
function dydx = F(x, y)           >>[X Y] = ode45 ( @ F , [0 1] , [1] ) ;
dydx = zeros(1,1);              >> plot (X,Y) ;
dydx(1) = 2*(x^2+y(1));         >> hold on; gtext ( ' y(x) ' )
                                >> [X Y]
```

Рисунок 2.20 – Поиск решения задачи Коши

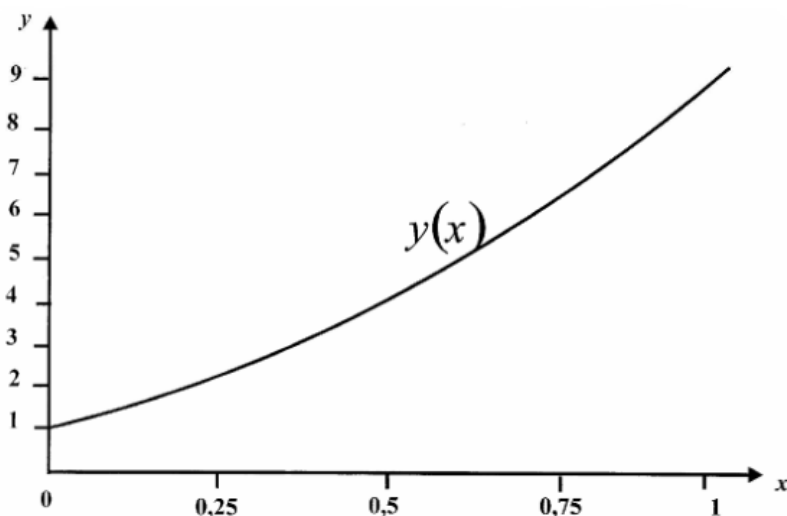


Рисунок 2.21 – Графическое решение задачи Коши



$x_i$	Метод Рунге-Кутта	Точное решение
0.0	1.0	1.0
0.1	1.2221	1.2221
0.2	1.4977	1.4977
0.3	1.8432	1.8432
0.4	2.2783	2.2783
0.5	2.8274	2.8274
0.6	3.5202	3.5202
0.7	4.3928	4.3928
0.8	5.4895	5.4895
0.9	6.8645	6.8645
1.0	8.5836	8.5836

Рисунок 2.22 – Узловые точки решения задачи Коши

### Индивидуальное задание

Построить график и вывести в виде таблицы решение задачи Коши на интервале  $[0; 1]$  методом Рунге-Кутта четвертого порядка согласно вариантам таблицы 2.14.

Таблица 2.14 – Варианты индивидуальных заданий

№	$f(x, y)$	$y_0$
1	$x^3 \sin y + 1$	0,0
2	$x^2 \sin y - 1$	0,1
3	$e^{-x} + 3y$	2,0
4	$\sqrt{y^2 + x^3}$	0,3
5	$\sqrt{y^3 + x^2}$	0,4
6	$1/(1 + y^2) + x^2$	0,0
7	$1/(1 + y^2) + xy$	0,1
8	$\cos y + xy$	0,2
9	$x^2 \cos y + 0.1$	0,3
10	$x^3 \cos y + 0.1$	0,4
11	$\cos(xy) - 0.5$	0,5
12	$e^{-y} + e^x - 2$	0,0
13	$e^{-y} - e^x - 0.1$	0,5
14	$e^{-xy} + 1$	0,4
15	$e^{-xy} + 1$	0,3
16	$\sqrt{y^2 + x^4}$	0,2

Продолжение таблицы 2.14.

№	$f(x, y)$	$y_0$
17	$(xy)^2 + \sqrt{x}$	0,1
18	$\cos(x + y) + x^2$	0,0
19	$\sin(x + y) - x^2$	0,1
20	$\sin(xy) + 1$	0,2
21	$\sin y + xy$	0,3
22	$(x^2 + y^2)x$	0,4
23	$e^x(1 + xy)$	0,5
24	$e^y(1 + xy)$	0,6
25	$\ln(1 + x^2) + y$	0,7
26	$1 + ye^{x^2}$	0,0
27	$\sin x^2 + y^2$	0,1
28	$\cos x^2 + xy$	0,2
29	$\sin[y/(1 + x^2)] + 1$	0,3
30	$\ln(1 + y^2) + 1$	0,4

### 2.1.12 Задание на лабораторную работу № 21

**Тема:** Приближенное вычисление интегралов

**Цель работы:** изучить методы вычисления определенных и неопределенных интегралов, приобрести навыки вычисления интегралов простейшими способами в среде программного пакета Matlab.

#### **Теоретические сведения**

К вычислению определенных интегралов сводятся многие практические задачи электромеханики. На практике не всегда удается воспользоваться известной формулой Ньютона-Лейбница для их вычисления, тогда используют методы численного интегрирования [15]. Они основаны на таких соображениях: с геометрической точки зрения определенный интеграл представляет собой площадь криволинейной трапеции. Идея методов численного интегрирования сводится к разбиению интервала  $[a; b]$  на множество меньших интервалов и нахождению искомой площади, как совокупности элементарных площадей, полученных на каждом частичном промежутке разбиения. В зависимости от использованной аппроксимации получают различные формулы численного интегрирования, имеющие различную точность.

Наиболее популярны методы трапеции и парабол (метод Симпсона).

Метод трапеций предполагает использование линейной аппроксимации, то есть график функции  $y = f(x)$  в виде ломаной, соединяющей точки  $y_i$ . Формула трапеций при постоянном шаге  $h = (b - a)/n$ , где  $n$  – число участков, имеет вид:

$$\int f(x)dx = h((y_0 + y_n)/2 + \sum y_i).$$

В Matlab этот метод реализован программой `trapz(x, y)`.

Метод парабол предполагает замену подынтегральной функции параболой, что соответствует формуле:

$$\int f(x)dx = h/3 \cdot [y_0 + 4(y_1 + y_3 + \dots + y_{n-1}) + 2(y_2 + y_4 + \dots + y_n) + y_n].$$

В Matlab этот метод реализован программой `quad`. Подынтегральная функция может задаваться с помощью дескриптора `@`, тогда она программируется в файле-функции.

Например, необходимо вычислить значение на отрезке  $[0; 1]$  определенного интеграла  $\int dx/(1 + x^2)$ . Ниже на рисунке 2.23 показаны фрагменты и результаты поиска решения двумя методами.

```
>> x = 0 : 0.0001 : 1.0 ;      >> quad ( ' (1./(1+x.^2)) ', 0, 1)
>> y = 1./ (1+x.^2) ;        ans =
>> z = trapz(x, y)           0.7854

z =
    0.7854
```

Рисунок 2.23 – Фрагмент листинга вычисления определенного интеграла

### Индивидуальное задание

Вычислить и вывести на печать значение определенного интеграла методами трапеций и Симпсона согласно данным таблицы 2.15.

Таблица 2.15 – Варианты индивидуальных заданий

№	Функция	Интервал [a; b]
1	$\ln x / x \sqrt{1 + \ln x}$	[1; 3,5]
2	$tg^2 x + ctg^2 x$	$[\pi/6; \pi/3]$
3	$1/x \ln x$	[1,5; 3]

Продолжение таблицы 2.15.

№	Функция	Интервал [a; b]
4	$\ln^2 x/x$	[1; 4]
5	$\sqrt{e^x - 1}$	[0; 2]
6	$xe^x \sin x$	[1; 4]
7	$x \frac{e^x - e^{-x}}{2}$	[0; 2]
8	$1/\sqrt{9+x^3}$	[2; 5]
9	$\sin(1/x)x^4$	[1; 2,5]
10	$x^3 \operatorname{arctg} x$	[0; 2]
11	$\arcsin \sqrt{x/(1+x)}$	[0; 3]
12	$x^2(1+\ln x)$	[1,5; 3]
13	$1/\sqrt{1+3x+2x^2}$	[0; 5]
14	$\sqrt{x^2 - 0.14}/x$	[2,3; 6]
15	$2^{3x} \ln \cos x $	[0; $\pi/2$ ]
16	$(e^{3x} + 1)/(e^x + 1)$	[0; 2]
17	$x \operatorname{arctg} x / \sqrt{1+x^2}$	[0; 2]
18	$\sin^5 x / \ln(1+x^3)$	[0; $\pi/4$ ]
19	$x^2 \sqrt{4-x^2}$	[0; 1,8]
20	$e^x \cos^2 x$	[0; $\pi$ ]
21	$(1.5x^2 + x)/(x^3 + 2)$	[0; 1,2]
22	$(\sin - x^3)/(x^2 + 2.7)$	[2; 4,4]
23	$(3.5 \operatorname{tg} x + x)/(x^3 + 3.7)$	[0; 1,2]
24	$(1.2 - 3.2x^2)/(1 + \sin^2 x)$	[2; 4,5]
25	$(3.17x + 4,2)/\cos^2 x$	[1; 2,2]
26	$(x^2 + 3.2)/(x^5 + 4.7)$	[0; 1,8]
27	$(1.5x^2 + x)/(x^3 + 2)$	[0; 1,2]
28	$e^x \ln \sin x^2 $	[1; 3]
29	$\operatorname{tg} x^2 / e^{-x^2}$	[0; 1]
30	$(1.6x - 2,7)/(1.5x^3 + 3.9)$	[1,1; 2]

### 2.1.13 Задание на лабораторную работу № 22

**Тема:** Численное решение нелинейных уравнений

**Цель работы:** изучить существующие методы решения нелинейных уравнений, приобрести навыки поиска неизвестных нелинейных уравнений в среде программного пакета Matlab.

#### **Теоретические сведения**

Достаточно часто в областях научно-технических решений встречается задача решения нелинейных уравнений. Уравнение типа  $F(x)=0$  или  $x=f(x)$  называется нелинейным. Решить уравнение это значит найти такое  $x$ , при котором уравнение превращается в тождество. В общем случае уравнение может иметь  $0; 1; 2; \dots \infty$  корней. Численные методы решения нелинейных уравнений позволяют находить один корень на заданном интервале  $[a,b]$ . При этом на интервале должен существовать только один корень. Существует множество численных методов поиска решения подобных уравнений: метод простых итераций, метод половинного деления, метод хорд, метод перебора, метод касательных. При начальном поиске решения можно использовать графический метод, который точного решения может и не дать, но на указанном интервале подтвердит существование корней уравнения.

В Matlab данная задача может быть решена следующим образом. Сначала необходимо построить график функции на заданном интервале и убедиться в существовании корня или нескольких корней. Если существует один корень и график пересекает ось абсцисс, то можно применить функцию `fzero` (в основе лежат методы половинного деления, секущей, обратной квадратичной интерполяции). Если  $f(x)$  имеет больше одного корня и может касаться и пересекать ось абсцисс, то необходимо применить более мощную функцию `fsolve` из пакета Optimization Toolbox, которая решает задачу методом наименьших квадратов [15].

Например, необходимо найти корень нелинейного уравнения  $f(x) = 10^x + 2x - 100 = 0$  на интервале  $[1; 2]$ . Листинг программы и результаты представлены на рисунках 2.24 и 2.25 соответственно.

```
>> x = 1.0 : 0.001 : 2.0; y = 10.0.^x + 2.0*x - 100.0;  
>> plot(x, y); grid on  
>> X1 = fzero(' (10.0.^x + 2.0*x - 100.0)', [1.0 2.0])  
>> X2 = fsolve(' (10.0.^x + 2.0*x - 100.0)', 1.0 : 2.0)
```

Рисунок 2.24 – Листинг кода поиска корней уравнения

Для поиска решения сначала вводятся исходные данные уравнения, выполняется построение его графика, и применяются ранее описанные функции, решение с помощью которых дает идентичный результат.

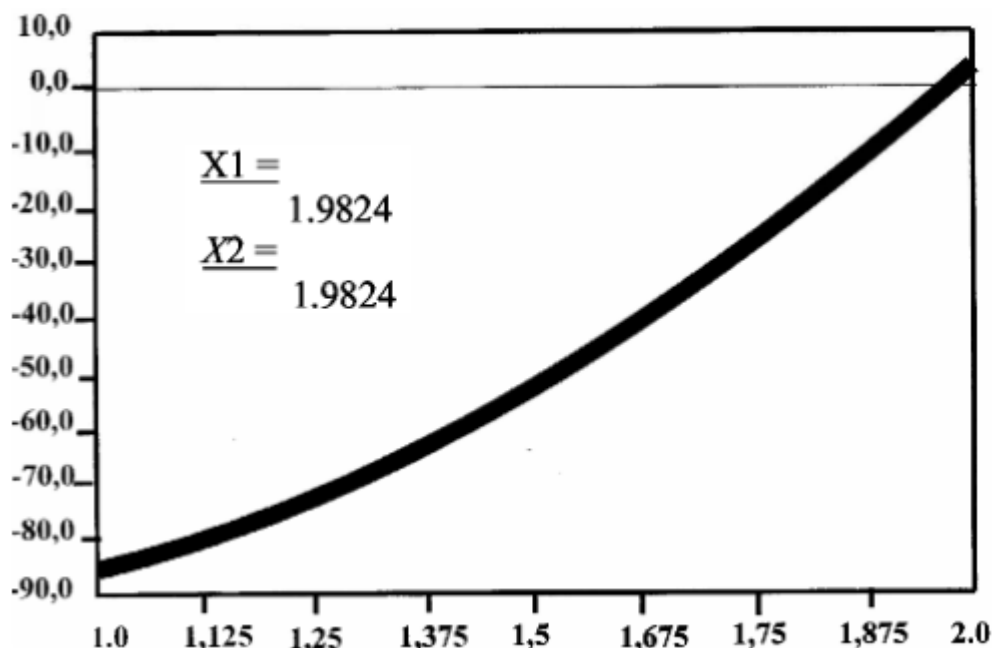


Рисунок 2.25 – График функции и корни уравнения

### Индивидуальное задание

Построить график и найти корни нелинейного уравнения. Варианты задания приведены в таблице 2.16.

Таблица 2.16 – Варианты индивидуальных заданий

№	Уравнение $f(x)=0$	Отрезок $[a; b]$
1	$x \arctg - 1 = 0$	$[1; 2]$
2	$e^{x-2} - \ln(x+2) = 0$	$[2; 3]$
3	$x^3 - 9x^2 + 5x - 6 = 0$	$[8; 9]$
4	$e^x - \frac{1}{x} - 1 = 0$	$[0,5; 1]$
5	$\arctg 2x - \frac{1}{1+x} = 0$	$[0; 1]$
6	$e^x - \ln x - 20 = 0$	$[3; 3,2]$
7	$\sqrt{x} - \operatorname{tg} x(1-x) = 0$	$[0; 1]$
8	$\sin \sqrt{x} - \cos \sqrt{x} + 2\sqrt{x} = 0$	$[0; 0,2]$
9	$x^4 + 2x^3 - x - 1 = 0$	$[0,8; 1]$
10	$x^3 - e^{4x} - 5.5 = 0$	$[2,6; 3]$

Продолжение таблицы 2.16.

№	Уравнение $f(x)=0$	Отрезок $[a; b]$
11	$x^6 - 3x^2 + x - 1 = 0$	[1; 1,5]
12	$\sqrt[3]{5-x} - x = 0$	[1; 2]
13	$x^2 - \ln x = 0$	[0; 1]
14	$x^2 - \cos x = 0$	[0; 1]
15	$\ln x - \operatorname{arctg} x = 0$	[3; 4]
16	$x^2 \operatorname{arctg} x - 1 = 0$	[1; 1,2]
17	$x^2 + \ln x - 4 = 0$	[1; 2]
18	$x - \operatorname{arctg} \sqrt[3]{x} = 0$	[0; 1]
19	$x^2 - e^x - 2 = 0$	[-0,2; -0,1]
20	$x^2 - \ln(x+1) = 0$	[0,1; 0,9]
21	$x^5 - x - 2 = 0$	[1; 1,4]
22	$x - 2 - \sqrt[4]{x} = 0$	[3; 4]
23	$x - \operatorname{tg} x = 0$	[0; 1,5]
24	$x + \ln x - 0.5 = 0$	[0; 1]
25	$\ln x + \sqrt{x} = 0$	[0,1; 1]
26	$\sqrt{x} - \cos \sqrt{x} = 0$	[0,4; 0,6]
27	$x^2 - \ln(1+x^2) - 9.75 = 0$	[3; 4]
28	$x + \sqrt[3]{x} - 6.09 = 0$	[4; 5]
29	$x^3 - \sqrt{x} - 9.5 = 0$	[2; 3]
30	$\arccos 2x - x^2 - 0.35 = 0$	[0; 0,5]

#### 2.1.14 Задание на лабораторную работу № 23

**Тема:** Основы визуализации вычислений в среде программного пакета Matlab.

**Цель работы:** изучить способы визуализации вычислений, приобрести навыки графического представления результатов вычислений различными способами в разных системах координат в среде программного пакета Matlab.

#### **Теоретические сведения**

Графические средства Matlab ориентированы на построение двух- и трехмерных графиков функциональных зависимостей в разных формах. Построение графиков осуществляется в специальном объекте –

графическом окне (figure) – создание которого происходит при реализации оператора figure или автоматически перед выполнением какой-либо графической функции [16].

Основные команды для работы с графическим окном следующие:

- а) figure – создание графического окна;
- б) figure(n) – создание графического окна с номером n;
- в) close – закрытие текущего графического окна;
- г) close all – закрытие всех открытых графических окон;
- д) close('name') – закрытие графического окна с указанным именем;
- е) subplot(m,n,i) – разбиение графического окна на части ( $m \leq n \leq 4$ );
- ж) clf – очистка текущего графического окна.

Наиболее распространенной функцией для построения графиков в декартовой системе координат является функция plot, которая может быть представлена в следующем виде:

- а) plot(y) – выполняет построение элементов вектора y в функции их порядкового номера;
- б) plot(x,y) – выполняет построение графика  $y=f(x)$ ;
- в) plot(z) – выполняет построение комплексных чисел;
- г) plot(x1, y1, s1, x2, y2, s2, ..., xn, yn, sn) – выполняет построение графиков с несколькими аргументами и функциями в одной системе координат.

По умолчанию графики строятся отрезками непрерывных линий, которые соединяют между собой соседние точки. Для большей гладкости графика необходимо брать не менее 50-100 точек.

Цвет кривой выбирается автоматически, но можно при необходимости задавать пользовательские параметры (LineStyle), такие как цвет, тип линии или маркера (таблица 2.17).

В таблице 2.17 приведены стили графиков системы Matlab.

Таблица 2.17 – Стили графиков

Тип точки		Цвет линии		Тип линии	
.	Точка	Y	Желтый	-	Сплошная
O	Окружность	M	Фиолетовый	:	Двойной пунктир
x	Крест	C	Голубой	-.	Штрих-пунктир
+	Плюс	R	Красный	--	Штриховая
*	Снежинка	G	Зеленый		
S	Квадрат	B	Синий		
D	Ромб	W	Белый		
V, ^, <, >	Треугольник вверх,вниз, влево, вправо	K	Черный		
P	Звезда				



В одной функции plot можно использовать несколько групп аргументов и тогда все графики изображаются в единой системе координат в одинаковом масштабе. Есть возможность построения графиков в логарифмическом масштабе (semilogx(x,y), semilogy(x,y), loglog(x,y)). Также можно строить графики двух функций с разным масштабом (plotyy, fplot). Гистограммы разных форм представления строят с помощью функций следующих функций:

- а) bar(y), bar(x,A), bar3(y) - диаграмма-столбец;
- б) pie(y), pie(y, labels), pie(y, explode), pie3(y) – круговая диаграмма;
- в) stairs(x,y) – ступенчатые диаграммы.

Иногда необходимо выполнять построение графиков в полярной системе координат, тогда стоит воспользоваться функцией polar(fi,ro), которая осуществляет построение графика в радианах.

Также в арсенале Matlab есть целый ряд вспомогательных функций, отвечающих за форматирование графических изображений:

- а) grid on/off – включение/выключение координатной сетки;
- б) hold on/off – включение/выключение режима наложения графиков в текущем графическом окне;
- в) title('name') – вывод заголовка в верхней части окна;
- г) xlabel('name'), ylabel('name') – подписи осей x и y;
- д) legend('name1', 'name2', 'name3', ...) – подпись названий графиков в той последовательности, в которой выполнено их построение;
- е) text(m,n, 'текст') – вывод текста рядом с точкой с указанными координатами;
- ж) gtext – вывод текста рядом с точкой, указанно мышью;
- з) axis([xmin xmax ymin ymax]) – установка указанных границ графического окна по указанным параметрам.

Рассмотрим пример построения графика функции  $y = \sin x \cdot e^{-x}$ .

В окне Command Window задается программа:

```
>> x=-5:0.5:5; % задание промежутка [-5;5] с шагом 0,5
>> plot(x,exp(x),['R','*','-.']) % выведение графика функции
% exp(x) красного цвета (R), точки графика в виде снежинок (*), линии
% штрихпунктирные (-.)
>> grid on % задание сетки
```

Для блокировки вывода результата вычисления некоторого выражения в командное окно после него надо установить знак ;

График функции приведен на рисунке 2.26.

Трехмерные поверхности обычно описываются функцией двух переменных  $z = f(x, y)$ . Специфика построения трехмерных графиков требует не просто задания ряда значений x и y, то есть векторов x и y, а определения двумерных массивов X и Y. Для создания таких массивов служит функция meshgrid.

[X,Y]=meshgrid(x,y) – преобразует область, заданную векторами x

и  $y$ , в двумерные массивы  $X$  и  $Y$ , которые могут быть использованы для вычисления значений функции двух переменных и построения трехмерных графиков.

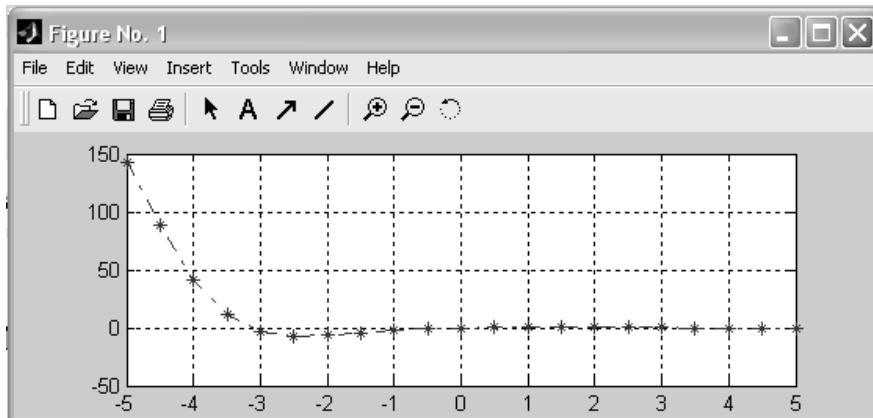


Рисунок 2.26 – График функции  $y = \sin x \cdot e^{-x}$

Эта функция формирует массивы  $X$  и  $Y$  таким образом, что строки выходного массива  $X$  являются копиями вектора  $x$ , а столбцы выходного массива  $Y$  – копиями вектора  $y$ .

Например, необходимо выполнить следующее построение:

```
[X,Y]=meshgrid(1:1:4,6:1:9)
```

```
z=x.^2+y.^2;
```

```
contour(x,y,z,12)
```

В этом примере формируются массивы  $X$  и  $Y$  для построения трехмерной поверхности при изменении  $x$  от 1 до 4 с шагом 1 и  $y$  от 6 до 9 с шагом 1, но в данном случае получаем изображение линий уровня (рисунок 2.27).

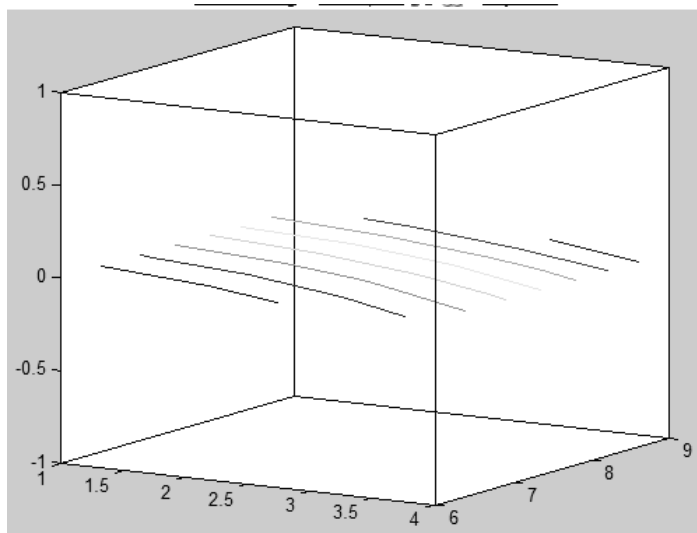


Рисунок 2.27 – Линии уровня

Команда `plot3(...)` является аналогом команды `plot(...)`, но относится к функции двух переменных  $z = f(x, y)$ . Она строит аксонометрическое изображение трехмерных (3D) поверхностей.

`Plot3(X,Y,Z)` вычерчивает различные строки, полученные из столбцов двумерных массивов  $X, Y, Z$  одинаковой размерности. Массивы  $X$  и  $Y$  можно получить с помощью `meshgrid`.

Например, выполним построение трехмерного графика средствами функции `plot3` (рисунок 2.28):

```
[x,y]=meshgrid(-2:0.1:2,-2:0.1:2);  
z=x.^2+y.^2;  
plot3(x,y,z)  
grid on.
```

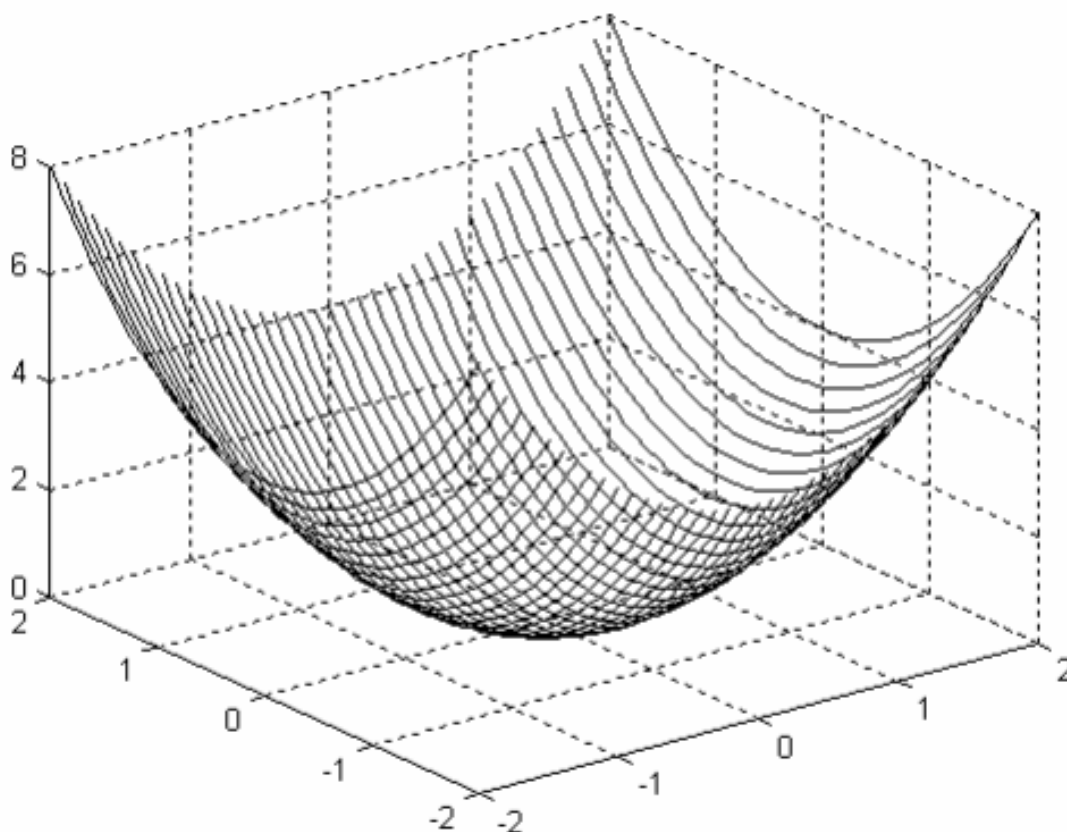


Рисунок 2.27 – Трехмерный график в функции `plot3`

Однако более наглядными являются сеточные графики трехмерных поверхностей с заданной или функциональной окраской. Такие графики выполняются командой `mesh`.

`Mesh(X,Y,Z,C)` – выводит в графическое окно сетчатую поверхность с цветами узлов поверхности, заданных массивом  $C$ .

`Mesh(X,Y,Z)` – аналог предшествующей команды при  $C=Z$ , с использованием функциональной окраски, при которой цвет задается высотой поверхности.

Например:

```
[x,y]=meshgrid(-2:0.1:2,-2:0.1:2);
```

```
z=x.^2+y.^2;
```

```
mesh(x,y,z).
```

В результате выполнения этой программы на экран будет выведена фигура, представленная на рисунке 2.29. Рисунок представлен в черно-белой палитре, хотя в действительности он формируется в цветной палитре.

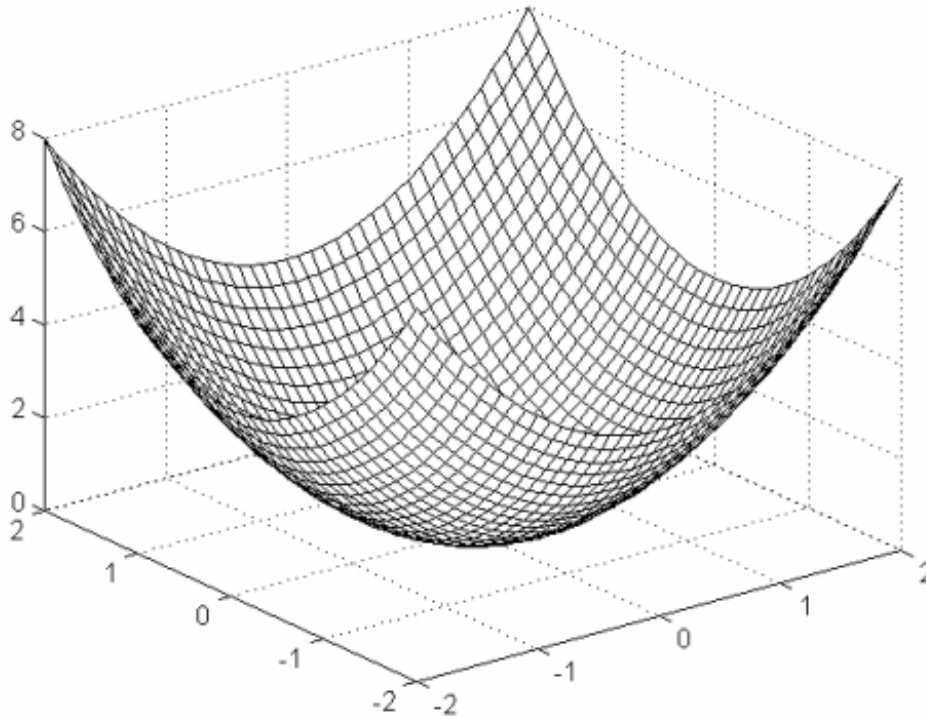


Рисунок 2.29 – Трехмерный график, построенный с помощью функции *meshgrid*

### ***Индивидуальное задание***

1 Создать функцию пользователя  $y=f(x)$ , вычислить ее значение в точке  $x_0$  и построить график (таблица 2.18).

2 Построить график функции  $f = \sin x$  (четный вариант) или  $f = \cos x$  (нечетный вариант) на интервал  $0 \leq x \leq 3\pi$ . В этом же графическом окне построить ступенчатый график с шириной ступеньки  $\Delta x = \pi/4$ .

3 Задать массив произвольных чисел  $Z$  (10-20 элементов) и построить круговую диаграмму.

4 Выполнить графическое отображение в трехмерном пространстве функции двух переменных  $z=f(x, y)$ , приведенной в таблице 2.19 средствами функций *meshgrid*, *plot3*, в виде сетчатых поверхностей средствами функций *mesh*, *surf*, в виде линий уровня средствами функций *contour*, *contour3*.

Таблица 2.18 – Варианты индивидуальных заданий

№	Функция	$x_0$	№	Функция	$x_0$
1	$y = \frac{e^{-x}}{\sqrt{e^{-x} + 1}} - x$	5,5	16	$y = e^{-x} \sqrt{1 + x + x^2} - x^2$	4,5
2	$y = \frac{x^2 - 1}{\ln(x^2 - 1)} + x$	2,75	17	$y = \sqrt{x} - 1 - \cos(0,5x)$	3,75
3	$y = shx + \sin x - 1$	3,1	18	$y = x^2 \ln(1 + x^2) - x$	3,6
4	$y = \frac{1}{x(1 - \ln x)} - 2$	4,21	19	$y = e^{\sqrt{\sin x}} - 1,5$	5,21
5	$y = \frac{2}{3} \sin^2 2x - \frac{3}{4} \cos^2 2x$	6,32	20	$y = \sin 2x - \frac{3}{4} \cos 2x$	2,32
6	$y = \frac{\sin^2 x}{\cos^4 x} - 1$	4,75	21	$y = \cos^4 x \frac{\sin^2 x}{5} - 2$	4,2
7	$y = e^{x^3} \sqrt[3]{x^2} - x - 1$	2,35	22	$y = e^x - 3x - 5$	3,35
8	$y = x^3 \sqrt[3]{(1-x)^2} - 1$	8,29	23	$y = x^2 \sqrt[3]{(1-3x)^2}$	4,2
9	$y = \frac{1}{\sqrt{e^{-x} + 1}} - 3x$	1,5	24	$y = e^{-x} \sqrt{1 + x^2} - x$	5,2
10	$y = \frac{x^2 - 1}{(3x^2 - 1)} + 4x$	2,7	25	$y = \sqrt{x} - 7 - \cos(5x)$	6,75
11	$y = \sin x - 2x$	3,4	26	$y = x \ln(1 + x) - 2x$	3,8
12	$y = \frac{1}{(1 - \ln x)} - 2$	6,21	27	$y = e^{2\sqrt{\sin x}} - 3$	2,6
13	$y = \sin^2 3x - \cos 2x + 4$	2,32	28	$y = \sin x - \frac{1}{4} \cos 3x$	5,3
14	$y = \frac{2}{\cos^4 x} - 10$	3,7	29	$y = \cos^3 x \frac{\sin^2 x}{2} - 8$	1,75
15	$y = e^{\sqrt[3]{x^2}} - x - 1$	2,3	30	$y = e^x + 2x - 4$	6,5

Таблица 2.19 – Варианты индивидуальных заданий

№	Функция $z=f(x, y)$	№	Функция $z=f(x, y)$
1	$x \cdot e^{-(x^2+y^2)}$	16	$\frac{\cos y}{\sqrt{x^2+y^2}}$
2	$X^2 - Y^2$	17	$(x^3 + y^3) \cdot  x + y $
3	$\frac{\sin x^2}{\sqrt{x^2+y^2}}$	18	$\frac{x^3 - y^3}{\sqrt{x^2+y^2}}$
4	$\sqrt{ x+y } \cdot \ln x^2$	19	$\sqrt{X^2 + Y^2}$
5	$\sqrt[3]{X^2 + Y^2}$	20	$e^{\sqrt{X^2+Y^2}}$
6	$\frac{x^2 - y^2}{x^2 + y^2}$	21	$\frac{x + y}{\sqrt{x^2 + y^2}}$
7	$X^2 + Y^2 - XY$	22	$XY$
8	$e^x \cdot \sin(x + y)$	23	$e^{XY}$
9	$(X + Y)^2$	24	$e^{-XY}$
10	$e^{-(X+Y)}$	25	$\cos\left(\frac{XY}{2\pi}\right)$
11	$(X - Y)^2$	26	$\sin\left(\frac{X^2+Y^2}{\pi}\right)$
12	$X^3 + Y^3$	27	$X^3Y - X - Y$
13	$X^2 + Y^2$	28	$:(X + Y)^3$
14	$(X + Y)^4 - XY$	29	$e^{-(X^2+Y^2)}$
15	$e^{-(X^2+Y^2)}XY$	30	$\frac{\sin x}{\sqrt{x+y}}$

2.1.15 Задание на самостоятельную работу № 1

**Тема:** Операции с комплексными числами

**Цель работы:** изучить операции с комплексными числами, приобрести навыки обработки комплексных чисел в среде программного пакета Matlab.

## *Теоретические сведения*

Для работы с комплексными числами применяются предварительно определенные переменные  $i$ ,  $j$ , которые равны мнимой единице  $\sqrt{-1}$ . При необходимости их необходимо вычислять явно, например,  $i = \text{sqrt}(-1)$ . Задание значения комплексными константами осуществляется по правилам записи алгебраических выражений в Matlab:

$$\begin{aligned}h &= -2 + 5 \cdot i \\d &= 4.5673 - 3.8765 \cdot j.\end{aligned}$$

Над комплексными данными в Matlab можно выполнять такие же арифметические операции, что и для действительных чисел.

В пакете Matlab применяются следующие функции комплексного аргумента:

- а)  $\text{abs}(z)$  – модуль комплексного числа;
- б)  $\text{complex}(x, y)$  – формирование комплексного числа  $x+yi$ ;
- в)  $\text{angle}(z)$  – аргумент;
- г)  $\text{conj}(z)$  – комплексно-сопряженное число к числу  $z$ ;
- д)  $\text{real}(z)$  – действительная часть;
- е)  $\text{imag}(z)$  – мнимая часть.

Для решения квадратного уравнения применяется функция  $\text{roots}(P)$ , где  $P = [a, b, c]$  – вектор коэффициентов уравнения.

Изображение комплексного числа на  $z$  на комплексной плоскости может быть реализовано с помощью команды  $\text{plot}(z)$ .

Для построения точки  $x$  на комплексной плоскости можно использовать функции  $\text{plot}(x, 'r^*')$ ,  $\text{polar}(\text{angle}(x), \text{abs}(x), 'go')$ ,  $\text{compass}(x)$ .

### *Индивидуальное задание*

Найти корни  $x_1$  и  $x_2$  квадратного уравнения  $ax^2 + bx + c = 0$  (для значений коэффициентов  $a$ ,  $b$  и  $c$ , заданы в таблице 2.20) двумя способами – алгебраически и с помощью функции  $\text{roots}$ , провести их сравнительный анализ.

Проверить правильность вычислений по теореме Виета.

По значению корней вычислить величину комплексной переменной  $P$  по выражению таблицы 2.20 и определить модуль и аргумент числа  $P$ . Построить на комплексной плоскости точки  $x_1$ ,  $x_2$  в виде «звездочек» и  $P$  в виде «окружности».

Таблица 2.20 – Варианты индивидуальных заданий

№	a	b	c	Выражение P	№	a	b	c	Выражение P
1	1	0,3	0,2	$\sqrt{-3} \frac{x_1 - x_2}{x_1 x_2}$	13	7,3	10	6,2	$\frac{\operatorname{Re}[x_1] + i \operatorname{Re}[x_2]}{x_1 x_2}$
2	5	3	1	$e^2 + \left  \frac{x_1}{2x_2} \right  i$	14	4,2	11	11	$\frac{ix_1 + \sqrt{ x_2 }}{\operatorname{Re}[x_1]}$
3	1,3	1,8	1,5	$x_1(3.2 + 5i) - \frac{1}{x_2^2}$	15	6,7	8,9	4,9	$\frac{x_1 + x_2}{x_1 x_2} + \sqrt{ x_1  \cdot  x_2 }$
4	15	2	3	$\frac{\operatorname{Re}[x_2]}{x_1^3(1+i)}$	16	8,3	12	11	$\sin(\operatorname{Re}[x_1]) + \frac{x_1}{ix_2}$
5	5,2	4,8	7,6	$\lg x_1  + \frac{2-6i}{x_2}$	17	5,7	3,2	11	$\frac{x_1 + \sqrt{ x_2 }}{2 \sin(3 + \operatorname{Re}[x_1])}$
6	17	16	21	$\sqrt{-5 \cdot \operatorname{Im}[x_2]} - \frac{x_1}{5x_2}$	18	31	22	4,2	$\operatorname{Im}[3i + x_1] + \frac{\sqrt{-7}}{ x_2 }$
7	27	32	31	$\frac{5+3i}{x_1+2x_2} - \sqrt{ x_1 x_2 }$	19	11	9,3	12	$\frac{x_1 x_2 - \operatorname{Im}[x_1 x_2]}{x_1 + x_2}$
8	5,2	3,1	7,1	$\frac{(x_1 + 3x_2)(3x_1 - x_2)}{(x_1 - x_2)i}$	20	6,7	5,3	17	$\frac{x_1 + x_2}{-5(x_1 - x_2)} + \sqrt{\frac{ x_1 }{ x_2 }}$
9	8	8	4	$\frac{\cos x_1  + i \sin x_2 }{x_1 x_2}$	21	4,3	21	56	$\sqrt{-13} \frac{ x_1 + x_2 }{\operatorname{Im}[x_1 + x_2]}$
10	6,4	9,2	7,3	$\frac{\sqrt{ x_1  + x_2}}{-3(x_1 + x_2)}$	22	20	10	1,5	$\frac{i \cos x_1  + \sqrt{ x_2 }}{2x_1}$
11	7,2	5,8	7,8	$\frac{2x_1}{x_2} + \operatorname{Re}[x_1] - \operatorname{Im}[x_2]$	23	94	7,3	82	$\sin \left  \frac{3x_1 - \sqrt{ x_2 }}{2x_1} \right  + ix_2$
12	3,2	12	21	$\ln \left  \frac{x_1}{x_2} \right  + i \sqrt{ x_2 }$	24	1,2	1,9	1,7	$\sqrt{-5} \frac{2x_1 + x_2}{i x_2 }$

### 2.1.16 Задание на самостоятельную работу № 2

**Тема:** Погрешность функций.

**Цель работы:** приобрести навыки работы с приближенными числами в применении формул погрешностей элементарных действий и функций в среде программного пакета Matlab.



## *Теоретические сведения*

Достаточно часто при обработке экспериментальных данных перед исследователем встает вопрос точности обработки данных – погрешности измерения величин.

При анализе экспериментальных измерений следует четко разграничивать два понятия: истинные значения физических величин и их эмпирические проявления – результаты измерений.

Истинные значения физических величин – это значения, идеальным образом отражающие свойства данного объекта, как в количественном, так и в качественном отношении. Они не зависят от средств нашего познания и являются абсолютной истиной.

Результаты измерений, напротив, являются продуктами нашего познания. Представляя собой приближенные оценки значений величин, найденные путем измерения, они зависят не только от них, но еще и от метода измерения, от технических средств, с помощью которых проводятся измерения, и от свойств органов чувств наблюдателя, осуществляющего измерения.

Разница  $\Delta$  между результатами измерения  $X'$  и истинным значением  $Q$  измеряемой величины называется погрешностью измерения  $\Delta = X' - Q$ .

Но поскольку истинное значение  $Q$  измеряемой величины неизвестно, то неизвестны и погрешности измерения, поэтому для получения хотя бы приближенных сведений о них приходится вместо истинного значения использовать в расчетах так называемое действительное значение.

Под действительным значением физической величины понимается ее значение, найденное экспериментально и настолько приближающееся к истинному, что для данной цели оно может быть использовано вместо него.

Причинами возникновения погрешностей являются: несовершенство методов измерений, технических средств, применяемых при измерениях, и органов чувств наблюдателя. В отдельную группу следует объединить причины, связанные с влиянием условий проведения измерений. Последние проявляются двояко. С одной стороны, все физические величины, играющие какую-либо роль при проведении измерений, в той или иной степени зависят друг от друга. Поэтому с изменением внешних условий изменяются истинные значения измеряемых величин. С другой стороны, условия проведения измерений влияют и на характеристики средств измерений и физиологические свойства органов чувств наблюдателя и через их посредство становятся источником погрешностей измерения.

Описанные причины возникновения погрешностей определяются совокупностью большого числа факторов, под влиянием которых складывается суммарная погрешность измерения. Их можно объединить в две основные группы:

а) факторы, проявляющиеся весьма нерегулярно и столь же неожиданно исчезающие или проявляющиеся с интенсивностью, которую трудно предвидеть. К ним относятся, например, перекосы элементов приборов в их направляющих, нерегулярные изменения моментов трения в опорах, малые флюктуации влияющих величин, изменения внимания операторов и др.;

б) факторы, постоянные или закономерно изменяющиеся в процессе измерительного эксперимента, например плавные изменения влияющих величин или погрешности применяемых при измерениях образцовых мер.

Пусть  $x$  – приближённое значение некоторой величины (например, полученное путём однократного измерения этой величины), а  $x_0$  – её точное значение

Абсолютной погрешностью величины называется разность:

$$\Delta x = |x - x_0|.$$

Относительной погрешностью приближённого числа называется отношение абсолютной погрешности приближённого числа к самому этому числу:

$$\varepsilon(x) = \frac{\Delta x}{x_0}.$$

В научных экспериментах многие величины определяются не непосредственно, а косвенным путём – измерением значений других величин.

Абсолютная погрешность суммы двух независимых величин равна сумме абсолютных погрешностей отдельных слагаемых:

$$\Delta(x + y) = \Delta x + \Delta y.$$

Абсолютная погрешность разности двух независимых величин равна сумме абсолютных погрешностей уменьшаемого и вычитаемого:

$$\Delta(x - y) = \Delta x + \Delta y.$$

Относительные погрешности при сложении и вычитании складывать нельзя. Относительная погрешность произведения приближённо равна сумме относительных погрешностей отдельных сомножителей:

$$\varepsilon(xy) \approx \varepsilon(x) + \varepsilon(y).$$

Рассмотрим примеры, позволяющие определять погрешности вычислений.

Определить, какое равенство точнее:  $9/11 = 0,818$ ;  $\sqrt{18} = 4,24$ .

Найдем значения данных выражений с большим числом десятичных знаков. Для этого выполним следующие действия:

```
>> format long %длинное представление числа (15 знаков)
```

```
>> a1=9/11
```

```
a1 = 0.81818181818182
```

```
>> a2=sqrt(18)
```

```
a2 =
```

```
4.24264068711928
```

Затем вычислим предельные абсолютные погрешности:

```
>> abs(a1-0.818)
```

```
ans =
```

```
1.818181818182829e-004
```

```
>> abs(a2-4.24)
```

```
ans =
```

```
0.00264068711928
```

Округлим их с избытком:

$$\Delta a_1 = 0,00019, \Delta a_2 = 0,0027.$$

Вычислим предельные относительные погрешности:

```
>> 0.00019/0.818
```

```
ans =
```

```
2.322738386308069e-004
```

```
>> 0.0027/4.24
```

```
ans =
```

```
6.367924528301887e-004
```

Таким образом,

$$\delta a_1 = \frac{\Delta a_1}{a_1} = \frac{0,00019}{0,818} = 0,00024 = 0,024\%;$$

$$\delta a_2 = \frac{\Delta a_2}{a_2} = \frac{0,0027}{4,24} = 0,00064 = 0,064\%.$$

Так как  $\delta a_1 < \delta a_2$ , то равенство  $9/11 = 0,818$  является более точным.

Округлить сомнительные цифры числа, оставив верные знаки: 2,3544;  $\delta = 0,2\%$ .

Пусть  $a = 2,3544$ ;  $\delta a = 0,2\%$ ; тогда  $\Delta a = a \cdot \delta a = 0,00471$ . В данном числе верными являются три цифры, поэтому округляем его, сохраняя эти цифры:

$$a_1 = 2,35; \quad \Delta a_1 = \Delta a + \Delta_{окр} = 0,0044 + 0,00471 = 0,00911 < 0,01.$$

Значит, и в округленном числе 2,35 все три цифры верны.

Найти предельную абсолютную и относительную погрешности числа 12,384, если оно имеет только верные цифры.

Так как все пять цифр числа  $a=12,384$  верны, то

$$\Delta a = 0,001; \quad \delta a = \frac{0,001}{12,384} = 0,0001 = 0,01\%.$$

Вычислить и определить погрешности результата  $N = \frac{(n-1)(m+n)}{(m-n)^2}$ ,

где  $n=3,0567(\pm 0,0001)$ ,  $m=5,72(\pm 0,02)$ .

Имеем:

$$n - 1 = 2,0567(\pm 0,0001),$$

$$m + n = 3,0567(\pm 0,0001) + 5,72(\pm 0,02) = 8,7767(\pm 0,0201),$$

$$m - n = 5,72(\pm 0,02) - 3,0567(\pm 0,0001) = 2,6633(\pm 0,0201),$$

$$N = \frac{2,0567 \cdot 8,7767}{2,6633^2} = 2,545 \approx 2,55,$$

$$\delta N = \frac{0,0001}{2,0567} + \frac{0,0201}{8,7767} + 2 \cdot \frac{0,0201}{2,6633} = 0,0175 = 1,75\%,$$

$$\Delta N = 2,55 \cdot 0,0175 = 0,045.$$

Ответ:  $N \approx 2,55(\pm 0,045)$ ;  $\delta N = 1,75\%$ .

### *Индивидуальное задание*

Согласно варианту (таблица 2.21):

- а) определить, какое равенство точнее;
- б) округлить сомнительные цифры числа, оставив верные знаки;
- в) найти предельную абсолютную и относительную погрешности числа, если они имеют только верные цифры;
- г) вычислить и определить погрешности результата.

*Таблица 2.21 – Варианты индивидуальных заданий*

№	Задание
1, 12	1) $\sqrt{44} = 6,63$ ; $19/41 = 0,463$ . 2) $2,8546$ ; $\delta = 0,3\%$ . 3) $42,884$ . 4) $X = \left[ \frac{(a+b)c}{m-n} \right]^2$ , где $a = 4,3(\pm 0,05)$ , $b = 17,21(\pm 0,02)$ , $c = 8,2(\pm 0,05)$ , $m = 12,417(\pm 0,003)$ , $n = 8,37(\pm 0,005)$ .

Продолжение таблицы 2.21.

№	Задание
2, 13	1) $\sqrt{30} = 5,48$ ; $7/15 = 0,467$ . 2) $6,4257(\pm 0,0024)$ . 3) $0,537$ . 4) $X = \frac{m^3(a+b)}{c-d}$ , где $a = 13,5(\pm 0,02)$ , $b = 3,7(\pm 0,02)$ , $c = 34,5(\pm 0,02)$ , $m = 4,22(\pm 0,004)$ , $d = 23,725(\pm 0,005)$ .
3, 14	1) $\sqrt{10,5} = 3,24$ ; $4/17 = 0,235$ . 2) $0,5748(\pm 0,0034)$ . 3) $2,043$ . 4) $X = \frac{(a+b)m}{(c-d)^2}$ , где $a = 2,754(\pm 0,001)$ , $b = 11,7(\pm 0,04)$ , $c = 10,536(\pm 0,002)$ , $m = 0,56(\pm 0,005)$ , $d = 6,32(\pm 0,008)$ .
4, 15	1) $\sqrt{10} = 3,16$ ; $15/7 = 2,14$ . 2) $0,34484$ ; $\delta = 0,4\%$ . 3) $0,745$ . 4) $X = \frac{(a+b)m}{\sqrt{c-d}}$ , где $a = 23,16(\pm 0,02)$ , $b = 8,23(\pm 0,005)$ , $c = 145,5(\pm 0,08)$ , $m = 0,28(\pm 0,006)$ , $d = 28,6(\pm 0,1)$ .
5, 16	1) $\sqrt{4,8} = 2,19$ ; $6/7 = 0,857$ . 2) $10,8441$ ; $\delta = 0,5\%$ . 3) $0,288$ . 4) $X = \frac{(a-b)c}{\sqrt{m+n}}$ , где $a = 27,16(\pm 0,006)$ , $b = 5,03(\pm 0,01)$ , $c = 3,6(\pm 0,02)$ , $m = 12,375(\pm 0,004)$ , $n = 86,2(\pm 0,05)$ .
6, 17	1) $\sqrt{6,8} = 2,61$ ; $12/11 = 1,091$ . 2) $0,12356(\pm 0,00036)$ . 3) $3,4453$ . 4) $X = \frac{a+b}{\sqrt{(c-d)m}}$ , где $a = 16,342(\pm 0,001)$ , $b = 2,5(\pm 0,03)$ , $c = 38,17(\pm 0,002)$ , $m = 3,6(\pm 0,04)$ , $d = 9,14(\pm 0,005)$ .

Продолжение таблицы 2.21.

№	Задание
7, 18	1) $\sqrt{22} = 4,69$ ; $2/21 = 0,095$ . 2) $24,5643$ ; $\delta = 0,1\%$ . 3) $4,348$ . 4) $S = \frac{1}{64} \pi \sqrt{D^4 - d^4}$ , где $D = 36,5(\pm 0,1)$ , $d = 26,35(\pm 0,005)$ , $\pi = 3,14$ .
8, 19	1) $\sqrt{9,8} = 3,13$ ; $23/15 = 1,53$ . 2) $8,3445(\pm 0,0022)$ . 3) $0,576$ . 4) $X = \frac{m\sqrt{a-b}}{c+d}$ , где $a = 9,542(\pm 0,001)$ , $b = 3,128(\pm 0,002)$ , $c = 0,172(\pm 0,001)$ , $m = 2,8(\pm 0,03)$ , $d = 5,4(\pm 0,02)$ .
9, 20	1) $\sqrt{83} = 9,11$ ; $6/11 = 0,545$ . 2) $3,7834(\pm 0,0041)$ . 3) $0,678$ . 4) $y = \frac{\sqrt[3]{a-b}}{m(n-a)}$ , где $a = 10,82(\pm 0,03)$ , $b = 2,786(\pm 0,0006)$ , $m = 0,28(\pm 0,006)$ , $n = 14,7(\pm 0,06)$ .
10, 21	1) $\sqrt{52} = 7,21$ ; $17/19 = 0,895$ . 2) $7,521$ ; $\delta = 0,12\%$ . 3) $0,0748$ . 4) $Q = \frac{(2n-1)^2(x+y)}{x-y}$ , где $n = 2,0435(\pm 0,0001)$ , $x = 4,2(\pm 0,05)$ , $y = 0,82(\pm 0,01)$ .
11, 22	1) $\sqrt{44} = 6,63$ ; $21/29 = 0,723$ . 2) $13,6253(\pm 0,0021)$ . 3) $2,16$ . 4) $X = \left[ \frac{(a+b)c}{m-n} \right]^2$ , где $a = 5,2(\pm 0,04)$ , $b = 15,32(\pm 0,01)$ , $c = 7,5(\pm 0,05)$ , $m = 21,823(\pm 0,002)$ , $n = 7,56(\pm 0,003)$ .

### 2.1.17 Задание на самостоятельную работу № 3

**Тема:** Расчет переходных процессов в электрических цепях в приложении Simulink пакета Matlab.

**Цель работы:** приобрести навыки моделирования электрических цепей с помощью приложения Simulink пакета Matlab.

#### *Теоретические сведения*

Simulink Matlab предназначен для структурного моделирования различных систем. Выходными параметрами для моделирования служат системы дифференциальных уравнений, которая описывает процессы в исследуемой системе в установившихся и переходных режимах. На основе математического описания получается структурная схема, которая представляется в среде пакета Matlab Simulink в виде структурной модели, получаемой в несколько этапов [14]:

- а) выходная система дифференциальных уравнений записывается в нормальной форме Коши;
- б) преобразуется в операторную форму ( $d/dt=p$  – оператор Лапласа);
- в) из каждого уравнения составляются структурно одинаковые блоки формирования сигналов производных на входе интеграторов;
- г) идентичные сигналы соединяются линиями связи.

Структурные модели строятся с помощью соответствующих стандартных блоков (в Simulink оператор Лапласа обозначается через  $s$ ). Перечень наиболее часто используемых блоков приведен на рисунке 2.30.

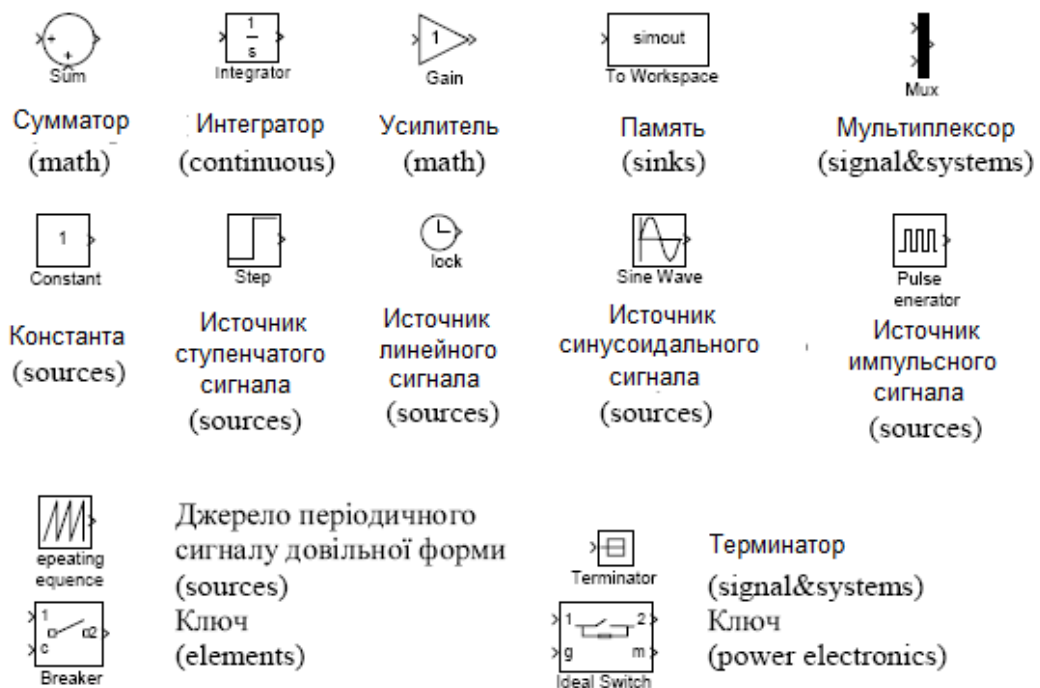


Рисунок 2.30 – Блоки Simulink Matlab

Рассмотрим электрическую цепь (рисунок 2.31), для которой необходимо рассчитать переходные процессы при замыкании ключа ( $i_1(t)$ ,  $i_2(t)$ ,  $i_3(t)$ ,  $U_c(t)$ ) [16].

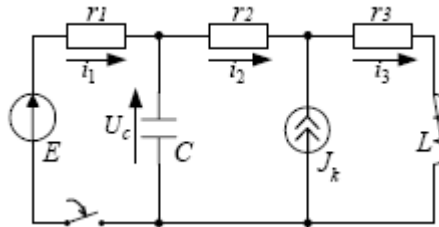


Рисунок 2.31 – Электрическая цепь

Согласно законам Кирхгофа математическое описание цепи:

$$E = i_1 r_1 - U_c,$$

$$E = i_1 r_1 + i_2 r_2 + i_3 r_3 + L \frac{di_3}{dt},$$

$$i_1 + C \frac{dU_c}{dt} = i_2,$$

$$i_2 + J_k = i_3.$$

Рассчитаем начальные условия. При разомкнутом ключе цепь имеет вид, представленный на рисунке 2.32.

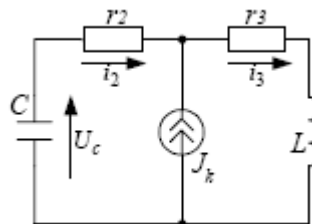


Рисунок 2.32 – Электрическая цепь при разомкнутом ключе

Уравнения для данной цепи имеют вид:

$$U_c + C \frac{dU_c}{dt} r_2 + i_3 r_3 + L \frac{di_3}{dt} = 0;$$

$$C \frac{dU_c}{dt} + J_k = i_3.$$



В установившемся режиме  $d/dt=0$ :

$$i_3(0) = J_k,$$

$$U_C(0) = -i_3 r_3 = -J_k r_3.$$

Таким образом, математическое описание схемы в нормальной форме Коши с начальными условиями будет записано:

$$\frac{dU_C}{dt} = (i_2 - i_1)/C, \quad U_C(0) = -J_k r_3,$$

$$\frac{di_3}{dt} = (E - i_1 r_1 - i_2 r_2 - i_3 r_3)/L, \quad i_3(0) = J_k.$$

Уравнение связи:

$$i_1 = (E + U_C)/r_1,$$

$$i_2 = i_3 - J_k.$$

Дифференциальные уравнения в операторной форме:

$$pU_C = (i_2 - i_1)/C,$$

$$pi_3 = (E - i_1 r_1 - i_2 r_2 - i_3 r_3)/L.$$

Структурная схема, составленная на основании данных уравнений, будет иметь вид, представленный на рисунке 2.33.

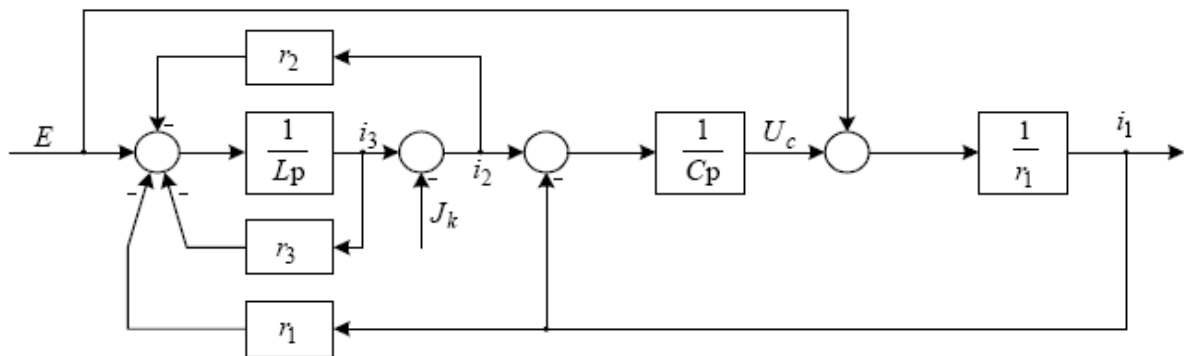


Рисунок 2.33 – Структурная схема

Структурная схема, составленная из блоков Simulink Matlab, представлена на рисунке 2.34.

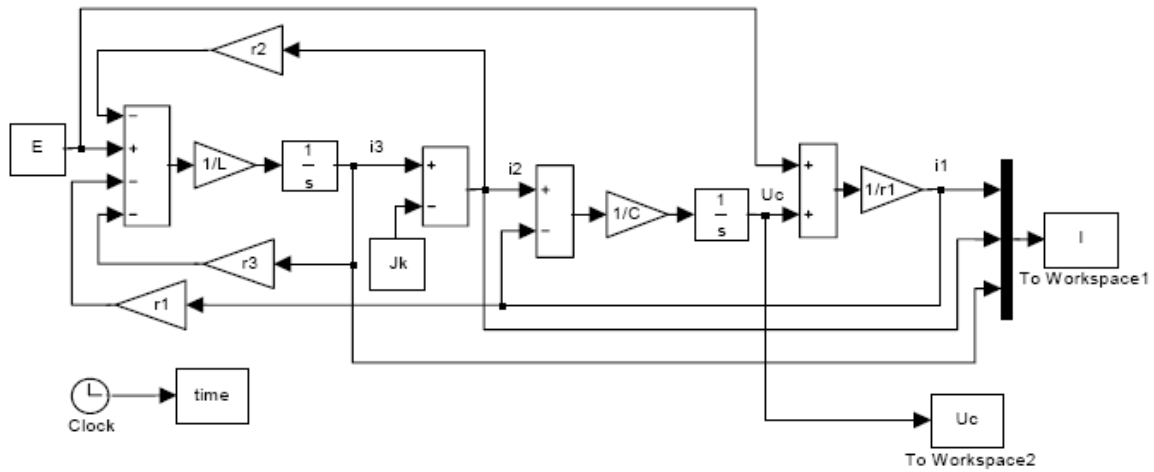


Рисунок 2.34 – Схема в среде пакета Matlab Simulink

### Индивидуальное задание

Рассчитать переходный процесс для электрических цепей (рисунок 2.35) с параметрами, указанными в таблице 2.22, для трех случаев источника постоянного напряжения ( $U=U_m$ ) и источника синусоидального напряжения ( $U=U_m \sin \omega t$ ). Построить графики напряжения на конденсаторе и токов в ветвях электрической цепи в функции времени.

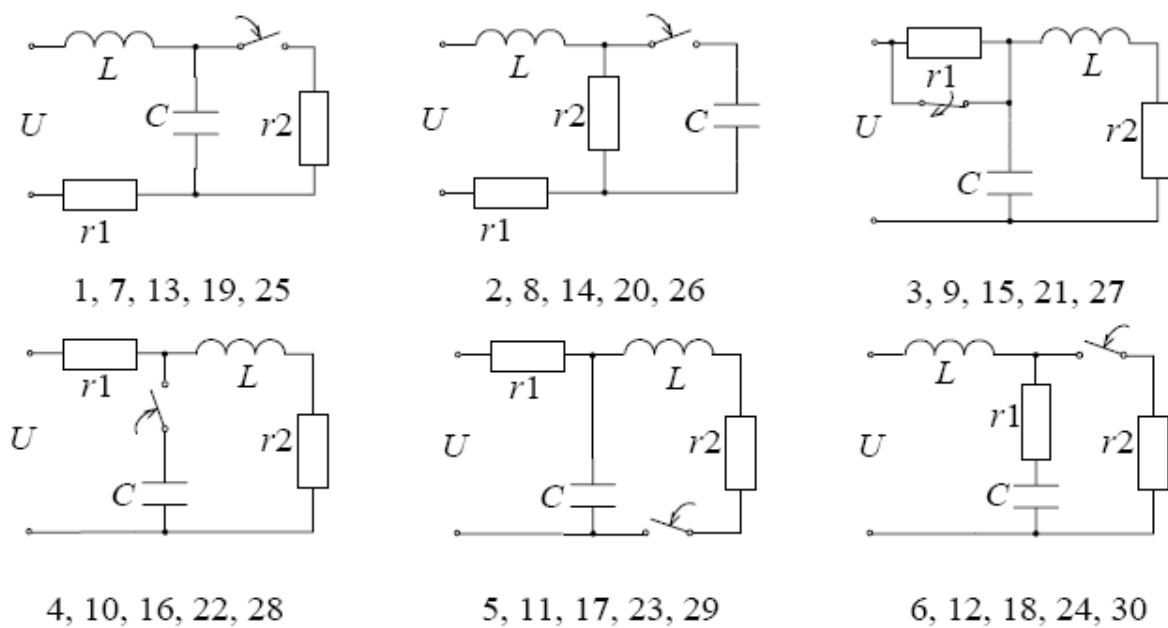


Рисунок 2.35 – Электрические схемы

Таблица 2.22 – Варианты индивидуальных заданий

№ вар.	$U_m, В$	$r1, Ом$	$r2, Ом$	$\omega, с^{-1}$	$L, Гн$	$C, мкФ$
1-6	200	50	100	125	0,1	10
7-12	100	100	120	100	0,2	5
13-18	300	60	120	75	0,15	15
19-24	400	150	100	150	0,25	2
25-30	500	100	200	100	0,05	10

## 2.7 Mathcad – универсальная вычислительная система

Mathcad – это популярная система компьютерной математики, предназначенная для автоматизации решения массовых математических задач в самых различных областях науки и техники. Название системы происходит от двух слов — MATHeMatica (математика) и CAD (Computer Aided Design — системы автоматического проектирования, или САПР), поэтому вполне правомерно считать Mathcad математическими САПР.

Сегодня различные версии Mathcad (рисунок 2.36) являются математически ориентированными универсальными системами.

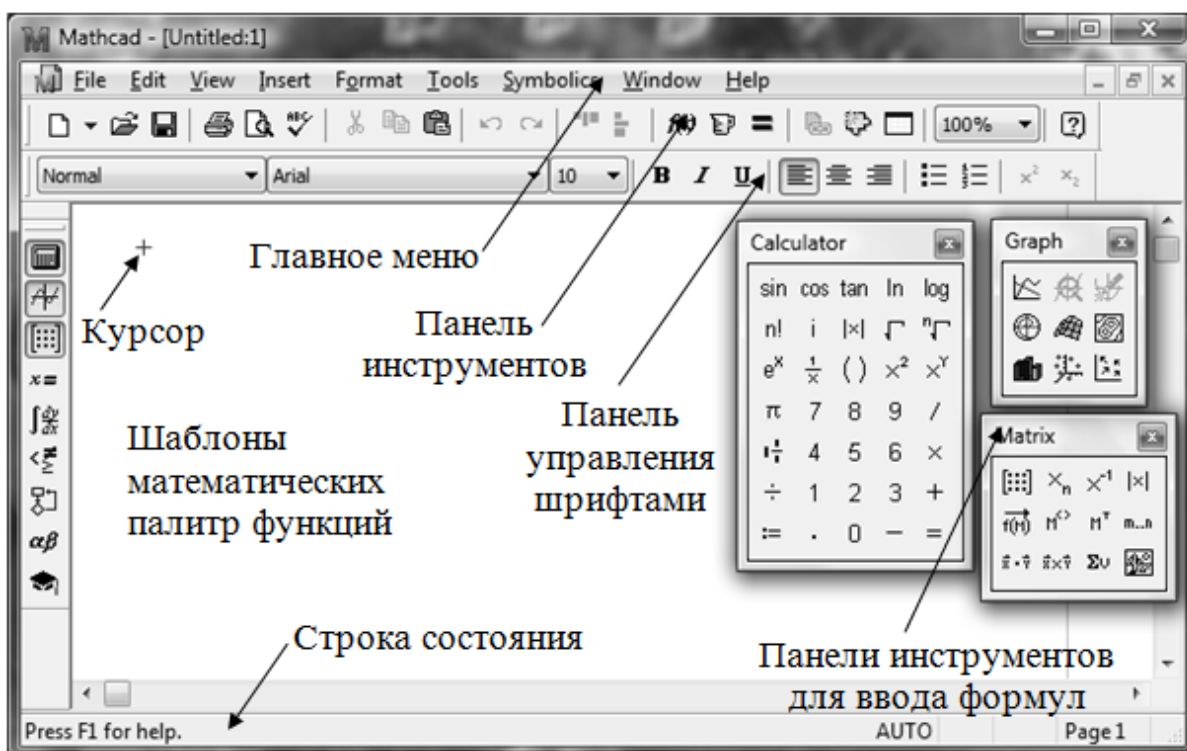


Рисунок 2.36 – Окно приложения Mathcad

Помимо собственно вычислений, как численных, так и аналитических, позволяет создавать интерактивные документы с вычислениями и визуальным сопровождением [18].

Mathcad имеет простой и интуитивный для использования интерфейс пользователя. Для ввода формул и данных можно использовать как клавиатуру, так и специальные панели инструментов.

Некоторые из математических возможностей Mathcad основаны на подмножестве системы компьютерной алгебры Maple (МКМ, Maple Kernel Mathsoft). Начиная с 14 версии – используется символьное ядро MuPAD.

Несмотря на то, что эта программа в основном ориентирована на пользователей-непрограммистов, Mathcad также используется в сложных проектах, чтобы визуализировать результаты математического моделирования, путем использования распределённых вычислений и традиционных языков программирования. Также Mathcad часто используется в крупных инженерных проектах, где большое значение имеет трассируемость и соответствие стандартам. Открытая архитектура приложения в сочетании с поддержкой технологий .NET и XML позволяют легко интегрировать Mathcad практически в любые ИТ-структуры и инженерные приложения.

Mathcad содержит сотни операторов и встроенных функций для решения различных технических задач. Программа позволяет выполнять численные и символьные вычисления, производить операции с скалярными величинами, векторами и матрицами, автоматически переводить одни единицы измерения в другие.

Среди возможностей Mathcad можно выделить: решение дифференциальных уравнений, в том числе и численными методами; построение двумерных и трёхмерных графиков функций (в разных системах координат, контурные, векторные и т. д.); использование греческого алфавита, как в уравнениях, так и в тексте; выполнение вычислений в символьном режиме; выполнение операций с векторами и матрицами; символьное решение систем уравнений; аппроксимация кривых; выполнение подпрограмм; поиск корней многочленов и функций; проведение статистических расчётов и работа с распределением вероятностей; поиск собственных чисел и векторов; вычисления с единицами измерения; интеграция с САПР системами, использование результатов вычислений в качестве управляющих параметров.

Документ программы Mathcad называется рабочим листом. Он содержит объекты: формулы и текстовые блоки. В ходе расчетов формулы обрабатываются последовательно, слева направо и сверху вниз, а текстовые блоки игнорируются. Ввод информации осуществляется в месте расположения курсора. Программа Mathcad использует три вида курсоров. Если ни один объект не выбран, используется крестообразный

курсор, определяющий место создания следующего объекта. При вводе формул используется уголкоый курсор, указывающий текущий элемент выражения. При вводе данных в текстовый блок применяется текстовый курсор в виде вертикальной черты.

Формулы – основные объекты рабочего листа. Новый объект по умолчанию является формулой. Чтобы начать ввод формулы, надо установить крестообразный курсор в нужное место и начать ввод букв, цифр, знаков операций. При этом создается область формулы, в которой появляется уголкоый курсор, охватывающий текущий элемент формулы, например имя переменной (функции) или число. При вводе бинарного оператора по другую сторону знака операции автоматически появляется заполнитель в виде черного прямоугольника. В это место вводят очередной операнд. Для управления порядком операций используют скобки, которые можно вводить вручную. Уголковый курсор позволяет автоматизировать такие действия. Чтобы выделить элементы формулы, которые в рамках операции должны рассматриваться как единое целое, используют клавишу «Пробел». При каждом ее нажатии уголкоый курсор «расширяется», охватывая элементы формулы, примыкающие к данному. После ввода знака операции элементы в пределах уголкового курсора автоматически заключаются в скобки. Элементы формул можно вводить с клавиатуры или с помощью специальных панелей управления.

Панели управления (рисунок 2.37) открывают с помощью меню View (Вид) или кнопками панели управления Math (Математика) [19].

Для ввода элементов формул предназначены следующие панели:

а) панель управления Arithmetic (Счет) для ввода чисел, знаков типичных математических операций и наиболее часто употребляемых стандартных функций;

б) панель управления Evaluation (Вычисление) для ввода операторов вычисления и знаков логических операций;

в) панель управления Graph (График) для построения графиков;

г) панель управления Matrix (Матрица) для ввода векторов и матриц и задания матричных операций;

д) панель управления Calculus (Исчисление) для задания операций, относящихся к математическому анализу;

е) панель управления Greek (Греческий алфавит) для ввода греческих букв (их можно также вводить с клавиатуры, если сразу после ввода соответствующего латинского символа нажимать сочетание клавиш CTRL+G, например [a][CTRL+G] –  $\alpha$ , [W][CTRL+G] –  $\Omega$ );

ж) панель управления Symbolic (Аналитические вычисления) для управления аналитическими преобразованиями.

Введенное выражение обычно вычисляют или присваивают переменной. Для вывода результата выражения используют знак вычисления, который выглядит как знак равенства и вводится при помощи кнопки Evaluate Expression (Вычислить выражение) на панели инструментов Evaluation (Вычисление).



Рисунок 2.37 – Панели инструментов программы Mathcad для ввода формул

Знак присваивания изображается как «:=», а вводится при помощи кнопки Assign Value (Присвоить значение) на панели инструментов Evaluation (Вычисление). Слева от знака присваивания указывают имя переменной. Оно может содержать латинские и греческие буквы, цифры, символы «'», «\_» и «∞», а также описательный индекс. Описательный индекс вводится с помощью символа «.» и изображается как нижний индекс, но является частью имени переменной, например  $V_{init}$ . «Настоящие» индексы, определяющие отдельный элемент вектора или матрицы, задаются по-другому.

Переменную, которой присвоено значение, можно использовать далее в документе в вычисляемых выражениях. Чтобы узнать значение переменной, следует использовать оператор вычисления.

Текст, помещенный в рабочий лист, содержит комментарии и описания и предназначен для ознакомления, а не для использования в расчетах. Программа Mathcad определяет назначение текущего блока автоматически при первом нажатии клавиши «Пробел». Если введенный текст не может быть интерпретирован как формула, блок преобразуется в текстовый и последующие данные рассматриваются как текст. Создать текстовый блок без использования автоматических средств позволяет команда Insert > Text Region (Вставка > Текстовый блок).

Иногда требуется встроить формулу внутрь текстового блока. Для этого служит команда Insert > Math Region (Вставка > Формула).

Для форматирования формул и текста (рисунок 2.9) в программе Mathcad используется панель инструментов Formatting (Форматирование).

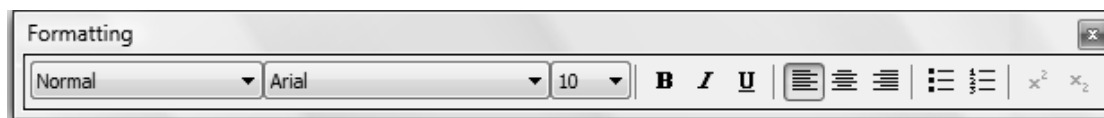


Рисунок 2.38 – Панели инструментов для форматирования данных

С ее помощью можно индивидуально отформатировать любую формулу или текстовый блок, задав гарнитуру и размер шрифта, а также полужирное, курсивное или подчеркнутое начертание символов. В текстовых блоках можно также задавать тип выравнивания и применять маркированные и нумерованные списки.

В качестве средств автоматизации используются стили оформления. Выбрать стиль оформления текстового блока или элемента формулы можно из списка Style (Стиль) на панели инструментов Formatting (Форматирование). Для формул и текстовых блоков применяются разные наборы стилей.

Чтобы изменить стиль оформления формулы или создать новый стиль, используется команда Formate Equation (Формат ^ Выражение). Изменение стандартных стилей Variables (Переменные) и Constants (Константы) влияет на отображение формул по всему документу. Стиль оформления имени переменной учитывается при ее определении. Так, переменные  $x$  и  $X$  рассматриваются как различные и не взаимозаменяемы. При оформлении текстовых блоков можно использовать более обширный набор стилей. Настройка стилей текстовых блоков производится при помощи команды Format > Style (Формат > Стиль).

Произвольные зависимости между входными и выходными параметрами задаются при помощи функций. Функции принимают набор параметров и возвращают значение, скалярное или векторное (матричное). В формулах можно использовать стандартные встроенные функции, а также функции, определенные пользователем.

Чтобы использовать функцию в выражении, надо определить значения входных параметров в скобках после имени функции. Имена простейших математических функций можно ввести с панели инструментов Arithmetic (Арифметические вычисления или Счет).

Информацию о других функциях можно почерпнуть в справочной системе. Вставить в выражение стандартную функцию (рисунок 2.10) можно при помощи команды Insert > Function (Вставка > Функция).

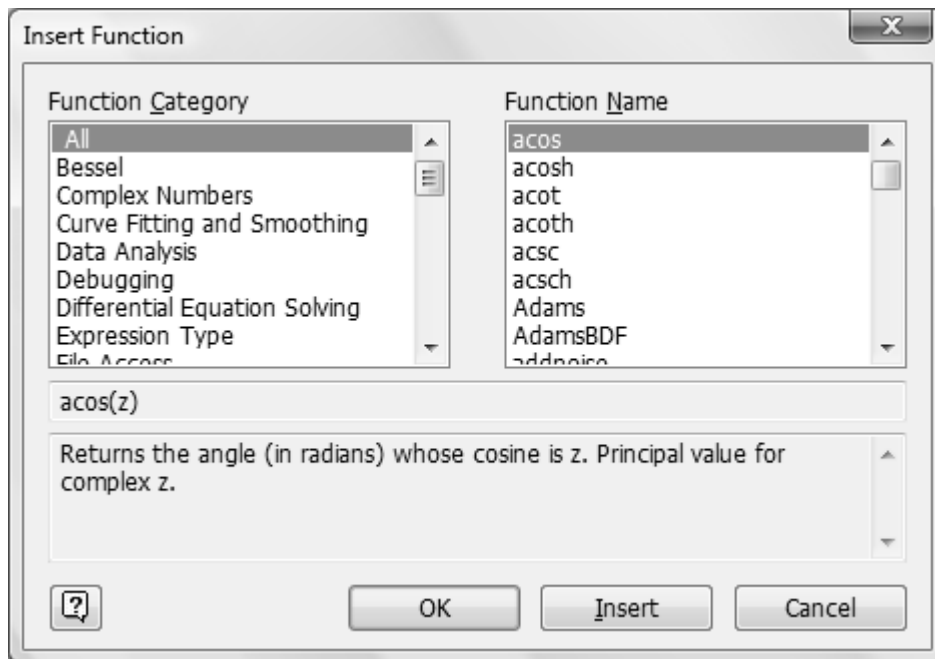


Рисунок 2.39 – Меню выбора predefined функций

В диалоговом окне Insert Function (Вставка функции) слева выбирается категория, к которой относится функция, а справа – конкретная функция. В нижней части окна выдается информация о выбранной функции. При вводе функции через это диалоговое окно автоматически добавляются скобки и заполнители для значений параметров.

Пользовательские функции должны быть сначала определены. Определение задается при помощи оператора присваивания. В левой части указывается имя пользовательской функции и, в скобках, формальные параметры — переменные, от которых она зависит. Справа от знака присваивания эти переменные должны использоваться в выражении. При использовании пользовательской функции в последующих формулах ее имя вводят вручную. В диалоговом окне Insert Function (Вставка функции) оно не отображается.

Далее приведено обозначение стандартных функций пакета Mathcad.

1. Тригонометрические и обратные функции:  
 $\sin(z)$ ,  $\cos(z)$ ,  $\tan(z)$ ,  $\operatorname{asin}(z)$ ,  $\operatorname{acos}(z)$ ,  $\operatorname{atan}(z)$ , где  $z$  – угол в радианах.
2. Гиперболические и обратные функции:  
 $\sinh(z)$ ,  $\cosh(z)$ ,  $\tanh(z)$ ,  $\operatorname{asinh}(z)$ ,  $\operatorname{acosh}(z)$ ,  $\operatorname{atanh}(z)$ .
3. Экспоненциальные и логарифмические:  
 $\exp(z) = e^z$ ;  
 $\ln(z)$  – натуральный логарифм;  
 $\log(z)$  – десятичный логарифм;
4. Статистические функции:  
 $\operatorname{mean}(x)$  – среднее значение;



$\text{var}(x)$  – дисперсия;  
 $\text{stdev}(x)$  – среднее квадратическое отклонение;  
 $\text{snorm}(x)$  – функция нормального распределения;  
 $\text{erf}(x)$  – функция ошибки.

5. Функции Бесселя:

$J_0(x)$ ,  $J_1(x)$ ,  $J_n(n,x)$  – функции Бесселя первого порядка;  
 $Y_0(x)$ ,  $Y_1(x)$ ,  $Y_n(n,x)$  – функции Бесселя второго порядка;

6. Функции комплексного переменного:

$\text{Re}(z)$  – вещественная часть комплексного числа;

$\text{Im}(z)$  – мнимая часть комплексного числа;

$\text{arg}(z)$  – аргумент комплексного числа.

7. Преобразование Фурье:

$U:=\text{fft}(V)$  – прямое преобразование ( $V$ – вещественное);

$V:=\text{ifft}(U)$  – обратное преобразование ( $V$ – вещественное) ;

$U:=\text{cfft}(V)$  – прямое преобразование ( $V$ – комплексное);

$V:=\text{icfft}(U)$  – обратное преобразование ( $V$ – комплексное).

8. Корреляционная функция – позволяет рассчитывать коэффициент корреляции двух векторов  $v_x$  и  $v_y$  и определить уравнение линейной регрессии:

$\text{corr}(v_x,v_y)$  – коэффициент корреляции;

$\text{slope}(v_x,v_y)$  – коэффициент наклона линии регрессии;

$\text{intercept}(v_x,v_y)$  – начальная координата линии регрессии.

9. Линейная интерполяция:

$\text{linterp}(v_x,v_y,x)$

$v_x,v_y$  – векторы значений аргумента и функций.  $x$ – значение аргумента, для которого проводится интерполяция.

10. Функция для определения корней алгебраических и трансцендентных уравнений:

$\text{root}(\text{уравнения}, \text{переменная})$  – значение переменной, когда уравнение равно нулю.

11. Датчик случайных чисел:

$\text{rnd}(x)$  – случайное число с равномерным распределением от 0 до  $x$ .

12. Целая часть переменной:

$\text{floor}(x)$  – ближайшее наименьшее целое число;

$\text{ceil}(x)$  – ближайшее наибольшее целое число.

13. Выделение остатка:

$\text{mod}(x,y)$  – остаток от деления  $x$  на  $y$ ;

14. Остановка итерации:

$\text{until}(x,y)$  – когда  $x < 0$ .

15. Функция условного перехода:

$\text{if}(\text{условие}, x, y)$  – если условие выполняется, то функция равняется  $x$ , иначе  $y$ .

16. Единичная функция (функция Хевисайда):

$\Phi(x)$  – если  $x > 0$ . То функция равна 1, иначе 0.

17. Логические выражения и операции. Простейшими видами

логических выражений являются следующие: логическая константа, логическая константа, логическая константа, логическая переменная, выражение отношения. Например, при  $x:=0.5$  операции отношения присваивают L истину или ложь (1 или 0):

$L := x \leq 1 \quad L=0;$

$L := x \geq 1 \quad L=0;$

$L := x \approx 1 \quad L=0;$

$L := x < 1 \quad L=1;$

$L := x > 1 \quad L=0.$

18.Функции, определяемые пользователем. Пользователь может самостоятельно определить необходимые ему функции, отсутствующие среди встроенных функций пакета.

На сегодняшний день популярностью пользуется версия Mathcad 14, в которой реализована возможность двусторонней интеграции с основным продуктом РТС — пакетом Pro/ENGINEER. Базовые величины, рассчитанные в системе Mathcad, могут быть переведены в параметры и размеры CAD-модели для управления геометрическим объектом. Параметры из модели Pro/ENGINEER также можно ввести в Mathcad для последующих инженерно-конструкторских расчетов. Существенные изменения коснулись также математического ядра системы, которое теперь использует символьную систему MuPAD. Это позволило во многих случаях повысить точность символьных вычислений и их детализацию, но есть и негативные последствия, связанные, в первую очередь, с совместимостью символьных алгоритмов с предыдущими версиями.

Основные нововведения заключаются в следующем: введено полную поддержку шрифтов Unicode, поддерживается проверка орфографии на 15-ти языках, добавлена возможность выбора формата отображения чисел на шкалах координатных осей, инструментарий решения дифференциальных уравнений дополнена тремя новыми алгоритмами,

А одна из последних версий – Mathcad Prime 1.0 – является последним релизом компании РТС в программном обеспечении для инженерных расчетов.

Основные отличия нового Mathcad Prime 1.0: изменён интерфейс пользователя, который теперь выполнен в стиле последних версий MS Office; среда вычислений, ориентированная на работу с документами, позволяет пользователям быстро и просто создавать детализированные технические документы, которые включают сложные вычисления, используя «живые» математические примечания с текстом, изображениями и диаграммами; полный пакет дополнительных функций численной математики позволяет быстро и качественно решить любую вычислительную задачу, экономя время и средства; интуитивно понятный редактор уравнений, работающий в режиме полного соответствия,

позволяет пользователям описывать условия и решения в естественном математическом виде.

Mathcad Prime 1.0 написан на совершенно новой программной основе, имеет максимальную производительность, а также полную поддержку Windows 7 и последнего релиза MS Excel.

Таким образом, Mathcad, являясь мировым стандартом для инженерных вычислений, обеспечивает уникальную, высокоинтуитивную инженерную среду. Эта среда позволяет разработчикам быстро осуществлять важные инженерные вычисления, в том числе и разработку технического задания, анализ исходных данных, выбор методов решения, уравнений, обоснование допущений, а также обмениваться этими вычислениями.

В отличие от классического инструмента программирования или электронных таблиц интерфейс Mathcad принимает от пользователя через клавиатуру или мышь и отображает на дисплее естественную математическую нотацию, понятную даже людям, далеким от компьютеров. Так как рабочие листы содержат вычисления в реальном времени, то одно единственное нажатие клавиши, изменяющее входные данные или вид уравнения, немедленно возвращает обновленный результат. В случае изменения значения переменной происходит немедленный пересчет результата и обновление двухмерных или трехмерных графиков, что избавляет от необходимости пересчета вручную. Результаты вычислений сохраняются в рабочих листах, которые могут использоваться повторно и легко преобразуются в целый ряд форматов, включая MS Word, PDF, HTML и XML. Эти гибкие форматы позволяют инженерам совместно использовать полностью документированный проект – включая концепцию и реализацию, а не просто некий программный код. Формат XML и поддержка стандартных интерфейсов облегчают совместное использование рабочих листов, методов или числовых значений с другими пользователями и системами, включая управление документооборотом, автоматизированным проектированием (CAD) и решениями по управлению сведениями об изделии (PDM).

### 2.2.1 Задание на лабораторную работу № 24

**Тема:** Пакет Mathcad – система выполнения инженерных расчетов. Работа с текстовым и символьным редактором.

**Цель работы:** изучить приемы работы в среде пакета Mathcad, изучить правила задания данных, приобрести навыки работы с формулами и текстом и особенностями символьных преобразований.

## *Теоретические сведения*

В основном, работа с системой MathCAD сводится к определению задания на вычисления, т.е. текстовая постановка задачи и определение формул для проведения вычислений. Текстовые и формульные данные определяются в виде блоков.

Текстовый редактор позволяет задавать текстовые комментарии. Они делают документ с формулами и графиками более понятным.

Для запуска формульного редактора необходимо установить курсор мыши в любом свободном месте окна редактирования и щелкнуть левой клавишей. Появится визир в виде маленького красного крестика. Визир указывает место, с которого начинается набор формул – вычислительных блоков. В области формул визир превращается в синий уголок, указывающий направление и место ввода [19].

Для облегчения и удобства создания вычислительных блоков в Mathcad представлена математическая панель инструментов с шаблонами различных математических символов.

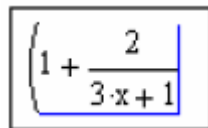
### *Ход работы*

1 Определить постановку задачи в текстовом редакторе:

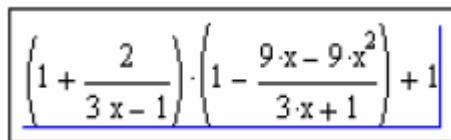
а) в меню Insert выбрать команду Text Region или щелкнуть по клавише “ и в открывшейся текстовой области набрать соответствующий текст;

б) для выхода в формульный редактор щелкнуть мышкой вне текстовой области.

2 Ввести выражение – где при вводе дробей или скобок необходимо соблюдать размещение синего уголка, т.к. при неправильном вводе результат будет ошибочным или вообще не будет вычисления. Допустим, при закрытии скобок уголок должен быть размещении так, как показано ниже.


$$\left(1 + \frac{2}{3x+1}\right)$$

3 Выделить выражение или разместить синий уголок так, чтобы он охватывал все выражение.


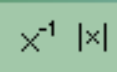
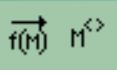
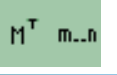
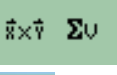






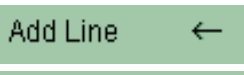
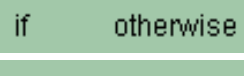

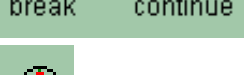











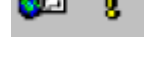




$$\left(1 + \frac{2}{3x-1}\right) \cdot \left(1 - \frac{9x-9x^2}{3x+1}\right) + 1$$

4 В меню Symbolics выбрать команду Simplify.

5 Полученный результат должен появиться слева от выражения или снизу – в зависимости от выбранного режима (меню Symbolics команда Evaluation Style).

## Индивидуальное задание

1 Выполнить согласно варианту описание следующего элемента пакета Mathcad:

<p>1 </p> <p>2 </p> <p>3 </p> <p>4 </p> <p>5 </p> <p>6 </p> <p>7 </p> <p>8 </p> <p>9 </p> <p>10 </p>	<p>11 </p> <p>12 </p> <p>13 </p> <p>14 </p> <p>15 </p> <p>16 </p> <p>17 </p> <p>18 </p> <p>19 </p> <p>20 </p> <p>21 </p>	<p>22 </p> <p>23 </p> <p>24 </p> <p>25 </p> <p>26 </p> <p>27 </p> <p>28 </p> <p>29 </p> <p>30 </p>
---	--	--

2 В текстовом редакторе вывести комментарии к решению задачи. Набрать соответствующее выражение и выполнить символьные преобразования алгебраических выражений (таблица 2.23).

Таблица 2.23 – Варианты индивидуальных заданий

№	Задание	Выражение
1 16	Упростить	$\left(1 - \frac{3 \cdot x - 3 \cdot x^2}{9 \cdot x + 1}\right) + 1 + \left(3 - \frac{2 \cdot x}{3 \cdot x}\right)$
2 17	Привести подобные	$x \cdot (z + 2)^2 - 4 \cdot z \cdot (x + 2 \cdot z)$
3 18	Разложить на множители	$a^2 \cdot b + a \cdot b^2 + 2 \cdot a \cdot b \cdot c + b^2 \cdot c + a^2 \cdot c + a \cdot c^2 + b \cdot c^2$
4 19	Упростить	$\frac{a^3 - b^3}{a - b}$
5 20	Привести подобные	$x \cdot (z + 1)^2 - 2 \cdot z \cdot (x + z)$

Продолжение таблицы 2.23.

6 21	Разложить на множители	$\frac{x^2 - 3x + 7}{(x-1)^2 \cdot (x^2 + x + 1)}$
7 22	Привести подобные	$\left[ x \cdot (z+2)^2 - 4 \cdot z \cdot (x+2 \cdot z) \right] + 4 \cdot x \cdot (x+4)^2$
8 23	Разложить на множители	$\frac{(x^2 - 6x + 2) + (2 + x)}{(x-1)^2 \cdot (x^2 + x + 1)}$
9 24	Упростить	$\left( 1 - \frac{x - 3x^2}{x+1} \right) + 1 + \left( 3 - \frac{x}{3x} \right)$
10 25	Привести подобные	$x \cdot (z+1)^2 - 2 \cdot z \cdot (x+z)^2$
11 26	Разложить на множители	$a^2 \cdot b + a \cdot b^2 + 2 \cdot a \cdot b \cdot c + b^2 \cdot c$
12 27	Привести подобные	$\frac{(x^2 - x + 2) + (2 + x)}{(x-1)^2 \cdot (x^2 + x + 1)}$
13 28	Разложить на множители	$a^2 \cdot c + a \cdot c^2 + (b \cdot c^2 + c) + c^2 \cdot a$
14 29	Упростить	$\frac{a^2 - b^2}{a - b}$
15 30	Разложить на множители	$a \cdot b + a \cdot b^2 + 2 \cdot a \cdot b \cdot c$

### 2.2.2 Задание на лабораторную работу № 25

**Тема:** Решение систем линейных уравнений в пакете Mathcad.

**Цель работы:** изучить простые и итерационные методы решения систем уравнений, приобрести навыки решения уравнений разными методами.

#### *Теоретические сведения*

Методы решения систем линейных уравнений ( $AX=B$ ) можно разделить на две основные группы: прямые методы и итерационные. Прямые методы дают точное решение за конечное число операций; к ним относятся, например, методы Крамера, Гаусса, Зейделя. Итерационные методы дают решение системы уравнений как предел последовательных

приближений. Для итерационных методов необходимо выполнение условий сходимости и дополнительных преобразований системы в эквивалентную ей.

Контроль выполняемых вычислений является важным элементом решения любой вычислительной задачи. Для контроля прямого хода пользуются контрольными суммами, которые представляют собой суммы коэффициентов при неизвестных и свободного члена для каждого уравнения заданной системы. Для контроля вычислений в основной части схемы единственного деления (столбцы коэффициентов при неизвестных и свободных членов) над контрольными суммами выполняют те же действия, что и над остальными элементами той же строки. При отсутствии вычислительных ошибок контрольная сумма для каждой строки в пределах влияния погрешностей округления и их накопления должна совпадать со строчной суммой – вторым столбцом контроля. Строчные суммы представляют собой суммы всех элементов из основной части этой строки.

Методом простой итерации решается система уравнений в виде матрицы, которая является квадратной и невырожденной.

Метод Зейделя отличается от метода простой итерации тем, что найдя какое-то значение для компоненты, мы на следующем шаге используем его для отыскания следующей компоненты.

Для решения системы линейных алгебраических уравнений как матричного уравнения  $AX=B$  (рисунок 2.40) необходимо выполнить следующие действия:

- а) ввести матрицу системы и матрицу-столбец правых частей;
- б) вычислить решение системы по формуле  $X=A^{-1}B$ ;
- в) проверить правильность решения умножением матрицы системы на вектор-столбец решения;
- г) найти решение системы с помощью функции `lsolve` и сравнить результаты.

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \\ 1 & 1 & 1 \end{bmatrix} \quad b := \begin{bmatrix} 7 \\ 5 \\ 3 \end{bmatrix}$$

$$x := A^{-1} \cdot b \quad x = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \quad A \cdot x - b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

*Рисунок 2.40 – Фрагмент решения системы уравнений матричным методом*

Решение линейной системы методом Гаусса предполагает применение функции  $\text{augment}(A,b)$ , которая формирует расширенную матрицу системы добавлением к матрице системы справа столбца правых частей. Функция  $\text{rref}$  приводит расширенную матрицу системы к ступенчатому виду, выполняя прямой и обратный ходы гауссова исключения. Последний столбец содержит решение системы.

Решение системы методом Крамера (рисунок 2.41) осуществляется по следующим пунктам (на примере системы уравнений с тремя неизвестными):

- а) вычисление главного определителя  $D$  матрицы  $A$ ;
- б) задание матрицы  $DX1$  с заменой первого столбца матрицы  $A$ , матрицей  $B$ ;
- в) вычисление определителя матрицы  $DX1$ ;
- г) задание матрицы  $DX2$  с заменой второго столбца матрицы  $A$  матрицей  $B$ ;
- д) вычисление определителя матрицы  $DX$ ;
- е) задание матрицы  $DX3$  с заменой третьего столбца матрицы  $A$  матрицей  $B$ ;
- ж) вычисление определителя матрицы  $DX3$ ;
- з) определение решения системы линейных уравнений  $x_1, x_2, x_3$ .

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \\ 1 & 1 & 1 \end{bmatrix} \quad b := \begin{bmatrix} 7 \\ 5 \\ 3 \end{bmatrix}$$

$$D := |A| \quad D = 9$$

$$DX1 := \begin{bmatrix} 7 & 2 & 3 \\ 5 & -3 & 2 \\ 3 & 1 & 1 \end{bmatrix} \quad DX1 := |DX1| \quad DX1 = 9$$

$$DX2 := \begin{bmatrix} 1 & 7 & 3 \\ 1 & 5 & 2 \\ 1 & 3 & 1 \end{bmatrix} \quad DX2 := |DX2| \quad DX2 = 0$$

$$DX3 := \begin{bmatrix} 1 & 2 & 7 \\ 1 & -3 & 5 \\ 1 & 1 & 3 \end{bmatrix} \quad DX3 := |DX3| \quad DX3 = 18$$

$$x_1 := \frac{DX1}{D} \quad x_1 = 1 \quad x_2 := \frac{DX2}{D} \quad x_2 = 0 \quad x_3 := \frac{DX3}{D} \quad x_3 = 2$$

Рисунок 2.41 – Фрагмент решения системы уравнений методом Крамера



Решение системы линейных алгебраических уравнений методом Зейделя:

- а) ввод матрицы C и d (аналог матриц A и B);
- б) преобразование системы  $Cx=d$  к виду  $x=b+A_1x+A_2x$ .
- в) определение нулевого приближения решения;
- г) задание количества итераций;
- д) вычисление последовательных приближений.

Алгоритмы метода простых итераций и метода Зейделя представлены в рабочем пространстве Mathcad на рисунках 2.42 и 2.43 соответственно.

ORIGIN := 1

$$C := \begin{bmatrix} 100 & 6 & -2 \\ 6 & 200 & -10 \\ 1 & 2 & 100 \end{bmatrix} \quad d := \begin{bmatrix} 200 \\ 600 \\ 500 \end{bmatrix}$$

$$i := 1..3 \quad j := 1..3$$

$$b_i := \frac{d_i}{C_{i,i}} \quad A_{i,j} := \frac{-C_{i,j}}{C_{i,i}} \quad A_{i,i} := 0$$

$$A = \begin{bmatrix} 0 & -0.06 & 0.02 \\ -0.03 & 0 & 0.05 \\ -0.01 & -0.02 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

$$x^{<1>} := b \quad k := 2..10 \quad x^{<k>} := b + A \cdot x^{<k-1>}$$

	1	2	3	4	5	6	7	8	9	10
x =	1	2	1.92	1.907	1.907	1.907	1.907	1.907	1.907	1.907
	2	3	3.19	3.188	3.189	3.189	3.189	3.189	3.189	3.189
	3	5	4.92	4.917	4.917	4.917	4.917	4.917	4.917	4.917

$$X := x^{<10>} \quad X = \begin{bmatrix} 1.907 \\ 3.189 \\ 4.917 \end{bmatrix}$$

Рисунок 2.43 – Метод простых итераций

ORIGIN := 1

$$C := \begin{bmatrix} 100 & 6 & -2 \\ 6 & 200 & -10 \\ 1 & 2 & 100 \end{bmatrix} \quad d := \begin{bmatrix} 200 \\ 600 \\ 500 \end{bmatrix}$$

$$i := 1..3 \quad b_i := \frac{d_i}{C_{i,i}} \quad i := 2..3 \quad j := 1..2$$

$$A1_{i,j} := \frac{-C_{i,j}}{C_{i,i}} \quad A2_{j,i} := \frac{-C_{j,i}}{C_{j,j}}$$

$$A1_{i,i} := 0 \quad A1_{j,i} := 0 \quad A2_{i,i} := 0 \quad A2_{i,j} := 0 \quad A := A1 + A2$$

$$A1 = \begin{bmatrix} 0 & 0 & 0 \\ -0.03 & 0 & 0 \\ -0.01 & -0.02 & 0 \end{bmatrix} \quad A2 = \begin{bmatrix} 0 & -0.06 & 0.02 \\ 0 & 0 & 0.05 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & -0.06 & 0.02 \\ -0.03 & 0 & 0.05 \\ -0.01 & -0.02 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

$$x^{<1>} := b \quad y^{<1>} := b \quad k := 2..10$$

$$x^{<k>} := b + A2 \cdot x^{<k-1>} \quad x^{<k>} := x^{<k>} + A1 \cdot x^{<k-1>}$$

	1	2	3	4	5	6	7	8	9	10
x =	1	2	1.92	1.905	1.905	1.905	1.905	1.905	1.905	1.905
	2	3	3.19	3.192	3.193	3.193	3.193	3.193	3.193	3.193
	3	5	4.92	4.917	4.917	4.917	4.917	4.917	4.917	4.917

Рисунок 2.44 – Метод Зейделя

### Индивидуальное задание

Решить систему уравнений согласно варианту (таблица 2.8) символьным и матричным методами, проверить сходимость результатов решения методами Гаусса, Зейделя и простым итерационным методом. Сделать выводы об эффективности каждого метода.

### Контрольные вопросы

1 К какому типу - прямому или итерационному - относится метод Гаусса?

2 В чем заключается прямой и обратный ход в схеме единственного деления?

3 Как организуется, контроль над вычислениями в прямом и обратном ходе?

4 Как строится итерационная последовательность для нахождения решения системы линейных уравнений?

5 Как формулируются достаточные условия сходимости итерационного процесса?

6 Как эти условия связаны с выбором метрики пространства?

7 В чем отличие итерационного процесса метода Зейделя от аналогичного процесса метода простой итерации?

### 2.2.3 Задание на лабораторную работу № 26

**Тема:** Численное решение нелинейных уравнений.

**Цель работы:** изучить методы и приобрести навыки работы с функциями для решения нелинейных уравнений.

#### *Теоретические сведения*

Многие уравнения, например трансцендентные, не имеют аналитических решений. Однако они могут решаться численными методами с заданной погрешностью.

Для простейших уравнений вида  $F(x)=0$  решение находится с помощью функции `root` (выражение, имя переменной). Эта функция возвращает значение переменной с указанным уровнем, при котором выражение дает 0. Функция реализует вычисления итерационным методом, причем можно задать начальное значение переменной. Это особенно полезно, если возможно несколько решений.

Рассмотрим уравнение  $f(x) = 0$ , где  $f(x)$  – определена и непрерывна на некотором конечном или бесконечном интервале  $b < x < a$ . Всякое значение  $x^*$ , обращающее функцию  $f(x)$  в нуль,  $f(x^*)=0$  – называется корнем уравнения, а способ нахождения этого значения  $x^*$  и есть решение уравнения. Найти корни уравнения вида точно удастся лишь в редких случаях. Кроме того, часто уравнение содержит коэффициенты известные лишь приблизительно и, следовательно, сама задача о точном определении корней уравнения теряет смысл [17].

Разработаны методы численного решения нелинейных уравнений, позволяющие отыскать приближенные значения корней этого уравнения.

При этом приходится решать две задачи:

1) отделение корней, т. е. отыскание достаточно малых областей, в каждой из которых заключен только один корень уравнения;

2) вычисление корней с заданной точностью.

Воспользуемся известным результатом математического анализа: если непрерывная функция принимает на концах некоторого интервала значения разных знаков, то интервал содержит, по крайней мере, один корень уравнения.

Для выделения областей, содержащих один корень, можно использовать, например, графический способ, либо двигаясь вдоль области определения с некоторым шагом, проверять на концах интервалов условие смены знака функции [20].

Для решения второй задачи существует многочисленные методы, из которых следует выделить четыре: метод итераций, метод половинного деления, метод хорд, метод касательных.

### *Ход работы:*

Расчетная формула для метода хорд [17]:

$$x_{n+1} = \frac{x_0 f(x_n) - x_n f(x_0)}{(f(x_n) - f(x_0))}$$

для метода касательных:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

где значение  $x_0$  для метода хорд и начальная точка для метода касательных выбирается из условия выполнения неравенства:

$$f(x_0)f''(x_0) > 0.$$

В результате вычислений по этим формулам может быть получена последовательность приближенных значений корня  $x_1, x_2, \dots, x_{n-1}, x_n$ . Процесс вычислений заканчивается при выполнении условия  $|x_n - x_{n-1}| < \varepsilon$  ( $\varepsilon = 10^{-5}$ ).

Например, для уравнения вида  $x - \sin(x) - 0,25 = 0$ . На рисунке 2.45 представлен фрагмент графического решения уравнения, на рисунке 2.46 – операторный и символьный метод решения уравнения, на рисунке 2.47 – метод итераций, касательных и хорд.

Вычисление корней кубического полинома предполагает, что кубическое уравнение обязательно имеет хотя бы один кубический корень  $x_1$ , который определяется с помощью функции `root` (фрагмент расчета приведен на рисунке 2.48).

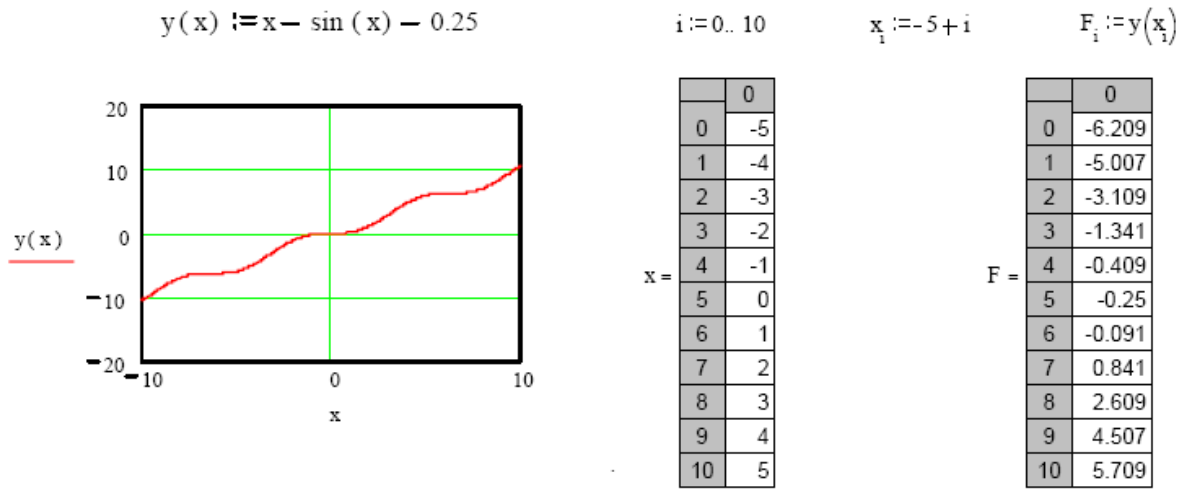


Рисунок 2.45 – Графическое решение уравнения

Given

$$x - \sin(x) - 0.25 = 0$$

$$\text{Find}(x) \rightarrow 1.17122965$$

$$x - \sin(x) - 0.25 \text{ solve, } x \rightarrow 1.17122965.$$

Рисунок 2.46 – Операторный и символьный методы решения

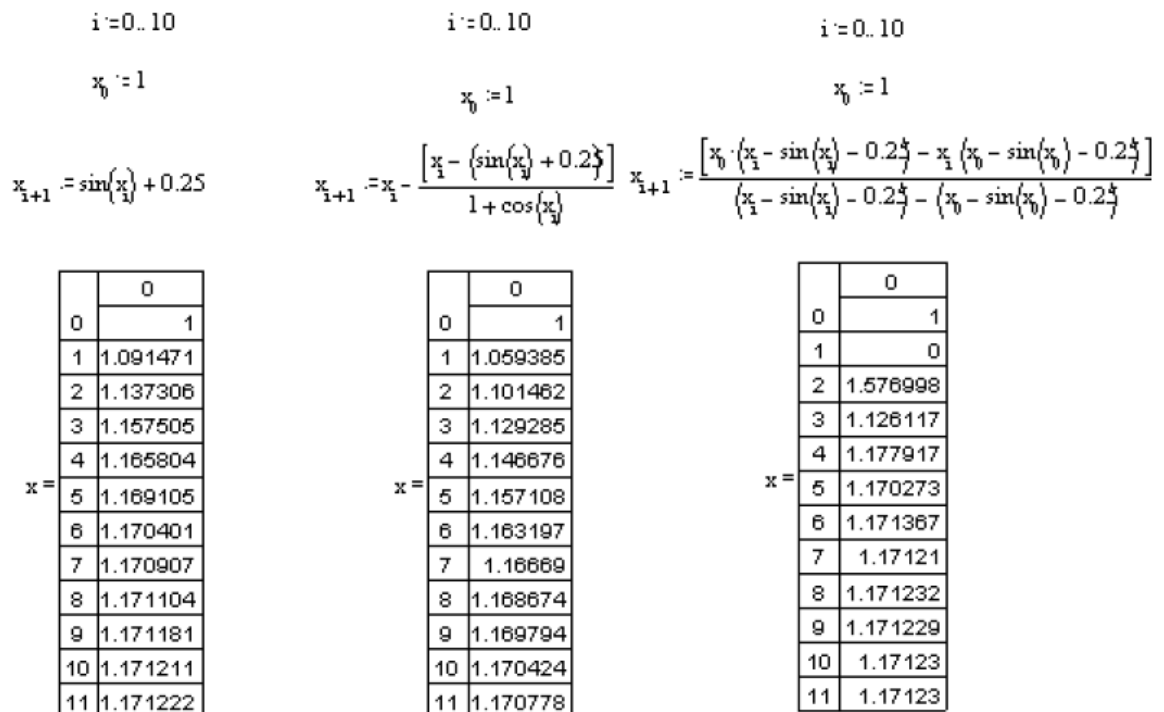


Рисунок 2.47 – Численные методы решения уравнения

$$a_3 := 1 \quad a_2 := -6 \quad a_1 := 21 \quad a_0 := -52$$

$$F(x) := a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0$$

$$x := 0 \quad x_1 := \text{root}(F(x), x) \quad x_1 = 4$$

$$i := \sqrt{-1} \quad x := 1 + 1 \cdot i \quad x_2 := \text{root}\left(\frac{F(x)}{x - x_1}, x\right) \quad x_2 = 1 + 3.464i$$

$$x_3 := \text{root}\left[\frac{F(x)}{(x - x_1) \cdot (x - x_2)}, x\right] \quad x_3 = 1 - 3.464i$$

Рисунок 2.48 – Фрагмент поиска корней функцией root

### Индивидуальное задание

1 Решить уравнение согласно варианту задания (таблица 2.24). Сделать отделение корней уравнения графически и по программе с заданной точностью ( $\varepsilon = 10^{-1}$ ). Провести уточнение корней методом хорд (X) и методом касательных (К) с точностью  $\varepsilon = 10^{-4}$ .

2 Вычислить корни и построить график кубического полинома вида  $f(x) = a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0$  (таблица 2.25).

Таблица 2.24 – Варианты индивидуальных заданий

№	М	№	М	Уравнение
1	X	16	К	$x + x \ln(x + 0.5) - 0.5 = 0$
2	К	17	X	$x^{2^x} - 1 = 0$
3	X	18	К	$x^3 - 2x^2 + x - 3 = 0$
4	К	19	X	$x^3 + 12x - 2 = 0$
5	X	20	К	$5x - 8 \ln(x) - 8 = 0$
6	К	21	X	$x^4 + 0.5x^3 - 4x^2 - 3x - 0.5 = 0$
7	X	22	К	$x + \sin(x) + 0.25 = 0$
8	К	23	X	$x^3 - 6x^2 + 20 = 0$
9	X	24	К	$5x^3 + 10x^2 + 5x - 1 = 0$
10	К	25	X	$0.1x^2 - x \ln(x) = 0$
11	X	26	К	$3x^4 - 4x^3 + 12x^2 + 1 = 0$
12	К	27	X	$6x^5 + 2x^2 \ln x - 7 = 0$
13	X	28	К	$30x^6 - 12x + \sin 3x = 0$
14	К	29	X	$2x^4 + 3x^3 + 2x^2 + 15 = 0$
15	X	30	К	$x^5 - 5x^3 - 10x^2 + x + 16 = 0$

Таблица 2.25 – Варианты индивидуальных заданий

№	$a_3$	$a_2$	$a_1$	$a_0$	№	$a_3$	$a_2$	$a_1$	$a_0$
1	2	-8	15	-30	16	-2	8	5	30
2	34	-3	23	48	17	4	3	2	8
3	5	-4	27	23	18	5	4	7	3
4	-34	67	-3	11	19	-3	6	-3	11
5	4	4	45	22	20	-4	4	5	2
6	6	12	12	35	21	6	1	2	5
7	-8	-24	26	-47	22	8	-4	6	-4
8	-10	-1	84	-2	23	10	1	8	-2
9	5	32	-34	8	24	5	3	4	8
10	7	17	25	48	25	-7	1	5	4
11	4	2	8	9	26	2	10	4	-5
12	5	-9	6	7	27	6	-4	2	13
13	3	8	-6	8	28	2	9	6	14
14	2	15	4	5	29	5	10	11	-7
15	1	14	-3	5	30	3	-4	8	9

#### 2.2.4 Задание на лабораторную работу № 27

**Тема:** Численное решение систем нелинейных уравнений.

**Цель работы:**

#### *Теоретические сведения*

При решении систем нелинейных уравнений используется специальный вычислительный блок, открываемый служебным словом – директивой Given – и имеющий следующую структуру:

Given

Уравнения

Ограничительные условия.

Также для решения применяются функции find и minerr, между которыми существуют принципиальные различия. Первая функция используется, когда решение реально существует. Вторая функция пытается найти максимальное приближение даже к несуществующему решению путем минимизации среднеквадратичной погрешности решения.

Find( $v_1, v_2, \dots, v_n$ ) – возвращает значение одной или ряда переменных для точного решения;

Minerr( $v_1, v_2, \dots, v_n$ ) – возвращает значение одной или ряда переменных для приближенного решения.

Ниже приведен пример решения уравнений вида  $x^2=3$  с помощью выше приведенных функций (рисунок 2.49).

Given

$$x^2 = 3$$

$$x0 := \text{Find}(x) \quad x0 = 1.732$$

Given

$$x^2 = 3$$

$$x1 := \text{Minerr}(x) \quad x1 = 1.732$$

Рисунок 2.49 – Фрагмент поиска решения уравнения

Также применяется графический метод решения, например, необходимо определить точки пересечения прямой  $8 + 3x$  с параболой  $x^2$ . С помощью графических функций в декартовой системе координат строим графики указанных функций (рисунок 2.50), где с помощью трассировки определяем координаты точек пересечения и проверяем решение путем подстановки искомых значений в функции (рисунок 2.51).

$$x := -6, -5.75 .. 6$$

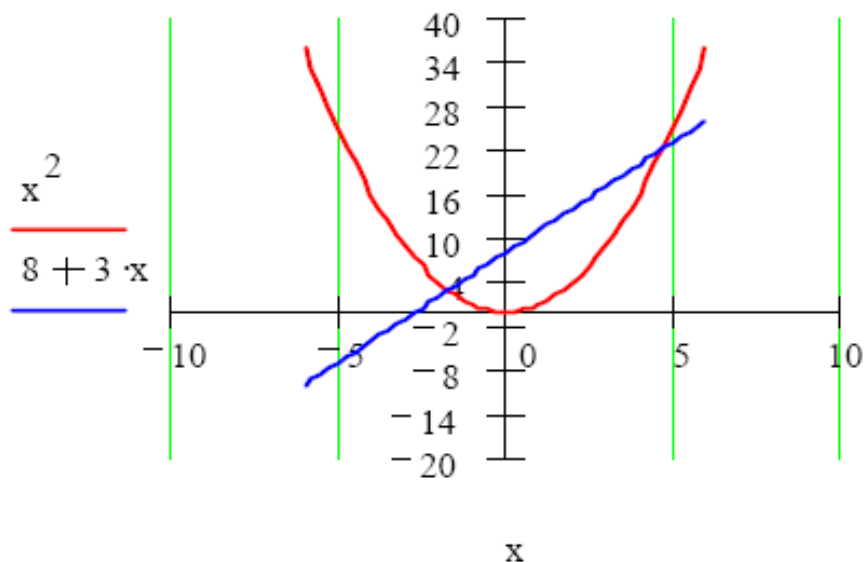


Рисунок 2.50 – Пересечение прямой с параболой



Ищем первое решение при  $x := 0$   $y := 0$

Given

$y = x^2$   $y = 8 + 3 \cdot x$  Решаемая система уравнений при ограничении  $x < 0$

$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} := \text{Find}(x, y)$   $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} -1.702 \\ 2.895 \end{bmatrix}$  Найдено первое решение

$x_0^2 = 2.895$   $8 + 3 \cdot x_0 = 2.895$  Проверка решения вычислением  $y$

Найдем второе решение при  $x > 0$

$x := 3$   $y := 0$  Given  $y = x^2$   $y = 8 + 3 \cdot x$

$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} := \text{Find}(x, y)$   $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 4.702 \\ 22.105 \end{bmatrix}$  Найдено второе решение

Рисунок 2.51 – Проверка решения нелинейного уравнения

### Индивидуальное задание

Вычислить координаты точек пересечения прямой с кривой второго порядка по варианту (таблица 2.26). Построить график.

Решить систему нелинейных уравнений (таблица 2.27). Найти приближенное решение. Построить график.

Таблица 2.26 – Варианты индивидуальных заданий

№	$y=f(x)$	Тип кривой 2го порядка	№	$y=f(x)$	Тип кривой 2го порядка
1	$y = 8+2x$	парабола	16	$y = 17+9x$	парабола
2	$y = 7+3x$	гипербола	17	$y = 18+2x$	гипербола
3	$y = 6+4x$	эллипс	18	$y = 20+6x$	эллипс
4	$y = 5+x$	парабола	19	$y = 21+7x$	парабола
5	$y = 4+2x$	гипербола	20	$y = 23+9x$	гипербола
6	$y = 3+6x$	эллипс	21	$y = 26+4x$	эллипс
7	$y = 2+8x$	парабола	22	$y = 25+6x$	парабола
8	$y = 10+2x$	гипербола	23	$y = 34+5x$	гипербола
9	$y = 9+3x$	эллипс	24	$y = 30+7x$	эллипс
10	$y = 11+7x$	парабола	25	$y = 32+18x$	парабола
11	$y = 12+4x$	гипербола	26	$y = 40+12x$	гипербола
12	$y = 13+6x$	эллипс	27	$y = 19+9x$	эллипс
13	$y = 14+x$	парабола	28	$y = 51+2x$	парабола
14	$y = 15+5x$	гипербола	29	$y = 62+8x$	гипербола
15	$y = 16+3x$	эллипс	30	$y=16+10x$	эллипс

Таблица 2.27 – Варианты индивидуальных заданий

№	Система уравнений	№	Система уравнений
1	$\begin{cases} \arccos(x) + \frac{x-2}{\sin(x)+1} = 0 \\ x^2 - 5 = 0 \end{cases}$	6	$\begin{cases} \sqrt[3]{\tan(x) + \log(x) + 7} = 0 \\ \sqrt{x^2 + \cos(x) - 9} = 0 \end{cases}$
2	$\begin{cases} \sin(x) + 1 = 0 \\ x^2 - 5 + \cos(x) = 0 \end{cases}$	7	$\begin{cases} \frac{x^3 - x^2 + 5x}{x^3 - 1} = 0 \\ \log(x^3 - e^x + 2) = 0 \end{cases}$
3	$\begin{cases} \sin(x) \cdot \cos(x-2) = 0 \\ x^2 - 3 + \cos(2x) = 0 \end{cases}$	8	$\begin{cases} 2x^2 + 3x - 8 = 0 \\ \frac{x-3}{x^2} = 0 \end{cases}$
4	$\begin{cases} \frac{\sin(x) \cos(x)}{x-3} = 0 \\ x^2 - 3 = 0 \end{cases}$	9	$\begin{cases} 2x^2 + 3x - \cos(\pi x) = 0 \\ 3x + 1 = 0 \end{cases}$
5	$\begin{cases} \frac{\sin(x)}{x-3} = 0 \\ x^2 - \frac{3}{x} = 0 \end{cases}$	10	$\begin{cases} e^x - 1 = 0 \\ x^2 + 3x = 0 \end{cases}$
11	$\begin{cases} \frac{\arcsin(x)}{\log(x)} + 5 = 0 \\ x^2 - 2x + 4 = 0 \end{cases}$	21	$\begin{cases} \frac{x^2 + 5x}{x^3 - 1} = 0 \\ \log(x^2 - 1^x) + 7 = 0 \end{cases}$
12	$\begin{cases} \frac{\arcsin(x)}{\ln(x)} + 5 = 0 \\ \sqrt{\sin(x) - \cos(3x)} = 0 \end{cases}$	22	$\begin{cases} \sin(1/x) + 5 = 0 \\ e^{2x} + 2 = 0 \end{cases}$
13	$\begin{cases} \sqrt[3]{\ln(x) + \log(x) + 4} = 0 \\ \sqrt{(2x)^3 + 2x^2 - 9} = 0 \end{cases}$	23	$\begin{cases} \frac{x^2 + 5x}{x^3 + 2x^2 - 1} = 0 \\ \ln(x^2 - 1) + 7 = 0 \end{cases}$
14	$\begin{cases} \sqrt[4]{x + \sin(x)} = 0 \\ \ln(x^2 + \sin(x)) = 0 \end{cases}$	24	$\begin{cases} 1/x^2 - x^2 = 0 \\ (3e)^{2x} - 1 = 0 \end{cases}$
15	$\begin{cases} \frac{x^3 - 4x^2 + x}{x-5} = 0 \\ \log(x^3 + \sin(2x)) = 0 \end{cases}$	25	$\begin{cases} \sin(x^2)/3x = 0 \\ \ln(x^{0.1} - 0.2^x) = 0 \end{cases}$
16	$\begin{cases} \frac{\arcsin(x)}{\log(x)} + 5 = 0 \\ \frac{x^2 - 2x + 4}{3x} = 0 \end{cases}$	26	$\begin{cases} 2^x - 1 = 0 \\ 2x^2 + 3x - \cos(x) = 0 \end{cases}$

Продолжение таблицы 2.27.

№	Система уравнений	№	Система уравнений
17	$\begin{cases} \frac{\sin(x)}{\ln(x)} = 0 \\ \sqrt{x^3 + 2x^2 - 6} = 0 \end{cases}$	27	$\begin{cases} 1/x^2 = 0 \\ e^{2x} - 1 = 0 \end{cases}$
18	$\begin{cases} \sin(x) + 1 = 0 \\ x^2 - 5 + \cos(x) = 0 \end{cases}$	28	$\begin{cases} 2x^2 + 3x - \cos(\pi x) = 0 \\ 3x + 1 = 0 \end{cases}$
19	$\begin{cases} \arccos(x) + \frac{x-2}{\sin(x)+1} = 0 \\ x^2 - 5 = 0 \end{cases}$	29	$\begin{cases} \frac{\sin(x)\cos(x)}{x-3} = 0 \\ x^2 - 3 = 0 \end{cases}$
20	$\begin{cases} 2x^2 + 3x - 8 = 0 \\ \frac{x-3}{x^2} = 0 \end{cases}$	30	$\begin{cases} \frac{x^3 - x^2 + 5x}{x^3 - 1} = 0 \\ \log(x^3 - e^x + 2) = 0 \end{cases}$

2.2.4 Задание на лабораторную работу № 27

**Тема:** Аппроксимация кривых средствами математического пакета Mathcad.

**Цель работы:** провести разложение несинусоидальной кривой ЭДС в тригонометрический ряд Фурье с помощью математического пакета Mathcad.

**Теоретические сведения**

Разложение несинусоидальной кривой ЭДС источника в тригонометрический ряд Фурье включает в себя следующие этапы:

- а) формализация записи выходного сигнала источника энергии;
- б) разложение выходного сигнала источника ЭДС в ряд Фурье;
- в) представление разложенной и заданной кривой на графике.

В среде Mathcad формализация любого графика может быть выполнена с использованием всего двух элементов:

- а) логической функции if(логическое условие, значение, если истина, значение, если ложь);
- б) функции линейной интерполяции linterp(X, Y, x) [20].

Формализация кривой с помощью логической функции крайне неудобна при большом количестве узлов графика, но проста в записи и реализации. Применение linterp исключительно удобно для формализации графиков, заданных отрезками прямых или координатами узловых точек.

Однако отметим особенность этой функции – координаты массива X, который стоит на первом месте в linterp, должны монотонно убывать или возрастать. То есть, недопустимы одинаковые координаты, моделирующие вертикальный скачок значения моделируемой функции. При необходимости же моделирования скачка можно изменить значение одной или нескольких координат на такую малую величину, что это не отразится на дальнейших вычислениях. В нашем примере это делается за счет уменьшения соответствующих координат на ничтожно малую величину dr. При разложении формализованного выходного сигнала в ряд Фурье необходимо воспользоваться известной формулой [17]:

$$f(t) = A_0 + \sum_{k=1}^{\infty} (B_k \sin(kt) + C_k \cos(kt)),$$

где коэффициенты ряда Фурье:

$$A_0 = \frac{1}{T} \cdot \int f(t) dt;$$

$$B_k = \frac{2}{T} \cdot \int f(t) \sin(kt) dt;$$

$$C_k = \frac{2}{T} \cdot \int f(t) \cos(kt) dt.$$

При выполнении разложения следует не забывать определять, что нумерация элементов массива начинается с 1. Для того чтобы разложенная в ряд кривая была ближе к исходной, необходимо увеличить число гармоник, по которым идет суммирование.

### ***Индивидуальное задание***

Согласно вариантам индивидуального задания (таблица 2.28) выбрать тип несинусоидальной функции – кривой ЭДС и выполнить её разложение в ряд Фурье аналитически, для чего необходимо выполнить такие пункты:

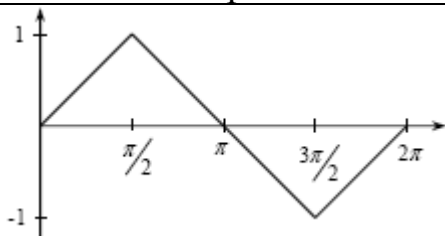
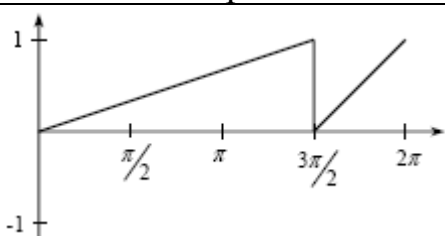
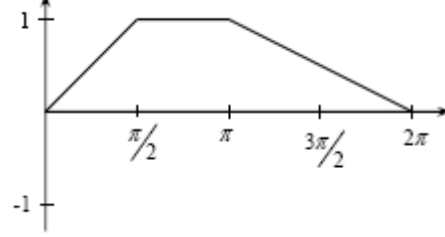
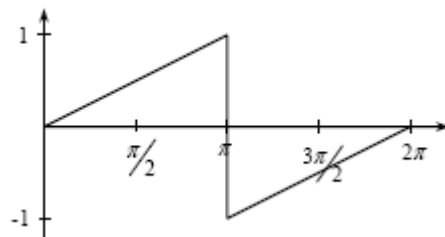
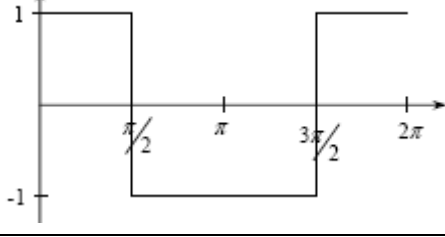
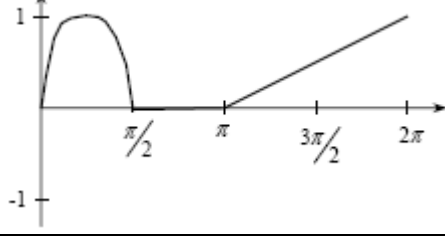
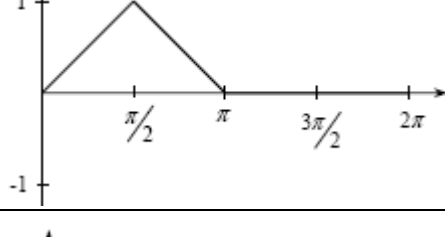
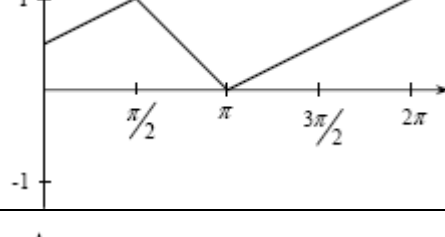
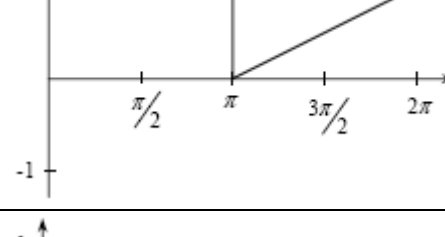
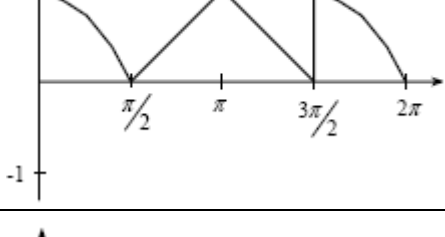
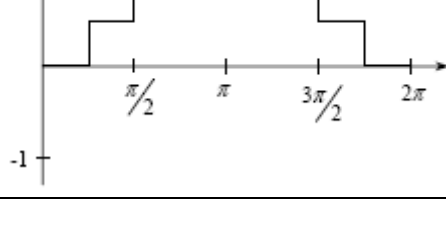
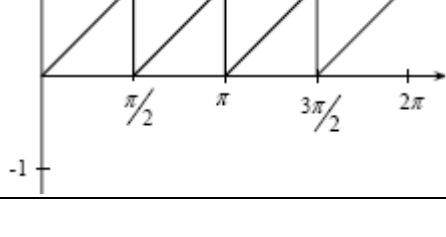
- а) задать период функции согласно виду кривой;
- б) задать координаты по осям X и Y и интерполировать координаты точек с помощью функции linterp;
- в) вывести на экран полученный график функции;
- г) задать количество гармоник и диапазонную переменную с последующим вычислением коэффициентов ряда Фурье;
- д) просуммировать ряд;

е) построить графики разложенной и исходной функций в одной плоскости.

Сделать выводы по особенностям полученных результатов работы.

Пример решения задачи в пакете Mathcad представлен на рисунке 2.52.

Таблица 2.28 – Варианты формы кривой ЭДС источника

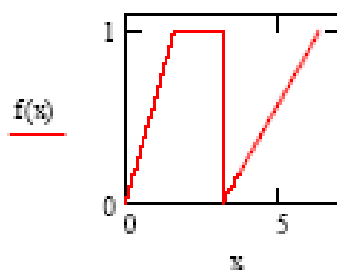
№	Вид кривой	№	Вид кривой
1		2	
3		4	
5		6	
7		8	
9		10	
11		12	

Зададим период функции:  $T := 2 \cdot \pi$      $dp := 1 \cdot 10^{-10}$      $ORIGIN := 1$

Зададим координаты по осям X и Y, и интерполируем координаты точек:

$$X := \begin{pmatrix} 0 \\ \pi \\ \pi - dp \\ \pi \\ 2\pi - dp \\ 2\pi \end{pmatrix} \quad Y := \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad f(t) := \text{interp}(X, Y, t)$$

Выведем на экран график:     $x := 0, 0.001 .. T$



Зададим количество гармоник и диапазонную переменную:

$$kg := 10 \quad k := 1 .. kg$$

Вычислим значения коэффициентов ряда Фурье     $A_0 := \left(\frac{1}{T}\right) \cdot \int_0^T f(t) dt$

$$B_k := \left(\frac{2}{T}\right) \cdot \int_0^T f(t) \cdot \sin(k \cdot t) dt \quad C_k := \left(\frac{2}{T}\right) \cdot \int_0^T f(t) \cdot \cos(k \cdot t) dt$$

Просуммируем ряд:

$$\hat{f}(t) := A_0 + \sum_k (B_k \cdot \sin(k \cdot t) + C_k \cdot \cos(k \cdot t))$$

Выведем на экран графики исходной и полученной функций:

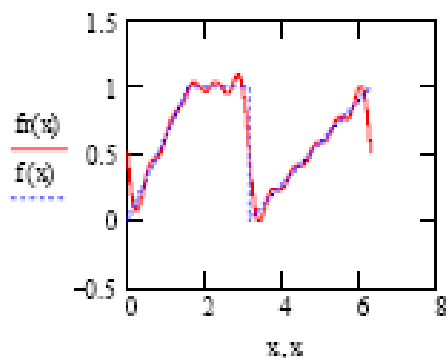
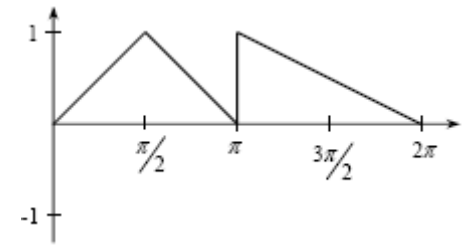
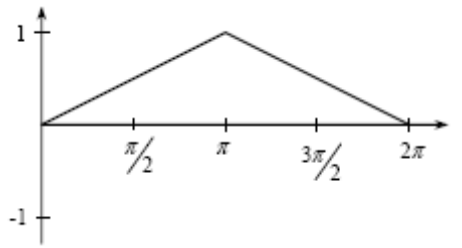
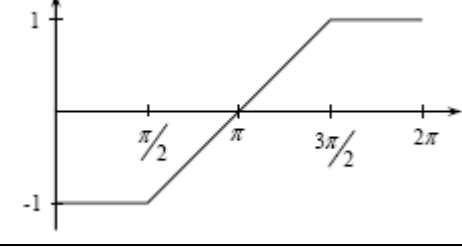
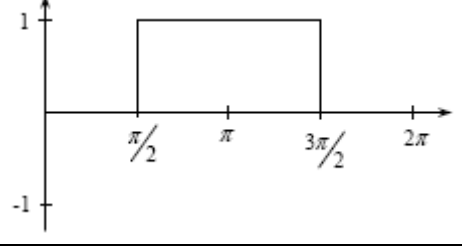
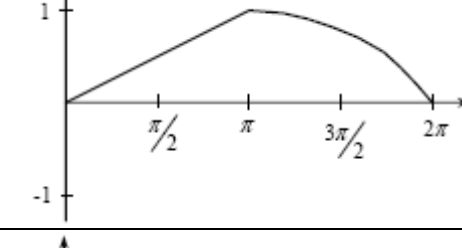
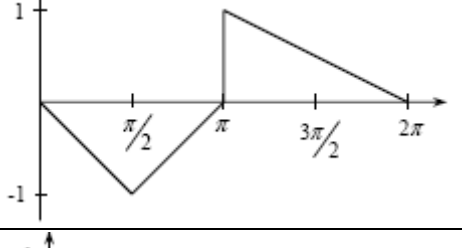
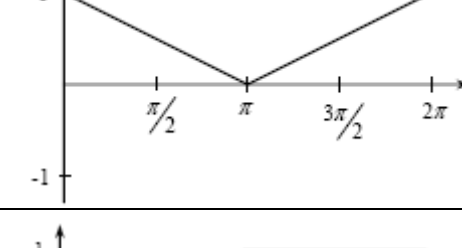
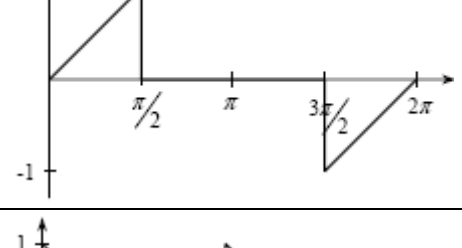
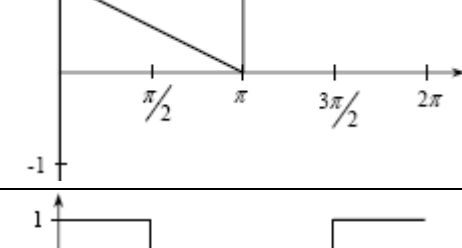
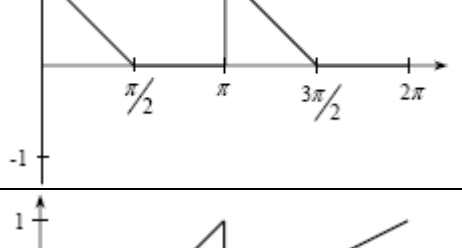
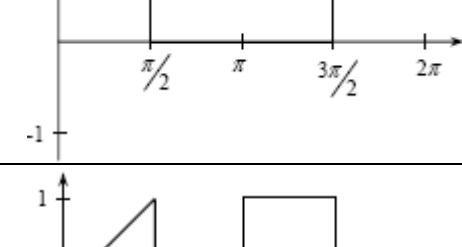
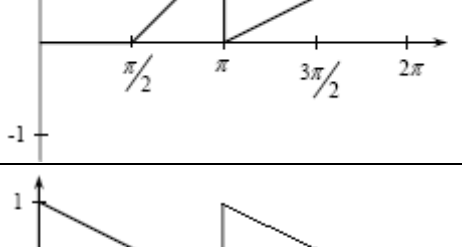
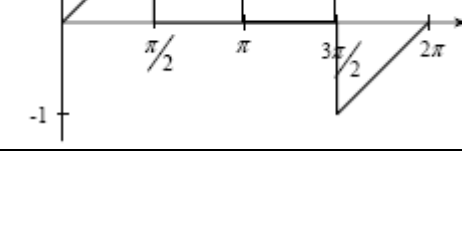
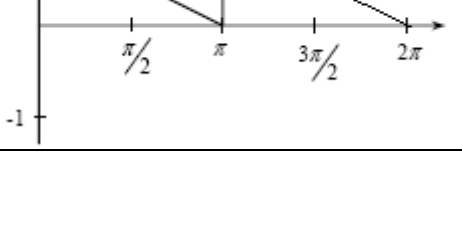
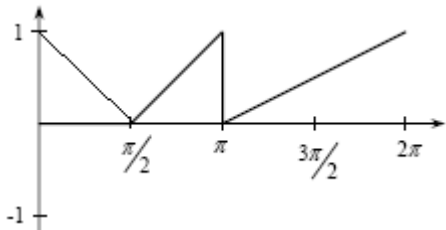
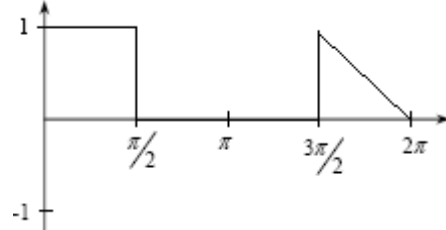
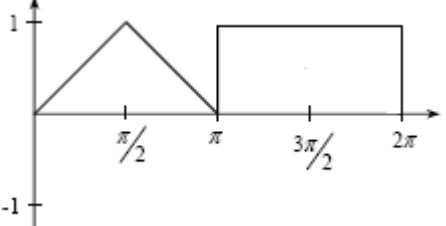
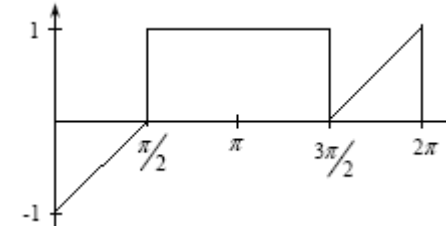


Рисунок 2.52 – Пример разложения функции в ряд Фурье

Продолжение таблицы 2.26.

№	Вид кривой	№	Вид кривой
13		14	
15		16	
17		18	
19		20	
21		22	
23		24	
25		26	

Продолжение таблицы 2.28.

№	Вид кривой	№	Вид кривой
27		28	
29		30	

**Контрольные вопросы**

- 1 Преимущества исследования полигармонических функций путем разложения в ряд Фурье.
- 2 Особенности функций пакета Mathcad для разложения в ряд и аппроксимации функций.
- 3 Правила симметрии несинусоидальных функций и правила вычисления их коэффициентов.
- 4 Методы сглаживания функций в пакете Mathcad.

2.2.5 Задание на лабораторную работу № 28

**Тема:** Определение наибольшего и наименьшего значений функций в пакете Mathcad (линейное программирование).

**Цель работы:** изучить свойства и особенности работы в среде пакета Mathcad, изучить правила задания данных, приобрести навыки поиска экстремумов функций.

**Теоретические сведения**

В общем случае задача линейного программирования формулируется следующим образом. Найти величины  $x_1, x_2, \dots, x_n$ , доставляющие минимум линейной функции:

$$f(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

и удовлетворяющие условиям, которые могут быть только равенствами и неравенствами вида:



$$\begin{aligned}
 a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n &\leq b_i, \\
 a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n &= b_k, \\
 a_{p1}x_1 + a_{p2}x_2 + \dots + a_{pn}x_n &\geq b_p, \quad i \neq k \neq p.
 \end{aligned}$$

Среди ограничений часто встречаются условия неотрицательности всех или части переменных:

$$x_j \geq 0.$$

Функция  $f(x_1, x_2, \dots, x_n)$  называется целевой функцией.

В практике планирования на любом его уровне возникает необходимость выбора оптимального варианта среди различных вариантов плана. Для решения подобных задач необходимы точные методы, дающие возможность сопоставлять различные варианты плана и выбирать оптимальный вариант. Один из методов, облегчающий нахождение оптимальных вариантов плана, носит название линейного программирования (в определении оптимального плана фигурируют только линейные уравнения или неравенства).

Решение задачи в данном методе состоит из трех этапов:

- а) составление математической модели;
- б) решение формализованной задачи;
- в) анализ полученного оптимального решения.

Формализация решения начинается с выбора переменных, совокупность числовых значений которых однозначно определяет один из вариантов плана [17]. При этом следует иметь в виду, что иной раз от удачного выбора этих переменных зависит простота модели и удобство ее дальнейшего анализа. После выбора переменных необходимо составить систему ограничений, которой эти переменные должны удовлетворять. При этом не допустимо включение лишних ограничений. То есть для системы уравнений согласно уравнениям ограничений на плоскости строится многоугольник – область решения (рисунок 2.53).

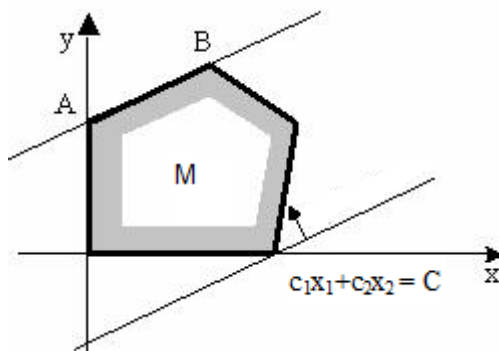


Рисунок 2.53 – Область решений целевой функции

Далее составляется целевая функция, которая в математической форме отражает критерий выбора лучшего варианта. То есть среди всех точек области – многоугольника необходимо найти такую, которая максимизирует (минимизирует) целевую функцию  $f$ .

Приравнявая выражение для функции  $f$  какой-либо постоянной ( $f = c_1x_1 + c_2x_2 = C$ ) получаем уравнение прямой линии, в точках которой целевая функция получает одно и тоже фиксированное значение  $C$ . Значение функции возрастает в направлении вектора градиента и убывает в направлении, противоположном направлению градиента. Значит для задачи максимума последняя точка области  $M$ , которую пересекает прямая при движении в направлении градиента, является точкой максимума, в противоположном направлении – точка минимума. Координаты оптимальной точки определяются из решения системы уравнений прямых, пересекающихся в этой точке.

Для решения задачи необходимо выполнить такие пункты:

- а) построить область, являющуюся пересечением всех полуплоскостей, заданных в качестве системы уравнений ограничений;
- б) найти градиент функции и построить вектор с началом в точке  $(0; 0)$  и концом в точке градиента;
- в) построить прямую, перпендикулярную вектору градиента и передвигая ее в направлении возрастания (максимум) или убывания (минимум) градиента определить точку (прямую) экстремума.

Метод решения задачи зависит от ее размерности. Если количество переменных равно 2, то задача называется плоской и может быть решена графически, но для многомерных задач этот метод уже не приемлем.

Для создания графиков в системе Mathcad имеется графический процессор. Для построения графиков используются шаблоны. Их перечень содержит математическая панель Graph [21]:

- а) X-Y Plot – создать шаблон двумерного графика в декартовой системе координат;
- б) Polar Plot – создать шаблон графика в полярных координатах;
- в) Surface Plot – создать шаблон для построения трехмерного графика;
- г) Countour Plot – создать шаблон для графика в виде точек (фигур) в трехмерном пространстве;
- д) 3D Scatter Plot – создать шаблон для изображения в виде совокупности столбиков в трехмерном пространстве;
- е) Vector Field Plot – создать шаблон для графика векторного поля на плоскости.

### ***Индивидуальное задание***

Найдите минимум целевой функции  $f(x,y)=ax+by$  при указанных ограничениях (таблица 2.29). Построить графики функций.

Решить задачу поиска экстремума двумя методами – графически и с помощью встроенной функции Mathcad Minimize.

Для графического метода (рисунок 2.54) порядок действий следующий:

1. Задаются на графике соответствующие прямые, и определяется область допустимых значений переменных.

2. Если задача имеет единственное решение, определяется вершина, в которой достигается искомое экстремальное значение (максимум или минимум) целевой функции и указываются ее координаты.

3. Вычисляется значение целевой функции в найденной точке.

Для численного метода (рисунок 2.55) необходимо задать функцию и её ограничения, а затем применить функцию поиска минимума Minimize.

Таблица 2.29 – Варианты индивидуальных заданий

№	$f(x, y)$	Ограничения				
1	$3x+2y$	$x+y \leq 4$	$x+2y \geq 5$	$2x+y \geq 6$	$x \geq 0$	$y \geq 0$
2	$3x+2y$	$x+y \leq 6$	$x+2y \geq 7$	$2x+y \geq 8$	$x \geq 0$	$y \geq 0$
3	$3x+2y$	$x+y \leq 8$	$x+2y \geq 8$	$2x+y \geq 8$	$x \geq 0$	$y \geq 0$
4	$3x+2y$	$x+y \leq 9$	$x+2y \geq 7$	$2x+y \geq 6$	$x \geq 0$	$y \geq 0$
5	$3x+2y$	$x+y \leq 6$	$x+2y \geq 5$	$2x+y \geq 3$	$x \geq 0$	$y \geq 0$
6	$2x+5y$	$x+y \geq 4$	$x+2y \leq 13$	$2x+y \leq 13$	$x \geq 0$	$y \geq 0$
7	$2x+5y$	$x+y \geq 5$	$x+2y \leq 14$	$2x+y \leq 14$	$x \geq 0$	$y \geq 0$
8	$2x+5y$	$x+y \geq 5$	$x+2y \leq 15$	$2x+y \leq 15$	$x \geq 0$	$y \geq 0$
9	$2x+5y$	$x+y \geq 6$	$x+2y \leq 11$	$2x+y \leq 8$	$x \geq 0$	$y \geq 0$
10	$2x+5y$	$x+y \geq 7$	$x+2y \leq 13$	$2x+y \leq 9$	$x \geq 0$	$y \geq 0$
11	$5x+4y$	$x+y \leq 5$	$x+4y \geq 7$	$3x+y \geq 7$	$x \geq 0$	$y \geq 0$
12	$5x+4y$	$x+y \leq 4$	$x+4y \geq 9$	$3x+y \geq 5$	$x \geq 0$	$y \geq 0$
13	$5x+4y$	$x+y \leq 10$	$x+4y \geq 10$	$3x+y \geq 11$	$x \geq 0$	$y \geq 0$
14	$5x+4y$	$x+y \leq 6$	$x+4y \geq 12$	$3x+y \geq 3$	$x \geq 0$	$y \geq 0$
15	$5x+4y$	$x+y \leq 7$	$x+4y \geq 14$	$3x+y \geq 4$	$x \geq 0$	$y \geq 0$
16	$4x+7y$	$x+y \geq 3$	$x+6y \leq 5$	$4x+y \leq 4$	$x \geq 0$	$y \geq 0$
17	$4x+7y$	$x+y \geq 5$	$x+6y \leq 7$	$4x+y \leq 6$	$x \geq 0$	$y \geq 0$
18	$4x+7y$	$x+y \geq 7$	$x+6y \leq 9$	$4x+y \leq 8$	$x \geq 0$	$y \geq 0$
19	$4x+7y$	$x+y \geq 9$	$x+6y \leq 3$	$4x+y \leq 10$	$x \geq 0$	$y \geq 0$
20	$4x+7y$	$x+y \geq 11$	$x+6y \leq 11$	$4x+y \leq 12$	$x \geq 0$	$y \geq 0$
21	$7x+6y$	$x+y \leq 4$	$x+8y \geq 10$	$5x+y \geq 3$	$x \geq 0$	$y \geq 0$
22	$7x+6y$	$x+y \leq 5$	$x+8y \geq 6$	$5x+y \geq 5$	$x \geq 0$	$y \geq 0$
23	$7x+6y$	$x+y \leq 7$	$x+8y \geq 8$	$5x+y \geq 7$	$x \geq 0$	$y \geq 0$
24	$7x+6y$	$x+y \leq 9$	$x+8y \geq 5$	$5x+y \geq 6$	$x \geq 0$	$y \geq 0$
25	$7x+6y$	$x+y \leq 11$	$x+8y \geq 12$	$5x+y \geq 9$	$x \geq 0$	$y \geq 0$
26	$6x+9y$	$x+y \geq 3$	$x+10y \leq 10$	$6x+y \leq 8$	$x \geq 0$	$y \geq 0$
27	$6x+9y$	$x+y \geq 7$	$x+10y \leq 12$	$6x+y \leq 10$	$x \geq 0$	$y \geq 0$
28	$6x+9y$	$x+y \geq 5$	$x+10y \leq 7$	$6x+y \leq 12$	$x \geq 0$	$y \geq 0$

Продолжение таблицы 2.29.

№	$f(x, y)$	Ограничения				
29	$6x+9y$	$x+y \geq 9$	$x+10y \leq 6$	$6x+y \leq 6$	$x \geq 0$	$y \geq 0$
30	$6x+9y$	$x+y \geq 6$	$x+10y \leq 8$	$6x+y \leq 4$	$x \geq 0$	$y \geq 0$

Найти минимум целевой функции  $f(x,y)=2x+3y$  при ограничениях

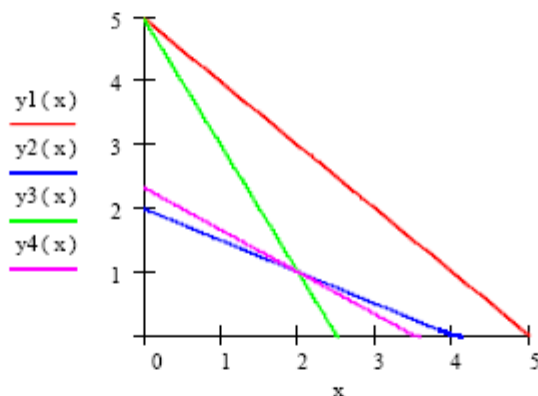
$$x+y \leq 5, \quad x+2y \geq 4, \quad 2x+y \geq 5, \quad x \geq 0, \quad y \geq 0.$$

$$x + y \leq 5 \quad -x + 5 \quad y_1(x) := -x + 5$$

$$x + 2 \cdot y \geq 4 \quad \frac{-1}{2} \cdot x + 2 \quad y_2(x) := \frac{-1}{2} \cdot x + 2$$

$$2 \cdot x + y \geq 5 \quad -2 \cdot x + 5 \quad y_3(x) := -2 \cdot x + 5$$

$$2 \cdot x + 3 \cdot y \leq c \quad \frac{-2}{3} \cdot x + \frac{1}{3} \cdot c \quad c := 7 \quad y_4(x) := \frac{-2}{3} \cdot x + \frac{1}{3} \cdot c$$



Задача имеет единственное решение. Минимум целевой функции достигается в точке пересечения прямых  $x+2y=4$ ,  $2x+y=5$

$$\text{Given } \begin{array}{l} x + 2 \cdot y \geq 4 \\ 2 \cdot x + y \geq 5 \end{array} \quad \text{Find}(x, y) \rightarrow \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$f(x, y) := 2 \cdot x + 3 \cdot y \quad f_{\min} := f(2, 1) \quad f_{\min} = 7$$

Минимальное значение целевой функции равно 7, оно достигается в точке  $x=2$ ,  $y=1$ .  $f_{\min}=f(2,1)=7$ .

Рисунок 2.54 – Графический метод решения

$$f(x,y) := 2 \cdot x + 3 \cdot y$$

$$x := 1 \quad y := 1$$

Given

$$x \geq 0 \quad y \geq 0 \quad y \leq -x + 5$$

$$y \geq \frac{-x}{2} + 2$$

$$y \geq 5 - 2 \cdot x$$

$$P := \text{Minimize}(f, x, y)$$

$$P = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad f(P_0, P_1) = 7$$

Рисунок 2.55 – Метод решения посредством встроенных функций

### **Контрольные вопросы**

1. Понятие целевой функции.
2. Функции построения графиков разных зависимостей в среде Mathcad.
3. Постановка задачи линейного программирования
4. Особенности поиска минимума и максимума гладких функций.
5. Функции поиска минимума функций средствами математического пакета Mathcad.

#### 2.2.6 Задание на лабораторную работу № 29

**Тема:** Методы и особенности обработки экспериментальных данных средствами пакета Mathcad.

**Цель работы:** изучить методы и способы обработки экспериментальных данных, приобрести навыки работы с функциями интерполяции, регрессии, сглаживания пакета Mathcad.

### **Теоретические сведения**

При проведении научно-технических расчетов часто используются эмпирические зависимости, причем число точек этих зависимостей ограничено.

Неизбежно возникает задача восполнения данных: имеется некоторое количество точек, через которые следует провести кривую. Это не что иное, как классическая задача интерполяции.

Интерполяция – частный случай более общей задачи аппроксимации (приближенного представления), возникающей при замене кривой, описываемой функцией сложной природы, другой кривой, в некотором смысле близкой заданной, имеющей более простые уравнения.

Задача сглаживания кривой возникает, когда данные, используемые для ее восстановления, определены в результате измерений или эмпирически с некоторой погрешностью либо представляет кривую, описываемую функцией, недостаточно гладкой (например, недифференцируемой или дифференцируемой всего несколько раз).

Система Mathcad предоставляет возможность интерполяции двух типов: кусочно-линейной и сплайновой.

Функция  $\text{linterp}(VX, VY, x)$  – для заданных векторов  $VX$  и  $VY$  узловых точек и заданного аргумента  $x$  эта функция возвращает значение функции при ее линейной интерполяции.

При небольшом числе узловых точек (менее 10) линейная интерполяция оказывается довольно грубой. При ней даже первая производная функции интерполяции испытывает резкие скачки в узловых точках.

Гораздо лучшие результаты дает сплайн-интерполяция. При ней исходная функция заменяется отрезками кубических полиномов, проходящих через три смежные узловые точки. Коэффициенты полиномов рассчитываются так, чтобы непрерывными были первая и вторая производные. Линия, которую описывает сплайн-функция, напоминает по форме гибкую линейку, закрепленную в узловых точках (отсюда и название интерполяции: spline – гибкая линейка).

Для осуществления сплайновой аппроксимации система Mathcad предлагает четыре встроенные функции. Три из них служат для получения векторов вторых производных сплайн-функций при различном виде интерполяции [21]:

а)  $\text{cspline}(VX, VY)$  — возвращает вектор  $VS$  вторых производных при приближении в опорных точках к кубическому полиному;

б)  $\text{pspline}(VX, VY)$  — возвращает вектор  $VS$  вторых производных при приближении к опорным точкам параболической кривой;

в)  $\text{lspline}(VX, VY)$  — возвращает вектор  $VS$  вторых производных при приближении к опорным точкам прямой;

г)  $\text{interp}(VS, VX, VY, x)$  - возвращает значение  $y(x)$  для заданных векторов  $VS, VX, VY$  и заданного значения  $x$ .

Таким образом, сплайн-аппроксимация проводится в два этапа. На первом с помощью функций  $\text{cspline}$ ,  $\text{pspline}$  или  $\text{lspline}$  отыскивается вектор вторых производных функции  $y(x)$ , заданной векторами  $VX$  и  $VY$  ее значений (абсцисс и ординат). Затем, на втором этапе для каждой искомой точки вычисляется значение  $y(x)$  с помощью функции  $\text{interp}$ .

Другой широко распространенной задачей обработки данных является представление их совокупности некоторой функцией  $y(x)$ . Задача регрессии заключается в получении параметров этой функции такими, чтобы функция приближала облако исходных точек (заданных векторами  $VX$  и  $VY$ ) с наименьшей среднеквадратичной погрешностью. Чаще всего используется линейная регрессия, при которой функция  $y(x)$  имеет вид  $y(x) = a + b \cdot x$  и описывает отрезок прямой. К линейной регрессии можно свести многие виды нелинейной регрессии при двухпараметрических зависимостях  $y(x)$ .

Для проведения линейной регрессии в систему встроен ряд приведенных ниже функций:

- а)  $\text{corr}(VX, VY)$  – возвращает скаляр-коэффициент корреляции Пирсона;
- б)  $\text{interp}(VX, VY)$  – возвращает значение параметра  $a$  (смещение линии регрессии по вертикали);
- в)  $\text{slope}(VX, VY)$  – возвращает значение параметра  $b$  (наклона линии регрессии).

В Mathcad реализована возможность выполнения линейной регрессии общего вида. При ней заданная совокупность точек приближается функцией вида:

$$F(x, K1, K2, \dots, Kn) = K1 \cdot F1(x) + K2 \cdot F2(x) + \dots + Kn \cdot Fn(x).$$

Таким образом, функция регрессии является линейной комбинацией функций  $K1 \cdot F1(x)$ ,  $K2 \cdot F2(x)$ , ...,  $Kn \cdot Fn(x)$ , причем сами эти функции могут быть нелинейными, что резко расширяет возможности такой аппроксимации и распространяет ее на нелинейные функции.

Для реализации линейной регрессии общего вида используется функция  $\text{lfit}(VX, VY, F)$ . Эта функция возвращает вектор коэффициентов линейной регрессии общего вида  $K$ , при котором среднеквадратичная погрешность приближения облака исходных точек, если их координаты хранятся в векторах  $VX$  и  $VY$ , оказывается минимальной. Вектор  $F$  должен содержать функции  $F1(x)$ ,  $F2(x)$ , ...,  $Fn(x)$ , записанные в символьном виде.

Введена в новую версию Mathcad и функция для обеспечения полиномиальной регрессии при произвольной степени полинома регрессии  $\text{regress}(VX, VY, n)$ . Она возвращает вектор  $VS$ , запрашиваемый функцией  $\text{interp}(VS, VX, VY, x)$ , содержащий коэффициенты многочлена  $n$ -й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в векторах  $VX$  и  $VY$ .

На практике не рекомендуется делать степень аппроксимирующего полинома выше четвертой – шестой, поскольку погрешности реализации регрессии сильно возрастают.

Функция  $\text{regress}$  создаст единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек, т. е. глобально. Иногда полезна другая функция полиномиальной

регрессии, дающая локальные приближения отрезками полиномов второй степени,  $\text{loess}(VX, VY, \text{span})$ . Эта функция возвращает используемый функцией  $\text{interp}(VS, VX, VY, x)$  вектор  $VS$ , дающий наилучшее приближение данных (с координатами точек в векторах  $VX$  и  $VY$ ) отрезками полиномов второй степени. Аргумент  $\text{span} > 0$  указывает размер локальной области приближаемых данных (рекомендуемое начальное значение – 0,75). Чем больше  $\text{span}$ , тем сильнее сказывается сглаживание данных. При больших  $\text{span}$  эта функция приближается к  $\text{regress}(VX, VY, 2)$ .

Mathcad позволяет выполнять также многомерную регрессию, самый типичный случай которой – приближение трехмерных поверхностей. Их можно характеризовать массивом значений высот  $z$ , соответствующих двумерному массиву  $Mxy$  координат точек  $(x, y)$  на горизонтальной плоскости.

Новых функций для этого не задано. Используются уже описанные функции в несколько иной форме:

а)  $\text{regress}(Mxy, Vz, n)$  – возвращает вектор, запрашиваемый функцией  $\text{interp}(VS, Mxy, Vz, V)$  для вычисления многочлена  $n$ -й степени, который наилучшим образом приближает точки множества  $Mxy$  и  $Vz$ .  $Mxy$  – матрица  $m \cdot 2$ , содержащая координаты  $x$  и  $y$ .  $Vz$  –  $m$ -мерный вектор, содержащий  $z$ -координат, соответствующих  $m$  точкам, указанным в  $Mxy$ ;

б)  $\text{loes}(Mxy, Vz, \text{span})$  – аналогична  $\text{loes}(VX, VY, \text{span})$ , но в многомерном случае;

в)  $\text{Interp}(VS, Mxy, Vz, V)$  – возвращает значение  $z$  по заданным векторам  $VS$  (создается функциями  $\text{regress}$  или  $\text{loess}$ ) и  $Mxy$ ,  $Vz$  и  $V$  (вектор координат  $x$  и  $y$  заданной точки, для которой находится  $z$ ).

Под нелинейной регрессией общего вида подразумевается нахождение вектора  $K$  параметров произвольной функции  $F(x, K1, K2, \dots, Kn)$ , при котором обеспечивается минимальная среднеквадратичная погрешность приближения облака исходных точек.

Для проведения нелинейной регрессии общего вида используется функция  $\text{genfit}(VX, VY, VS, F)$ . Эта функция возвращает вектор  $K$  параметров функции  $F$ , дающий минимальную среднеквадратичную погрешность приближения функцией  $F(x, K1, K2, \dots, Kn)$  исходных данных.

$F$  должен быть вектором с символьными элементами, содержащими уравнение исходной функции и ее производных по всем параметрам. Вектор  $VS$  должен содержать начальные значения элементов вектора  $K$ , необходимые для решения системы нелинейных уравнений регрессии итерационным методом.

Данные большинства экспериментов имеют случайные составляющие погрешности. Поэтому часто возникает необходимость статистического сглаживания данных. Ряд функций Mathcad предназначен для выполнения операций сглаживания данных различными методами (в их названии имеется слово  $\text{smooth}$  – гладкий).



## Индивидуальное задание

Построить интерполяционный линейный и кубический сплайн для функции  $y=f(x)$ , заданной таблицей. Сравнить результаты и сделать выводы. Используя найденную зависимость, найти значение  $y$  в точке  $x = N + 0.55$ , где  $N$  – номер варианта. Пример аппроксимации приведен на рисунке 2.56.

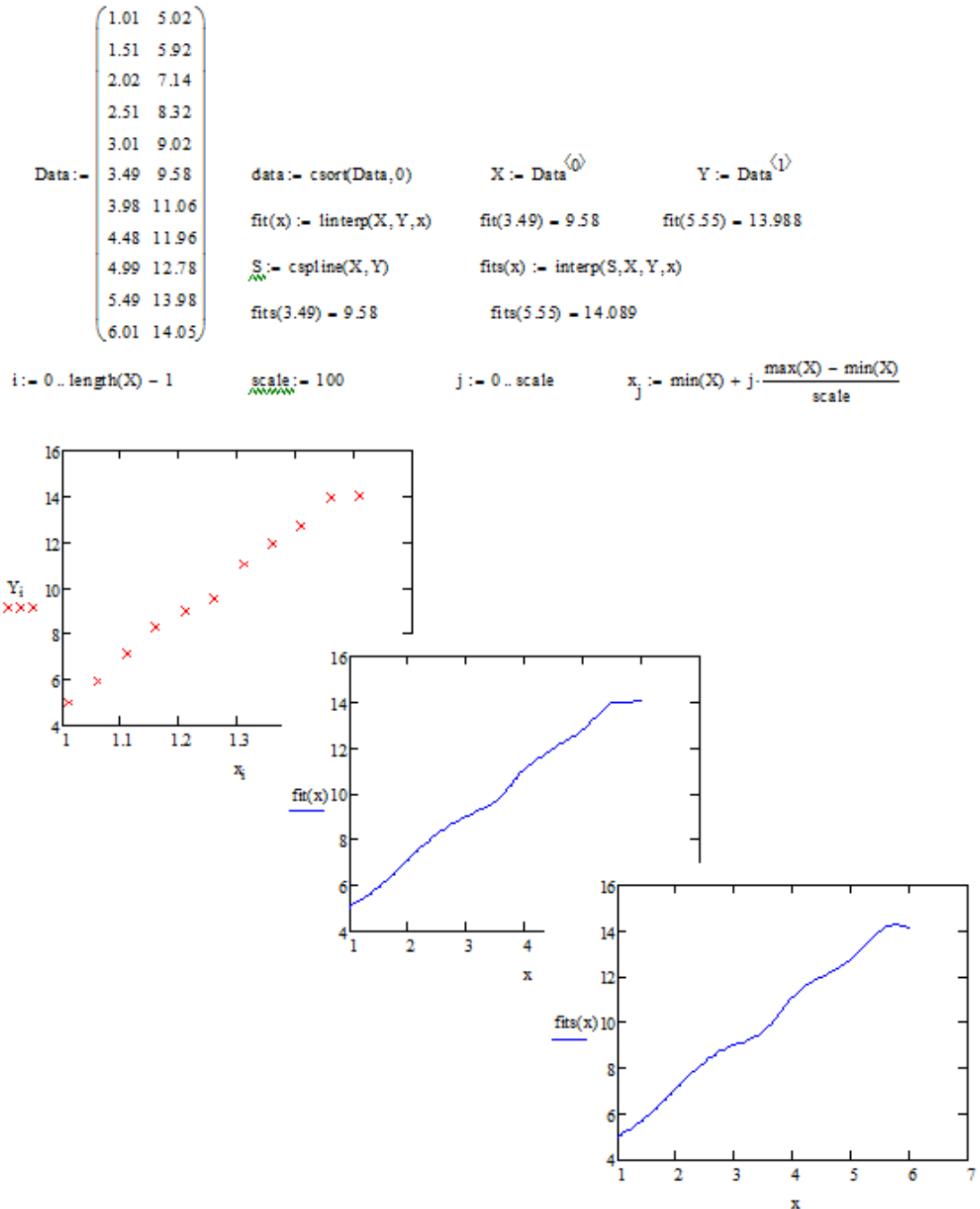


Рисунок 2.56 – Аппроксимация кривой линейным и кубическим сплайном

Таблица 2.30 – Варианты индивидуальных заданий

№	Значения X и Y											
	X	1.01	1.51	2.02	2.51	3.01	3.49	3.98	4.48	4.99	5.49	6.01
1	X	1.01	1.51	2.02	2.51	3.01	3.49	3.98	4.48	4.99	5.49	6.01
	Y	5.02	5.92	7.14	8.32	9.02	9.58	11.06	11.96	12.78	13.98	14.05
2	X	1,02	1,59	2,12	2,61	3,05	3,56	4	4,5	5,03	5,56	5,78
	Y	3,04	4,03	4,4	5,02	5,43	5,34	6,07	5,71	6,63	6,03	6,98
3	X	1,1	1,55	2,09	2,52	3,07	3,57	4,05	4,56	5,06	5,53	5,6
	Y	3,12	4,69	8,86	16,46	31,08	53,13	81,2	122,6	173,5	234,3	235,2
4	X	1,08	1,53	2,05	2,58	3,02	3,58	4,06	4,56	5,01	5,51	5,54
	Y	1,04	4,09	6,39	6,15	6,18	5,42	6,53	8,04	12,31	19,3	20,4
5	X	1,1	1,33	1,58	1,81	2,09	2,32	2,59	2,85	3,14	3,43	3,5
	Y	3,46	5,52	9,24	12,7	19,13	24,92	34,35	43,86	57,61	71,99	72
6	X	1,21	1,3	1,56	1,9	2,19	2,35	2,61	3,01	3,16	3,53	3,58
	Y	75,4	95,8	216,5	599,7	1425	2310	5030	16698	26190	79471	79501
7	X	2,01	2,51	3,02	3,51	4,01	4,49	4,98	5,48	5,99	6,49	6,54
	Y	7,02	7,92	9,14	10,32	11,02	11,58	13,06	13,96	14,78	15,98	16,895
8	X	2,02	2,59	3,12	3,61	4,05	4,56	5	5,5	6,03	6,56	6,6
	Y	4,41	5	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36	6,78
9	X	2,1	2,55	3,09	3,52	4,07	4,57	5,05	5,56	6,06	6,53	6,68
	Y	9,18	17	31,88	50,53	82,49	123,5	172,5	239,3	318	407,7	407,89
10	X	2,08	2,53	3,05	3,58	4,02	4,58	5,06	5,56	6,01	6,51	6,6
	Y	6,34	6,17	6,25	5,42	6,24	8,16	13,08	20,02	30,46	45,73	47,896
11	X	2,1	2,33	2,58	2,81	3,09	3,32	3,59	3,85	4,14	4,43	4,5
	Y	19,37	25,42	33,9	42,11	54,78	66,08	82,38	98,92	121	144,1	145,6
12	X	0,1	0,33	0,58	0,81	1,09	1,32	1,59	1,85	2,14	2,43	2,56
	Y	2,7	2,38	12,39	24,72	50,62	108,9	235,8	512,5	1228	2931	3000
13	X	3,01	3,51	4,02	4,51	5,01	5,49	5,98	6,48	6,99	7,49	7,54
	Y	9,02	9,92	11,14	12,32	13,02	13,58	15,06	15,96	16,78	17,98	18,59
14	X	3,02	3,59	4,12	4,61	5,05	5,56	6	6,5	7,03	7,56	7,64
	Y	5,21	5,66	5,73	6,16	6,44	6,23	6,88	6,44	7,3	6,65	6,78
15	X	3,1	3,55	4,09	4,52	5,07	5,57	6,05	6,56	7,06	7,53	7,69
	Y	32,44	51,92	83,99	118,8	174,7	240,6	316,4	414,7	527,2	651,5	652,8
16	X	3,08	3,53	4,05	4,58	5,02	5,58	6,06	6,56	7,01	7,51	7,68
	Y	6,12	5,42	6,41	8,16	12,42	20,39	31,99	47,36	66,67	93,22	94,23
17	X	3,1	3,33	3,58	3,81	4,09	4,32	4,59	4,85	5,14	5,43	5,5
	Y	55,26	66,82	81,67	96,07	116,7	134,8	159,6	184,6	216,6	250,1	251,46
18	X	1,17	1,53	1,56	1,85	2,09	2,15	2,31	3,01	3,16	3,23	3,36
	Y	66,9	194	216,5	516,5	1055	1269	2045	16698	26190	32310	32400
19	X	4,01	4,52	4,03	4,53	5,21	5,49	5,98	6,48	6,79	7,39	7,45
	Y	11	11,94	11,16	12,36	13,42	13,58	15,06	15,96	16,38	17,78	18,9
20	X	2,34	2,61	3,24	3,87	4,15	4,62	5,02	5,53	6,23	7,56	7,78
	Y	4,7	5,02	5,25	5,81	6,05	5,86	6,53	6,12	7,06	6,65	6,75
21	X	3,12	3,55	4,11	4,52	5,17	5,47	6,25	6,56	7,06	7,43	7,46
	Y	33,17	51,92	85,41	118,8	186,7	226,5	352,6	414,7	527,2	623,6	700,6
22	X	3,18	3,43	4,25	4,58	5,12	5,38	6,26	6,56	7,21	7,58	8,9
	Y	6,03	5,45	7	8,16	13,61	16,9	37,79	47,36	76,61	97,49	100
23	X	1,65	1,98	2,08	2,18	2,35	2,48	2,65	3,05	3,42	3,53	4,5
	Y	10,15	16,34	19,08	21,12	26,21	29,9	36,52	52,69	72,42	77,84	79,56
24	X	0,09	0,23	0,35	0,67	0,99	1,12	1,38	1,65	2,04	2,13	2,65
	Y	2,62	0,99	6,72	16,93	36,98	61,58	125,6	280,4	909,7	1192	1199
25	X	5,21	5,61	6,12	6,61	7,01	7,59	7,98	8,48	8,99	10,49	11,2
	Y	13,4	14,12	15,34	16,52	17,02	17,78	19,06	19,96	20,78	23,98	25
26	X	1,2	1,31	1,57	1,83	2,09	2,32	2,59	2,85	3,14	3,43	3,56
	Y	3,42	5,54	9,24	12,7	19,13	24,92	34,35	43,86	57,61	71,99	73,589

*Продолжение таблицы 2.30.*

27	X	1,25	1,32	1,56	1,9	2,19	2,35	2,61	3,01	3,16	3,53	4,569
	Y	75,5	95,8	216,5	599,7	1425	2310	5030	16698	26190	79471	80251
28	X	2,21	2,41	3,02	3,51	4,01	4,49	4,98	5,48	5,99	6,49	7,1
	Y	7,12	7,91	9,14	10,32	11,02	11,58	13,06	13,96	14,78	15,98	16,54
29	X	2,05	2,56	3,12	3,61	4,05	4,56	5	5,5	6,03	6,56	6,78
	Y	4,42	5,01	5,18	5,67	6	5,83	6,52	6,11	6,99	6,36	6,89
30	X	2,8	2,65	3,09	3,52	4,07	4,57	5,05	5,56	6,06	6,53	7,06
	Y	9,16	17,04	31,88	50,53	82,49	123,5	172,5	239,3	318	407,7	412,3

### 2.2.7 Задание на лабораторную работу № 30

**Тема:** Приближенное решение дифференциальных уравнений.

**Цель работы:** приобрести навыки решения дифференциальных уравнений с использованием встроенных функций Mathcad.

#### *Теоретические сведения*

В систему Mathcad введена возможность решения дифференциальных уравнений и систем с такими уравнениями в численном виде. Эту возможность трудно переоценить, так как многие серьезные научно-технические задачи (особенно относящиеся к анализу динамических систем и к их математическому моделированию) базируются на численных методах решения систем дифференциальных уравнений [20].

Нелинейные дифференциальные уравнения и системы с такими уравнениями, как правило, не имеют аналитических методов решения, и здесь особенно важна возможность их решения численными методами. В большинстве случаев желательно представление решений в графическом виде, что и позволяет Mathcad.

Для решения дифференциальных уравнений (систем) различного порядка и различными методами в Mathcad введены 13 встроенных функций: rkadapt, Rkadapt, rkfixed, Bulstoer, bulstoer, bvalfit, multigird, relax, sbval, Stiffb, stiffb, Stiffr и stiffr [21].

Пример решения обыкновенного дифференциального уравнения с помощью одной из описанных функций приведен на рисунке 2.57.

В функцию rkfixed заложен широко распространенный метод решения дифференциальных уравнений – метод Рунге-Кутты. Несмотря на то, что это не самый быстрый метод, функция rkfixed почти всегда справляется с поставленной задачей. Однако есть случаи, когда лучше использовать более сложные методы. Эти случаи попадают под три широкие категории: система может быть жесткой (Stiffb, Stiffr), функции системы могут гладкими (Bulstoer) или плавными (Rkadapt).

Нередко приходится пробовать несколько методов на одном дифференциальном уравнении (на одной системе), чтобы определить, какой метод лучше. Как известно, что решение гладкое, используется функция `Vulstoer`, куда заложен метод Бурлиш-Штера, а не Рунге-Кутта, используемый функцией `rkfixed`. В этом случае решение будет точнее. Можно решить задачу более точно (более быстро), если уменьшить шаг там, где производная меняется быстро, и увеличить шаг там, где она ведет более спокойно. Для этого предусмотрена функция `Rkadar`. Но, несмотря на то что она при решении дифференциального уравнения использует непостоянный шаг, функция `Rkadar` представит ответ для точек, находящихся на одинаковом расстоянии, заданном пользователем.



Рисунок 2.57 – Пример решения уравнения с помощью функции `rkfixed`

Система дифференциальных уравнений, записанная в матричной форме  $y=A \cdot x$ , где  $A$  – почти вырожденная матрица, называется жесткой. При решении жестких систем следует использовать одну из двух встроенных функций, разработанных специально для таких случаев: `Stiffb` и `Stiffc`. Они используют метод Булирш-Штера ( $b$ ) или Розенброка ( $r$ ). Функции, начинающиеся со строчной буквы, дают решения только для конечной точки. Для решения двухточечных краевых задач предназначены функции: `sbval` и `bvalfit`. Для решения дифференциальных уравнений

Пуассона (в частных производных второго порядка) и уравнений Лапласа в систему введены следующие функции: *bvalfit*, *multigrid*, *relax*, *sbval*.

Большое число примеров на решение дифференциальных уравнений с описанными функциями дается в подсказках QuickSheets, размещенных в центре ресурсов.

Пример численной реализации метода Рунге-Кутты приведен на рисунке 2.58.

$$f(x,y) \equiv \frac{10}{x+1} - y \quad \text{startx} \equiv 0 \quad \text{endx} \equiv 15 \quad n \equiv 100 \quad \text{inity} \equiv 1$$

$$k1(x,y,h) \equiv f(x,y)$$

$$k2(x,y,h,hd) \equiv f(x+hd, y+hd \cdot k1(x,y,h))$$

$$k3(x,y,h,hd) \equiv f(x+hd, y+hd \cdot k2(x,y,h,hd))$$

$$k4(x,y,h,hd) \equiv f(x+hd, y+hd \cdot k3(x,y,h,hd))$$

$$rk(x,y,h,hd) \equiv k1(x,y,h) + 2 \cdot k2(x,y,h,hd) + 2 \cdot k3(x,y,h,hd) + k4(x,y,h,hd)$$

$$x_0 \equiv \text{startx} \quad h \equiv \frac{\text{endx} - \text{startx}}{n} \quad \text{hd} \equiv 0.5 \cdot h$$

$$j \equiv 1..n \quad x_j \equiv \text{startx} + j \cdot h \quad y_0 \equiv 1 \quad y_j := y_{j-1} + rk(x_{j-1}, y_{j-1}, h, hd) \cdot \frac{h}{6}$$

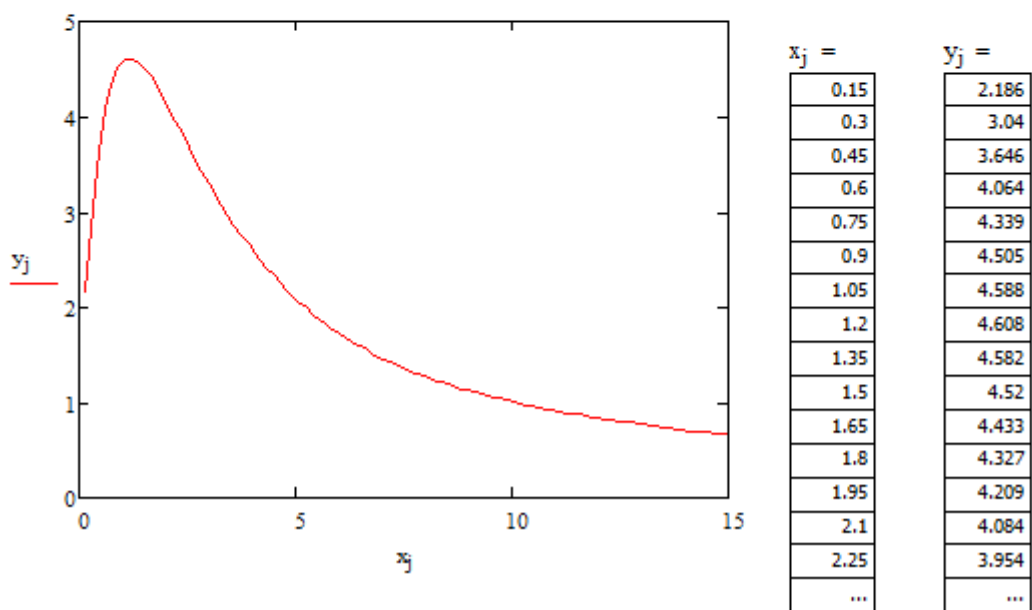


Рисунок 2.58 – Реализация метода Рунге-Кутты

### Индивидуальное задание

Построить график и вывести в виде таблицы решение задачи Коши на интервале  $[0; 1]$  методом Рунге-Кутты 4-го порядка (данные таблицы 2.14). Выполнить сравнительный анализ методик решения в различных математических пакетах.

## 2.2.8 Задание на лабораторную работу № 31

**Тема:** Построение поверхностей, заданных в явном виде в декартовой системе координат.

**Цель работы:** приобрести навыки работы с функциями построения графиков в трехмерном пространстве пакета Mathcad.

### *Теоретические сведения*

Построение поверхностей в декартовой системе координат рассмотрим на примере поверхности параболоида вращения.

Построение параболоида вращения ( $F(X, Y) = X^2 + Y^2$ ) производится в декартовой системе координат. Функция двух переменных, определяющая параболоид вращения.

Необходимо определить интервалы, в пределах которых изменяются значения  $X$  и  $Y$  ( $a := -5$   $b := 5$ ) и ввести количество дискретных точек по осям  $X$  и  $Y$  (рисунок 2.59).

$$M := 25 \quad N := 25 \quad i := 0..M \quad j := 0..N$$
$$X_{i,j} := a + (b - a) \cdot \frac{i}{M} \quad Y_{i,j} := a + (b - a) \cdot \frac{j}{N} \quad Z_{i,j} := F(X_{i,j}, Y_{i,j})$$

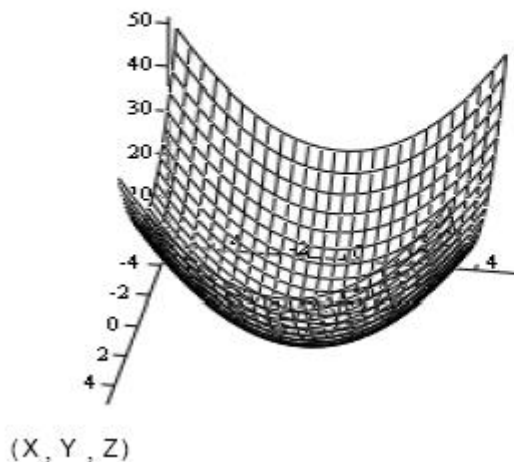


Рисунок 2.59 – Построение поверхности

### *Индивидуальное задание*

Построить графиков функций от двух переменных в декартовой системе координат. Варианты заданий – таблица 2.19. Выполнить сравнительный анализ построения поверхностей в различных математических пакетах.

## 2.2.9 Задание на лабораторную работу № 32

**Тема:** Расчет линейных цепей постоянного тока при помощи матриц и с использованием блока решений Given математического пакета Mathcad.

**Цель работы:** получить практические навыки применения матричного исчисления на основании стандартных и специфических функций работы с матрицами в пакете Mathcad.

### Теоретические сведения

Рассмотрим пример решения электротехнической задачи [16].

На рисунке 2.60 дана электрическая цепь постоянного тока.

В схеме дано:  $E_1=30$  В,  $E_4=100$  В,  $R_1=2$  Ом,  $R_2=6$  Ом,  $R_3=2,5$  Ом,  $R_4=10$  Ом,  $R_5=4$  Ом.

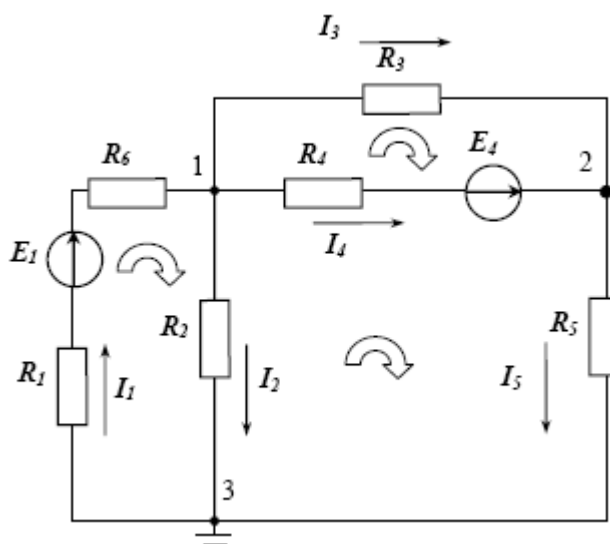


Рисунок 2.60 – Электрическая цепь

Для решения этой задачи необходимо задать условно положительные направления токов. Затем указать направления обхода контуров. После того как был определен порядок системы уравнений, записывается система уравнений, составленная на основе законов Кирхгофа. Расчет цепи постоянного тока в матричной форме и с помощью блока решений Given в системе *Mathcad* показан на рисунках 2.61 и 2.62 соответственно. При составлении системы уравнений следует придерживаться ряда принципов:

а) направления искомых токов целесообразно выбирать одинаковым с направлением ЭДС;

б) уравнения Кирхгофа записывать в виде, близком к матричному (номера столбцов должны совпадать с номерами токов ветвей; индексы токов нарастают слева направо; при отсутствии элемента ставится ноль).

Последующий переход к чисто матричной форме (матрицы  $A$  и  $B$ ) и его использование для решения системы линейных алгебраических уравнений очевиден и как показывает практика, выполняется безошибочно [18].

Известно из матричной алгебры, что вектор решения системы линейных уравнений определяется как  $X = A^{-1} \cdot B$ , где  $A^{-1}$  – инвертированная матрица коэффициентов системы, а  $B$  – вектор свободных членов системы уравнений,  $X$  – вектор решения.

```

Вводим исходные данные:
ORIGIN := 1
E1 := 30 В      E4 := 100 В      R1 := 2 Ом      R2 := 6 Ом      R3 := 2.5 Ом
R4 := 10 Ом     R5 := 4 Ом      R6 := 1 Ом
Запишем уравнения по законам Кирхгофа
I1-I2-I3-I4=0
I3+I4-I5=0
I1*(R1+R6)+I2*R2=E1
I3*R3-I4*R4=-E4
I4*R4+I5*R5-I2*R2=E4
В матричной форме уравнения выглядят так:
A := [ [ 1      -1  -1  -1  0 ]
      [ 0      0   1   1 -1 ]
      (R1 + R6) R2  0   0  0 ]
      [ 0      0   R3 -R4  0 ]
      [ 0     -R2  0   R4  R5 ]
B := [ 0 ]
      [ 0 ]
      E1 ]
      [-E4 ]
      [ E4 ]
Рассчитаем токи в цепи:
I := A^-1 * B
I = [ 6.667 ]
      [ 1.667 ]
      [-4 ]
      [ 9 ]
      [ 5 ]

```

Рисунок 2.61 – Пример расчета электрической цепи в Mathcad

Для решения системы уравнений с помощью блока решений Given необходимо задать блок уравнений, который имеет следующую структуру:  
а) начало блока (задается с помощью ключевого слова Given);



- б) тело блока (сюда входят все уравнения и ограничения);  
 в) конец блока (заканчивается с помощью выражения Find).

Перед началом блока решений обязательно задаются начальные условия, т.е. начальные значения искомых величин, т.к. решение находится с помощью итераций. При записи уравнений вместо строгого равенства используется знак приближенно равно  $\approx$ . Если поставить строгое равенство, то уравнения разрешены не будут. Как видно из примера расчета токи в цепи, полученные прямым решением уравнений и полученные с помощью блока решений Given равны.

```

Найдем токи в ветвях с помощью блока решений Given
Зададим начальные условия:
I1 := 2      I2 := 1      I3 := -1      I4 := 3      I5 := 2      Начало блока
Given
-I1 + I2 + I3 + I4 = 0
-I3 - I4 + I5 = 0
I1 · (R1 + R6) + R2 · I2 = E1      Тело блока
I3 · R3 - R4 · I4 = -E4
I4 · R4 + I5 · R5 - I2 · R2 = E4
Ig := Find(I1, I2, I3, I4, I5)
Конец блока      Получим результат:      Ig =  $\begin{pmatrix} 6.667 \\ 1.667 \\ -4 \\ 9 \\ 5 \end{pmatrix}$       I =  $\begin{pmatrix} 6.667 \\ 1.667 \\ -4 \\ 9 \\ 5 \end{pmatrix}$ 

Выполним проверку баланса мощности:

Pist := E1 · I1 + E4 · I4      Pist = 1100      Вт

Ppot := (I1)2 · (R1 + R6) + (I2)2 · R2 + (I3)2 · R3 + (I4)2 · R4 + (I5)2 · R5

Ppot = 1100      Вт
I1 - I2 - I3 - I4 = 0
I3 + I4 - I5 = 0
I1 · (R1 + R6) + I2 · R2 = 30
I3 · R3 - I4 · R4 = -100
-I2 · R2 + I4 · R4 + I5 · R5 = 100
Найденные токи запишем в файл:
WRITEPRN ("Otvet.prn" ) := I      (- форма записи для MCAD под Windows)
  
```

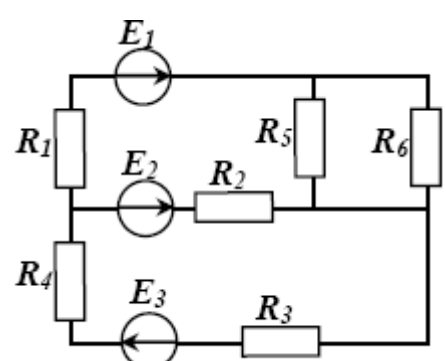
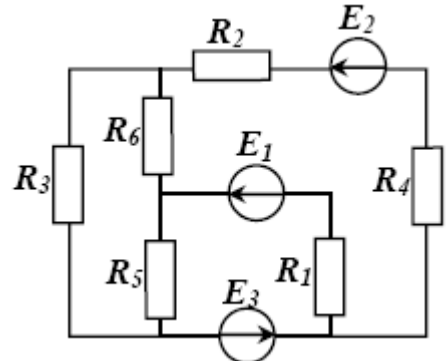
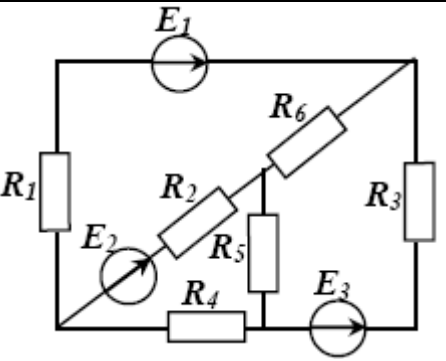
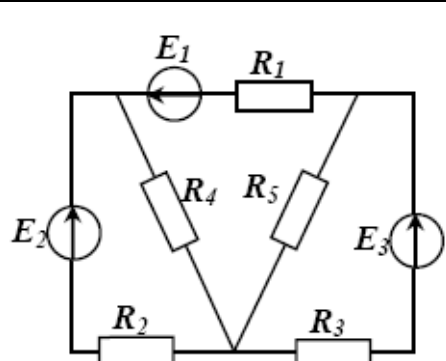
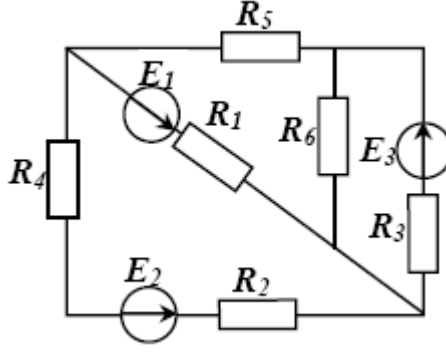
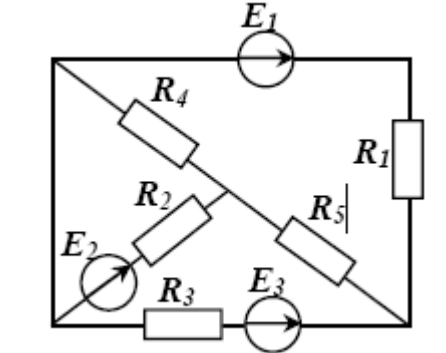
Рисунок 2.62 – Поиск токов и проверка баланса мощностей

## Индивидуальное задание

Приведена схема электрической цепи постоянного тока. Составить систему уравнений с использованием законов Кирхгофа. Найти токи в ветвях цепи. Составить баланс мощности. Выполнить проверку решения. Задачу решить двумя методами: в матричной форме; с помощью блока решений Given.

Данные, необходимые для расчета линейной цепи постоянного тока находятся в таблицах 2.31 и 2.32.

Таблица 2.31 – Варианты схемы электрической цепи

№	Схема	№	Схема
1		4	
2		5	
3		6	

№	Схема	№	Схема
7		8	
9		10	
11		12	
13		14	
15		16	

Продолжение таблицы 2.31.

№	Схема	№	Схема
17		18	
19		20	
21		22	
23		24	

Продолжение таблицы 2.31.

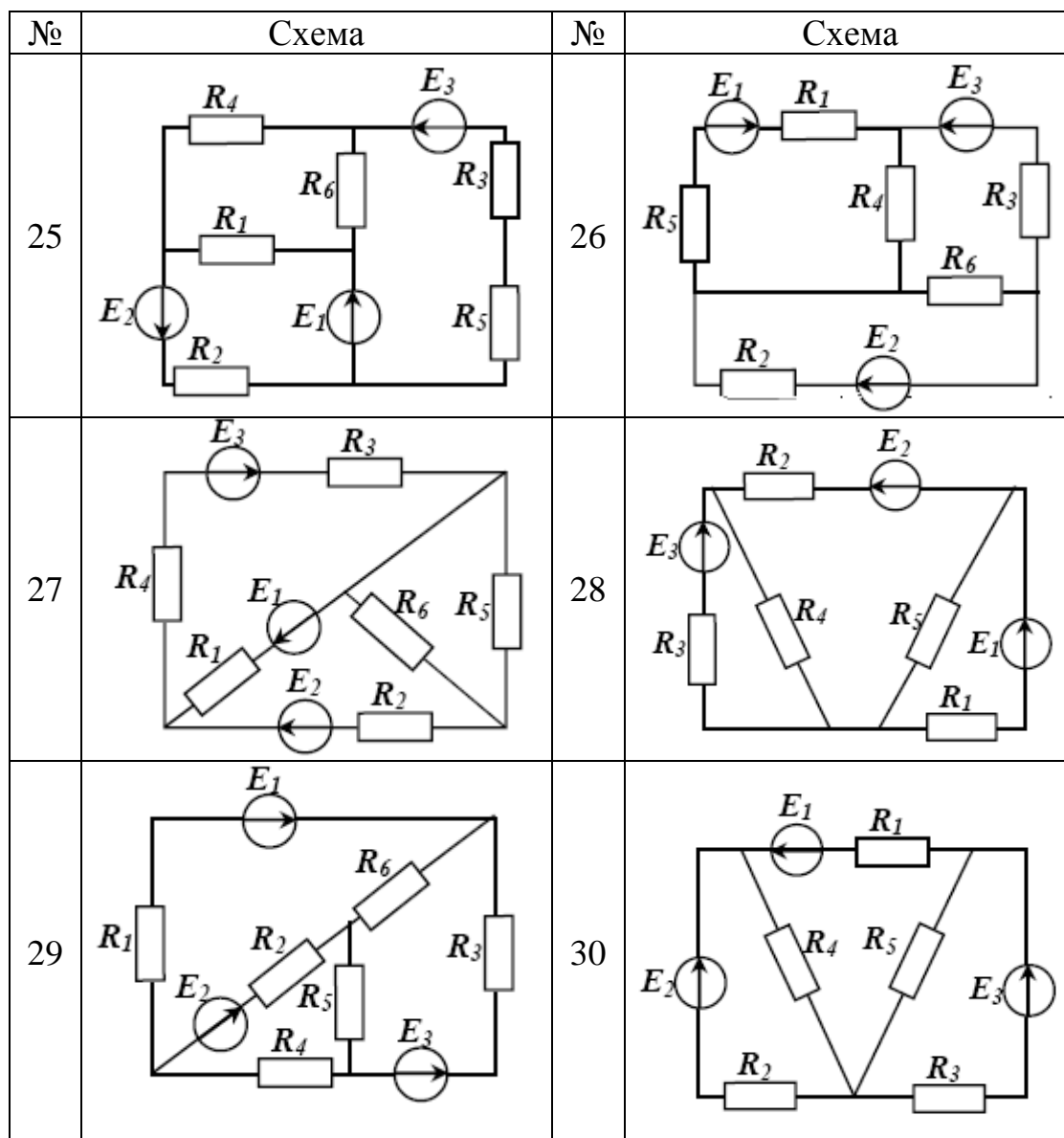


Таблица 2.32 – Параметры источников энергии и резисторов

№	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>
	В			Ом					
1	50	30	20	4	2	8	9	15	7
2	120	10	15	6	3	7	8	20	3
3	50	75	30	2	8	10	5	12	5
4	100	50	25	8	10	23	2	11	9
5	10	20	80	10	4	20	8	3	11
6	70	60	100	4	15	12	3	10	20
7	40	20	115	15	20	11	2	7	12
8	125	10	95	20	35	10	12	5	19
9	110	15	120	14	12	6	4	9	15
10	60	65	40	10	5	3	2	8	12

Продолжение таблицы 2.32.

№	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>	R <sub>1</sub>
	B			Oм					
11	55	15	65	3	6	3	9	22	18
12	34	25	50	5	8	10	11	8	13
13	95	35	40	10	23	2	8	25	20
14	75	15	90	23	8	11	6	11	21
15	85	25	46	12	10	12	23	4	14
16	65	22	55	11	2	20	2	9	13
17	115	32	25	6	7	8	15	14	9
18	110	45	15	3	5	15	34	18	7
19	120	54	60	22	9	2	6	25	3
20	130	23	30	15	8	25	9	10	23
21	10	20	80	10	4	20	8	3	11
22	70	60	100	4	15	12	3	10	20
23	40	20	115	15	20	11	2	7	12
24	125	10	95	20	35	10	12	5	19
25	110	15	120	14	12	6	4	9	15
26	60	65	40	10	5	3	2	8	12
27	120	10	15	6	3	7	8	20	3
28	50	75	30	2	8	10	5	12	5
29	100	50	25	8	10	23	2	11	9
30	85	25	46	12	10	12	23	4	14

### 2.2.10 Задание на лабораторную работу № 33

**Тема:** Исследование частотных характеристик системы автоматического регулирования.

**Цель работы:** провести исследование частотных характеристик линейной САР и построить передаточную функцию замкнутой системы с помощью математического пакета Mathcad.

#### **Теоретические сведения**

Частотная передаточная функция получается из обычной заменой оператора Лапласа  $p$  или  $s$  на комплексную частоту  $j\omega$ , т.е. в результате перехода от изображения Лапласа к изображению Фурье. Дифференциальное уравнение движения системы связывает входной и выходной сигналы (т.е. функции времени). Передаточная функция связывает изображения Лапласа тех же сигналов, а частотная передаточная функция связывает их спектры.

Амплитудно-фазовая характеристика (годограф Найквиста) – графическое отображение для всех частот спектра отношений выходного сигнала системы автоматического регулирования ко входному, представленных в комплексной форме. Величина отрезка от начала координат до каждой точки годографа показывает во сколько раз на данной частоте выходной сигнал больше входного, а сдвиг фазы между сигналами определяется углом до упомянутого отрезка.

От АФХ порождаются все другие частотные зависимости:

а)  $U(\omega)$  – четная (для замкнутых систем регулирования( $\omega$ ));

б)  $V(\omega)$  – нечетная;

в)  $A(\omega)$  – четная (АЧХ);

г)  $\varphi(\omega)$  – нечетная (ФЧХ);

д) ЛАЧХ & ЛФЧХ – используются наиболее часто.

Построение ЛАЧХ & ЛФЧХ производится по выражениям:

$$L(\omega) = 20 \lg |W(j\omega)| = 20 \lg A(\omega), \text{ [дБ]}; \quad \varphi(\omega) = \arg(W(j\omega)), \text{ [рад]}.$$

Числитель и знаменатель передаточной функции системы автоматического регулирования могут быть представлены либо в виде отношения полиномов:

$$W(s) = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{c_0 s^n + c_1 s^{n-1} + \dots + c_{n-1} s + c_n}, \quad \text{где: } m < n.$$

либо в виде отношения их разложений на элементарные множители:

$$W(s) = \frac{K_r}{s^r} \cdot \frac{\prod_{j=1}^m (1 + T_j s)}{\prod_{i=1}^{n-r-2d} (1 + T_i s)} \cdot \frac{1}{\prod_{k=1}^{2d} (1 + 2\zeta_k T_k s + T_k^2 s^2)}.$$

Методы ЛАЧХ и ФЧХ используется для решения задач синтеза систем автоматического регулирования, предполагающие поиск рациональной структуры системы или установление оптимальных величин параметров отдельных звеньев.

Пример построения характеристик и кривой переходного процесса приведены на рисунках 2.63 и 2.64 соответственно.

### ***Индивидуальное задание***

Согласно вариантам индивидуального задания (таблицы 2.33 и 2.34) задать передаточную функцию (можно задать передаточные функции отдельных узлов САР (рисунок 2.65), а потом объединить их в одну с использованием соответствующих правил и при необходимости средств символьной математики пакета). Задать диапазон частот, в котором будет исследоваться система. Получить логарифмическую амплитудно-частотную и фазо-частотную характеристики САР. Получить переходную характеристику замкнутой системы. Структурную схему построить в виде

математической модели в среде Simulink пакета Matlab. Сделать вывод о свойствах системы автоматического регулирования по виду переходной характеристики и кривой переходного процесса выходной координаты (по виду осциллограмм).

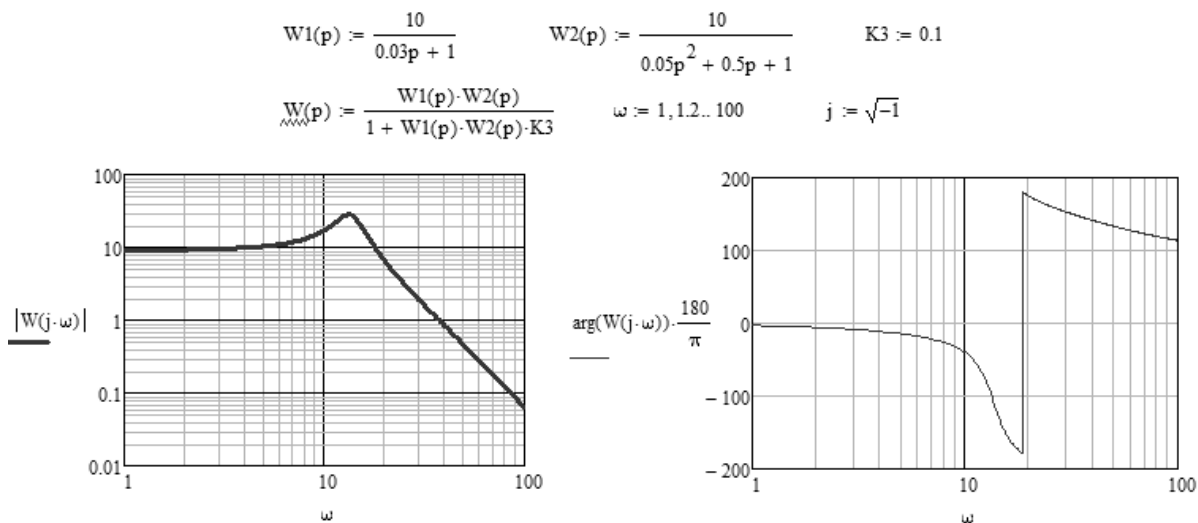


Рисунок 2.63 – Построение характеристик

$$\underline{\underline{H}}(s) := \frac{W1(s) \cdot W2(s)}{(1 + W1(s) \cdot W2(s) \cdot K3) \cdot s}$$

$$\underline{\underline{H}}(s) := \frac{W(s)}{s}$$

$$H(s) \begin{cases} \text{invlaplace, s} \\ \text{simplify} \rightarrow 9.091 - 1.100 \cdot \exp(-39.09 \cdot t) - 7.991 \cdot \exp(-2.120 \cdot t) \cdot \cos(13.53 \cdot t) - 4.430 \cdot \exp(-2.120 \cdot t) \cdot \sin(13.53 \cdot t) \\ \text{float, 2} \end{cases}$$

$$h(t) := 9.091 - 1.100 \cdot \exp(-39.09 \cdot t) - 7.991 \cdot \exp(-2.120 \cdot t) \cdot \cos(13.53 \cdot t) - 4.430 \cdot \exp(-2.120 \cdot t) \cdot \sin(13.53 \cdot t)$$

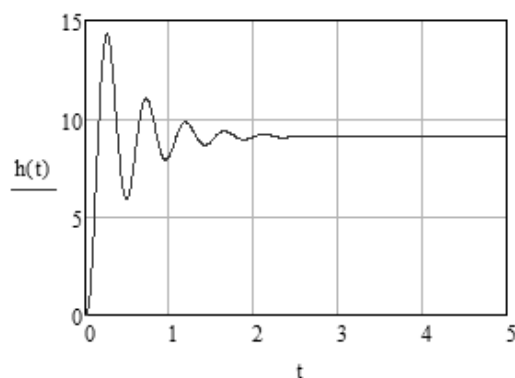


Рисунок 2.64 – Построение кривой переходного процесса



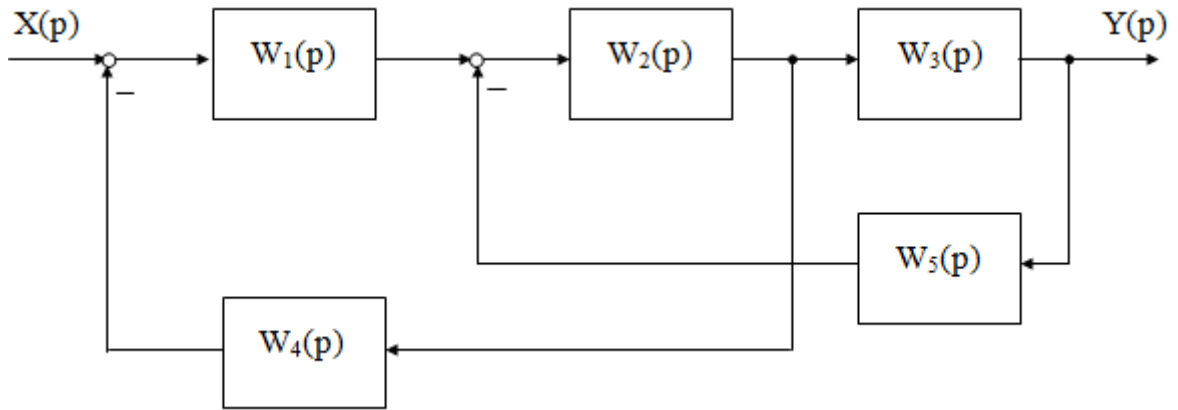


Рисунок 2.65 – Структурная схема САР

Таблица 2.33 – Передаточные функции САР

№	$W_1(p)$	$W_2(p)$	$W_3(p)$	$W_4(p)$	$W_5(p)$
1	$K_1$	$\frac{K_2}{T_2 p + 1}$	$\frac{1}{T_3 p}$	$K_4$	$K_5 p$
2	$\frac{K_1}{T_1 p + 1}$	$\frac{1}{T_2 p + 1}$	$\frac{K_3}{p}$	$K_4 p$	$K_5$
3	$\frac{T_2 p + 1}{T_1 p + 1}$	$K_2$	$\frac{K_3}{T_3 p}$	$\frac{K_4}{p}$	$K_5 p$
4	$T_2 p + 1$	$\frac{K_2}{p}$	$K_3$	$\frac{1}{T_4 p}$	$K_5$
5	$K_1 p$	$\frac{K_2}{T_2 p}$	$\frac{1}{T_3 p}$	$\frac{K_4}{T_4 p}$	$K_5$
6	$\frac{K_1}{p}$	$\frac{1}{T_2 p}$	$\frac{K_3}{T_3 p}$	$K_4$	$K_5 p$
7	$K_1$	$\frac{1}{p}$	$\frac{K_3}{p}$	$K_4 p$	$K_5$
8	$\frac{K_1}{T_1 p}$	$\frac{K_2}{T_2 p + 1}$	$K_3$	$\frac{K_4}{p}$	$K_5$
9	$\frac{K_1}{T_1 p + 1}$	$K_2$	$\frac{1}{p}$	$\frac{1}{T_4 p}$	$K_5 p$
10	$\frac{1}{T_1 p}$	$\frac{T_1 p + 1}{K_2 T_2 p}$	$\frac{1}{K_3 p}$	$\frac{K_4}{T_4 p}$	$K_5$
11	$K_1$	$\frac{1}{T_2 p + 1}$	$\frac{1}{T_3 p}$	$K_4 p$	$K_5$
12	$\frac{K_1}{T_1 p + 1}$	$K_2$	$\frac{K_3}{p}$	$\frac{K_4}{p}$	$K_5 p$
13	$\frac{T_2 p + 1}{T_1 p + 1}$	$\frac{K_2}{p}$	$\frac{K_3}{T_3 p}$	$\frac{1}{T_4 p}$	$K_5$

Продолжение таблицы 2.33.

14	$T_{2p+1}$	$\frac{K_2}{T_{2p}}$	$K_3$	$\frac{K_4}{T_{4p}}$	$K_5$
15	$K_{1p}$	$\frac{1}{T_{2p}}$	$\frac{1}{T_{3p}}$	$K_4$	$K_{5p}$
16	$\frac{K_1}{p}$	$\frac{1}{p}$	$\frac{K_3}{T_{3p}}$	$K_{4p}$	$K_5$
17	$K_1$	$\frac{K_2}{T_{2p+1}}$	$\frac{K_3}{p}$	$\frac{K_4}{p}$	$K_5$
18	$\frac{K_1}{T_{1p}}$	$K_2$	$K_3$	$\frac{1}{T_{4p}}$	$K_{5p}$
19	$\frac{K_1}{T_{1p+1}}$	$\frac{T_{1p+1}}{K_2 T_{2p}}$	$\frac{1}{p}$	$\frac{K_4}{T_{4p}}$	$K_5$
20	$\frac{1}{T_{1p}}$	$\frac{K_2}{T_{2p+1}}$	$\frac{1}{K_3 p}$	$K_{4p}$	$K_5$
21	$\frac{T_{2p+1}}{T_{1p+1}}$	$\frac{K_2}{T_{2p+1}}$	$\frac{K_3}{T_{3p}}$	$\frac{1}{T_{4p}}$	$K_5$
22	$T_{2p+1}$	$\frac{1}{T_{2p+1}}$	$K_3$	$\frac{K_4}{T_{4p}}$	$K_{5p}$
23	$K_{1p}$	$K_2$	$\frac{1}{T_{3p}}$	$K_4$	$K_5$
24	$\frac{K_1}{p}$	$\frac{K_2}{p}$	$\frac{K_3}{T_{3p}}$	$K_{4p}$	$K_5$
25	$K_1$	$\frac{K_2}{T_{2p}}$	$\frac{K_3}{p}$	$\frac{K_4}{p}$	$K_5$
26	$\frac{K_1}{T_{1p}}$	$\frac{1}{T_{2p}}$	$K_3$	$\frac{1}{T_{4p}}$	$K_{5p}$
27	$\frac{K_1}{T_{1p+1}}$	$\frac{1}{p}$	$\frac{1}{p}$	$\frac{K_4}{T_{4p}}$	$K_{5p}$
28	$\frac{1}{T_{1p}}$	$\frac{K_2}{T_{2p+1}}$	$\frac{1}{K_3 p}$	$\frac{1}{T_{4p}}$	$K_5$
29	$K_1$	$K_2$	$\frac{K_3}{T_{3p}}$	$K_4$	$K_{5p}$
30	$\frac{K_1}{T_{1p}}$	$\frac{1}{T_{2p}}$	$\frac{K_3}{p}$	$K_{4p}$	$K_5$

Таблица 2.34 – Параметры коэффициентов и постоянных времени

№	$K_1$	$K_2$	$K_3$	$K_4$	$K_5$	$T_1$	$T_2$	$T_3$	$T_4$
1	2	50	0,1	6	4	0,02	0,001	1	0,1
2	5	20	0,2	8	6	0,05	0,005	2	0,2
3	10	10	0,4	10	8	0,08	0,01	3	0,3
4	20	5	0,8	12	10	0,12	0,02	4	0,4
5	50	2	1,2	20	12	0,2	0,05	5	0,5
6	2	50	0,1	6	4	0,02	0,001	1	0,1
7	5	20	0,2	8	6	0,05	0,005	2	0,2
8	10	10	0,4	10	8	0,08	0,01	3	0,3
9	20	5	0,8	12	10	0,12	0,02	4	0,4
10	50	2	1,2	20	12	0,2	0,05	5	0,5
11	5	20	0,2	8	6	0,05	0,005	2	0,2
12	10	10	0,4	10	8	0,08	0,01	3	0,3
13	20	5	0,8	12	10	0,12	0,02	4	0,4
14	50	2	1,2	20	12	0,2	0,05	5	0,5
15	2	50	0,1	6	4	0,02	0,001	1	0,1
16	5	20	0,2	8	6	0,05	0,005	2	0,2
17	10	10	0,4	10	8	0,08	0,01	3	0,3
18	20	5	0,8	12	10	0,12	0,02	4	0,4
19	50	2	1,2	20	12	0,2	0,05	5	0,5
20	10	10	0,4	10	8	0,08	0,01	3	0,3
21	20	5	0,8	12	10	0,12	0,02	4	0,4
22	50	2	1,2	20	12	0,2	0,05	5	0,5
23	2	50	0,1	6	4	0,02	0,001	1	0,1
24	5	20	0,2	8	6	0,05	0,005	2	0,2
25	10	10	0,4	10	8	0,08	0,01	3	0,3
26	20	5	0,8	12	10	0,12	0,02	4	0,4
27	50	2	1,2	20	12	0,2	0,05	5	0,5
28	20	5	0,8	12	10	0,12	0,02	4	0,4
29	50	2	1,2	20	12	0,2	0,05	5	0,5
30	2	50	0,1	6	4	0,02	0,001	1	0,1

### Контрольные вопросы

- 1 Функциональные схемы типовых динамических звеньев САУ.
- 2 Особенности преобразования структурных схем.
- 3 Понятие, виды, особенности передаточных функций САУ.
- 4 Понятие, виды, особенности частотных характеристик САУ.
- 5 Порядок и правила построения ЛАЧХ и ФЧХ.

## 2.2.11 Задание на самостоятельную работу № 4

**Тема:** Построение поверхностей в сферической и цилиндрической системах координат.

**Цель работы:** приобрести навыки построения графических объектов в разных системах координат в пакете Matlab.

### *Теоретические сведения*

Связь между декартовой системой координат и сферической определяется формулами:  $Y = R \cos(\alpha) \sin(\varphi)$ ;  $X = R \cos(\alpha) \cos(\varphi)$ ;  $Z = R \sin(\alpha)$  (рисунок 2.66). Если существует функция  $F(X,Y,Z) = 0$ , связывающая координаты  $X, Y, Z$ , то возможно определить взаимосвязь между переменными  $\alpha, \varphi, R$  (рисунок 2.67). Угол  $\alpha$  изменяется в пределах от 0 до  $\pi$ , угол  $\varphi$  от 0 до  $2\pi$  [21].

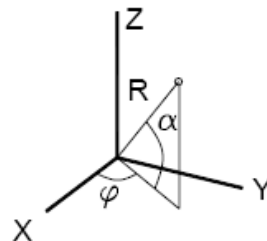


Рисунок 2.66 – Сферическая система координат

$$\begin{aligned} M &:= 25 \quad N := 50 \quad i := 0..M \quad j := 0..N \\ \alpha_i &:= \frac{\pi \cdot i}{M} \quad \phi_j := \frac{2 \cdot \pi \cdot j}{N} \\ X_{i,j} &:= \tan(\alpha_i) \cdot \cos(\phi_j) \quad Y_{i,j} := \tan(\alpha_i) \cdot \sin(\phi_j) \quad Z_{i,j} := \tan(\alpha_i)^2 \end{aligned}$$

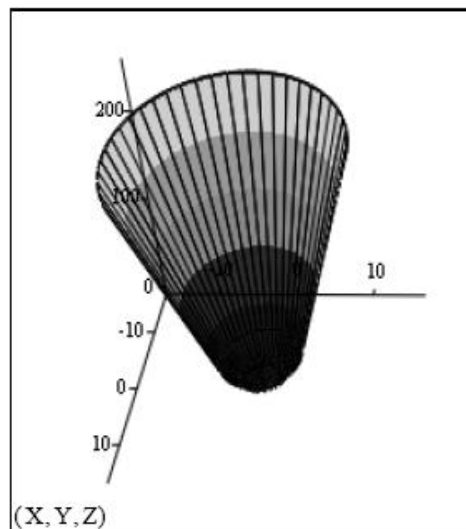


Рисунок 2.67 – Построение поверхности

Связь между декартовой и цилиндрической системами координат выражается формулами:  $Y = r\sin(\varphi)$ ;  $X = r\cos(\varphi)$ ;  $Z = Z$  (рисунок 2.68).

$M := 25$   $N := 50$   $i := 0..M$   $j := 0..N$   $r := 5$

$$\phi_{i,j} := \frac{2 \cdot \pi \cdot j}{N} \quad Z_{i,j} := r \cdot \frac{i}{M} \quad R_{i,j} := \sqrt{Z_{i,j}}$$

$$X_{i,j} := R_{i,j} \cdot \cos(\phi_{i,j}) \quad Y_{i,j} := R_{i,j} \cdot \sin(\phi_{i,j})$$

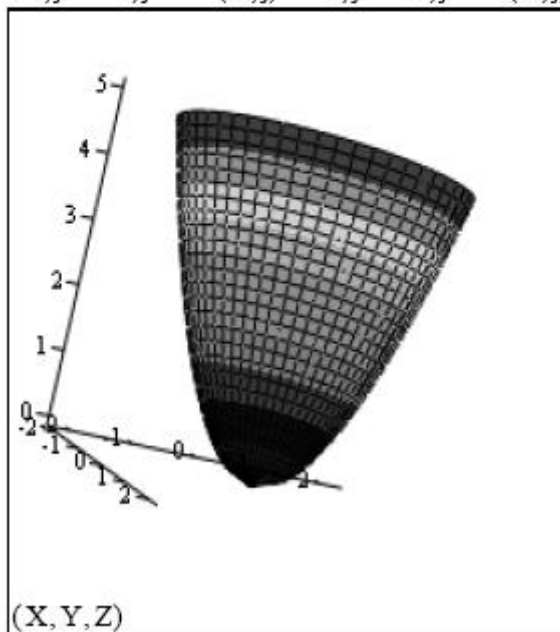


Рисунок 2.68 – Цилиндрическая система координат

### ***Индивидуальное задание***

Варианты заданий на построение поверхностей в сферической и цилиндрической системах координат:

- 1 Построить сферу радиуса  $R=4$ .
- 2 Построить в сферической системе координат поверхность, модуль радиус-вектора точек которых изменяется по закону  $R = \cos(\alpha)$ .
- 3 Построить в сферической системе координат конус с углом раствора 45 градусов.
4. Построить в цилиндрической системе координат поверхность цилиндра радиуса  $R=4$  и высотой  $h=10$ .
- 5 Построить в цилиндрической системе координат поверхность конуса с углом раствора 45 градусов и высотой  $h=10$ .
- 6 Построить поверхность, которая получается в результате вращения отрезка вокруг оси  $OZ$ . Координаты концов отрезка:  $A(0,0,0)$  и  $B(1,0,2)$ .
- 7 Построить поверхность, которая получается в результате вращения отрезка вокруг оси  $OX$ . Координаты концов отрезка:  $A(0,0,0)$  и  $B(1,0,2)$ .

8 Построить поверхность, которая получается в результате вращения отрезка вокруг оси OZ. Координаты концов отрезка: A(0,0,0) и B(1,0,2).

9 Построить в цилиндрической системе координат поверхность  $Z(X, Y) = e^{-X+Y}$ .

10 Построить в сферической системе координат эллипсоид с полуосями a=1, b=2, c=3.

## 2.2.12 Задание на самостоятельную работу № 5

**Тема:** Особенности программирования и реализация циклических процессов в Mathcad.

**Цель работы:** изучить операторы программирования и приобрести навыки по созданию программ с линейными и разветвленными вычислительными процессами.

### *Теоретические сведения*

Программные операторы сосредоточены в наборной панели программных элементов (рисунок 2.69).

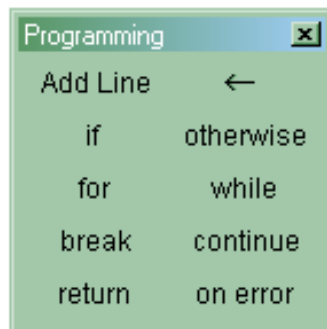


Рисунок 2.69 – Панель операторов программирования

Набор программных элементов весьма ограничен [19]:

а) Add Line – создает и при необходимости расширяет жирную вертикальную линию, справа от которой в шаблонах задается запись программного блока;

б) < – символ локального присваивания (в теле программного модуля);

в) if – оператор условного выражения;

г) for – оператор задания цикла с фиксированным числом продолжений;

д) while – оператор задания цикла типа "пока" (цикл выполняется, пока выполняется некоторое условие);

е) otherwise – оператор иного выбора (обычно применяется с оператором if );

ж) break – оператор прерывания;

з) continue – оператор продолжения;

и) return – оператор-функция возврата;

к) on error – оператор обработки ошибок.

Пример использования программных операторов приведен далее.

Оператор Add Line выполняет функции расширения программного блока. Расширение фиксируется удлинением вертикальной черты программных блоков или их древовидным расширением.

Оператор внутреннего присваивания «<>» выполняет функции внутреннего локального присваивания.

Оператор создания условных выражений if задается в виде:

### Выражение if Условие

Если условие выполняется, то возвращается значение выражения.

Совместно с этим оператором часто используются операторы прерывания break и оператор иного выбора otherwise. Оператор for служит для организации циклов с заданным числом повторений. Он записывается в виде:

for Var . Nmin..Nmax.

Эта запись означает, что если переменная Var меняется с шагом +1 от значения Nmin до Nmax, то выражение, помещенное в шаблон, будет выполняться. Переменную счетчика Var можно использовать в выражениях программы.

Оператор otherwise (иначе) обычно используется совместно с оператором if. Его использование поясняет следующая программная конструкция:

$f(x) := 1 \text{ if } x > 0$       Возвращает 1, если  $x > 0$   
-1 otherwise      Возвращает -1 во всех иных случаях

### *Индивидуальное задание*

С помощью операций программирования средствами Mathcad с учетом варианта (таблица 2.35):

а) определить сумму ряда (с помощью оператора for);

б) определить значение функции с учетом граничных условий и построить график функции (с помощью оператора if).

Составить программу для поиска суммы ряда с помощью итерационной обработки данных (таблица 2.36).

$$y = \sum \frac{f_1}{f_2}, \text{ где } a \leq x \leq b, \Delta x = c.$$

Составить программу для обработки массива согласно индивидуальному заданию (таблица 2.37).

Таблица 2.35 – Варианты индивидуальных заданий

№	$\Sigma$	№	Функция
1	$\sum_{i=1}^{100} \frac{1}{i^2}$	1	$y(x) = \begin{cases} -x & \text{при } x < 0 \\ -x^2 & \text{при } x \geq 0 \end{cases}$
2	$\sum_{i=1}^{50} \frac{1}{i^3}$	2	$y(x) = \begin{cases} \frac{1}{x^2} & \text{при } x < -1 \\ x^2 & \text{при } -1 \leq x \leq 2 \\ 4 & \text{при } x > 2 \end{cases}$
3	$\sum_{i=1}^{128} \frac{1}{i^3}$	3	$y(x) = \begin{cases} x^2 & \text{при } x < 0 \\ -x^4 & \text{при } x \geq 0 \end{cases}$
4	$\sum_{i=1}^{120} \frac{1}{i}$	4	$y(x) = \begin{cases} -x-1 & \text{при } x < -1 \\ x+1 & \text{при } -1 \leq x \leq -0 \\ -x+1 & \text{при } -0 \leq x \leq 1 \\ x-1 & \text{при } x > 1 \end{cases}$
5	$\sum_{i=1}^9 \frac{1}{i^5}$	5	$y(x) = \begin{cases} -3x-1 & \text{при } x \leq -2 \\ -x+6 & \text{при } -2 \leq x \leq 1 \\ x+4 & \text{при } 1 \leq x \leq 3 \\ 3x-2 & \text{при } x > 3 \end{cases}$
6	$\sum_{i=1}^{39} \frac{1}{(2i+1)^2}$	6	$y(x) = \begin{cases} -1 & \text{при } x < -1 \\ x & \text{при } -1 \leq x \leq 1 \\ 1 & \text{при } x > 1 \end{cases}$
7	$\sum_{i=1}^{12} \frac{(-1)^{i+1}}{i(i+1)}$	7	$y(x) = \begin{cases} -x^2 & \text{при } x < 0 \\ -x^3 & \text{при } x \geq 0 \end{cases}$
8	$\sum_{i=1}^{25} \frac{(-1)^i}{i(2i+1)}$	8	$y(x) = \begin{cases} -x & \text{при } x < -1 \\ 1 & \text{при } -1 \leq x \leq 1 \\ x & \text{при } x > 1 \end{cases}$



Продолжение таблицы 2.35.

№	$\Sigma$	№	Функция
9	$\sum_{i=1}^{90} \frac{(-1)^{i+1}}{i(i+1)(i+2)}$	9	$y(x) = \begin{cases} -\frac{1}{x^2} + 1 & \text{при } x < -1 \\ x^2 + 1 & \text{при } -1 \leq x \leq 2 \\ 5 & \text{при } x > 2 \end{cases}$
10	$\sum_{i=1}^{100} \frac{1}{i^2}$	10	$y(x) = \begin{cases} -x & \text{при } x < 0 \\ -x^2 & \text{при } x \geq 0 \end{cases}$
11	$\sum_{i=1}^5 \frac{(-1)^{i+1}}{4^i + 5^{i+2}}$	11	$y(x) = \begin{cases} x+1 & \text{при } x < -1 \\ -x-1 & \text{при } -1 \leq x \leq -0 \\ x-1 & \text{при } -0 \leq x \leq 1 \\ -x+1 & \text{при } x > 1 \end{cases}$
12	$\sum_{i=1}^{120} \frac{1}{i}$	12	$y(x) = \begin{cases} -1 & \text{при } x < -1 \\ x & \text{при } -1 \leq x \leq 1 \\ 1 & \text{при } x > 1 \end{cases}$
13	$\sum_{i=1}^{128} \frac{1}{i^3}$	13	$y(x) = \begin{cases} -3x-1 & \text{при } x \leq -2 \\ -x+6 & \text{при } -2 \leq x \leq 1 \\ x+4 & \text{при } 1 \leq x \leq 3 \\ 3x-2 & \text{при } x > 3 \end{cases}$
14	$\sum_{i=1}^{39} \frac{1}{(2i+1)^2}$	14	$y(x) = \begin{cases} x^2 & \text{при } x < 0 \\ -x^4 & \text{при } x \geq 0 \end{cases}$
15	$\sum_{i=1}^{12} \frac{(-1)^{i+1}}{i(i+1)}$	15	$y(x) = \begin{cases} -x-1 & \text{при } x < -1 \\ x+1 & \text{при } -1 \leq x \leq -0 \\ -x+1 & \text{при } -0 \leq x \leq 1 \\ x-1 & \text{при } x > 1 \end{cases}$

Таблица 2.36 – Варианты индивидуальных заданий

№	$f_1$	$f_2$	$a$	$b$	$c$
1	$3x-1$	$e^{-1/x} + x/(x+1)$	3	5	0,5
2	$X^3-3x^2$	$x^4+2x^2+3$	1	3	0,2
3	$e^{-x}+4x$	$\sqrt{(1+x^3+4x^2)}$	0,6	4,2	0,3
4	$\text{Sin}^2(x+4x^3)$	$(x+2x^3) \cdot \sqrt{x}$	0,5	4,8	0,2
5	$X \text{sin} x^3 - \ln 2x$	$\text{arctg} x / 4 + e^{-x+2}$	2	6,3	0,4
6	$X^4 - \cos x$	$\text{tg} x + 2x$	1	5	0,5

Продолжение таблицы 2.36.

№	$f_1$	$f_2$	a	b	c
7	$2x + \sin^2 x$	$\sqrt{(2x + \ln x)}$	5	8	0,3
8	$\ln(4x+8)$	$e^{-x} + \sin 2x$	1	4	0,2
9	$x^3 \ln(2x)$	$4x^2 + 6x^3 - 2$	0,5	6	0,3
10	$x^2 + \sin 3x$	$\cos 3x + e^{-2x}$	-2	3	0,4
11	$xe^{-x}$	$\sin 4x + x^3$	1,5	5	0,3
12	$\sqrt{x^2 + 1}$	$\arctg x / 5 + 2x$	0,6	4	0,2
13	$x^2 / (3x+2)$	$\sin 2(\pi x + 1)$	0,5	5,2	0,3
14	$\sqrt{2 + x^3}$	$3x / (x-2)$	1,2	6,3	0,4
15	$x^3 x + 1 + 8x$	$ x-8  + \sin x$	4	7,5	0,3
16	$x^4 + ex + 3$	$x \arctg(x/3)$	2	6,4	0,2
17	$\ln 2(x+4)$	$\sin 3(x/5)$	1	6,8	0,3
18	$ex - 2 + x^3$	$x - \ln x-1 $	0	4	0,4
19	$2\cos(x+3)$	$4x^2 / (3+x^3)$	2	5	0,3
20	$\sqrt{1 + x^4}$	$\tg 2(x+4) - e^{-x}$	1	6	0,4
21	$3 + 2\sin 2(x-3)$	$4 + x/10$	2	7	0,5
22	$\ln(1(1+2x))$	$\sin 2(4x+1)$	1,5	6,8	0,4
23	$\sqrt{ x } + e^{-x}$	$5 \arctg(4x)$	2	7	0,5
24	$\arcsin(x+2)$	$3(x-4)/(x^2+1)$	3	8	0,2
25	$e x+2 $	$\ln 2(x+4)$	-2	6	0,3
26	$(4-x)\cos 2x$	$\sqrt{ x+1 } + e^{-3x}$	1	7	0,4
27	$\sqrt{ x  + 2^x}$	$\sin x^4 - 4$	-2	5	0,2
28	$2x + 4 + \cos 2x$	$\ln x+8 $	-4	2	0,5
29	$(x+2)/\sin 3x$	$\sqrt{x + \tg^2 x}$	1	4	0,3
30	$ex + 3 + 4x^2$	$\arcsin x^3$	2	5	0,2

Таблица 2.37 – Варианты индивидуальных заданий

№	Условие задачи
1	Найти сумму четных чисел массива
2	Вычислить произведение отрицательных чисел массива
3	Определить количество нечетных чисел массива
4	Найти сумму отрицательных чисел массива
5	Определить количество отрицательных чисел массива
6	Вычислить произведение положительных чисел массива
7	Найти сумму положительных чисел массива
8	Определить количество четных чисел массива
9	Вычислить произведение четных чисел массива
10	Найти сумму нечетных чисел массива

*Продолжение таблицы 2.37.*

№	Условие задачи
11	Определить количество кратных 3 чисел массива
12	Вычислить произведение нечетных чисел массива
13	Найти сумму кратных 3 чисел массива
14	Определить количество не кратных 3 чисел массива
15	Вычислить произведение кратных 3 чисел массива
16	Найти сумму не кратных 3 чисел массива
17	Определить количество кратных 4 чисел массива
18	Вычислить произведение не кратных 3 чисел массива
19	Найти сумму кратных 4 чисел массива
20	Определить количество не кратных 4 чисел массива
21	Вычислить произведение кратных 4 чисел массива
22	Найти сумму не кратных 4 чисел массива
23	Определить количество кратных 5 чисел массива
24	Вычислить произведение не кратных 4 чисел массива
25	Найти сумму кратных 5 чисел массива
26	Вычислить среднее арифметическое положительных четных элементов массива
27	Найти среднее геометрическое отрицательных нечетных элементов массива
28	Найти произведение отрицательных не кратных пяти элементов массива
29	Найти среднее арифметическое элементов массива, находящихся в интервале $[-10,20]$
30	Найти среднее геометрическое элементов массива, находящихся в интервале $[5,20]$

## ***2.8 Моделирование структурных схем (VisSim)***

Программа VisSim предназначена для построения, исследования и оптимизации виртуальных моделей физических и технических объектов, в том числе и систем управления. VisSim – это аббревиатура выражения Visual Simulator – визуальная, воспринимаемая зрением, среда и средство.

Программа VisSim (рисунок 2.70) разработана и развивается компанией Visual Solutions (USA). Эта программа – мощное, удобное в использовании, компактное и эффективное средство моделирования физических и технических объектов, систем и их элементов [22].

Программа предоставляет разработчику развитый графический интерфейс, используя который он создает модель из виртуальных элементов с некоторой степенью условности так же, как если бы он строил реальную систему из настоящих элементов. Это позволяет создавать, а

затем исследовать и оптимизировать модели систем широкого диапазона сложности.

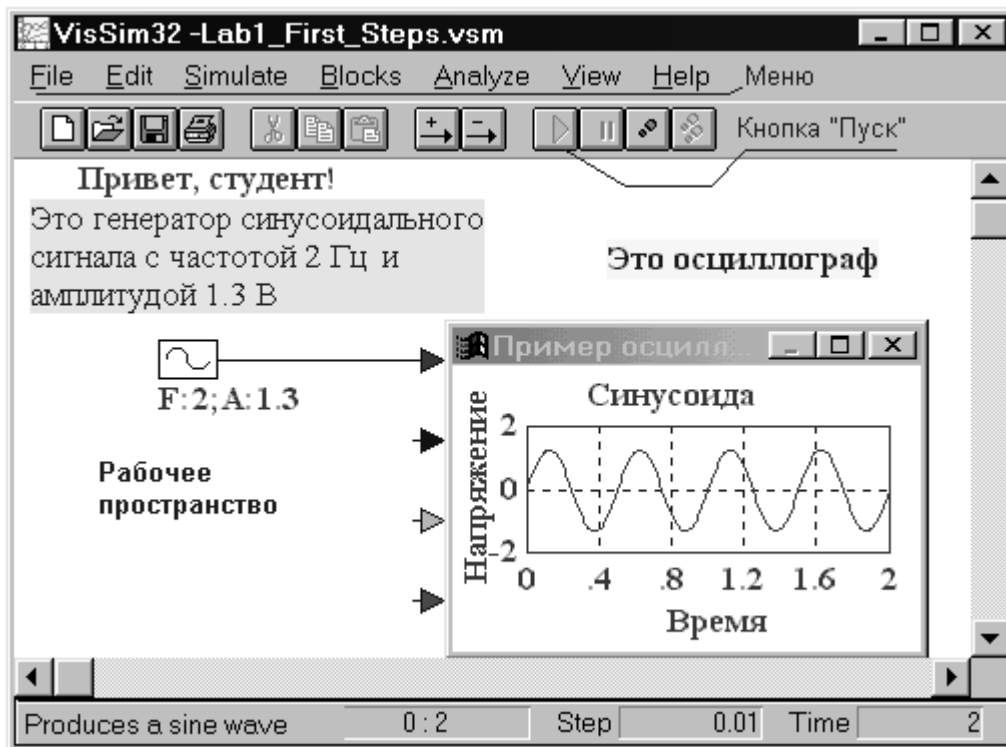


Рисунок 2.70 – Общий вид окна VisSim

При описании и последующем построении модели в среде VisSim нет необходимости записывать и решать дифференциальные уравнения, программа это сделает сама по предложенной ей исследователем структуре системы и параметрам ее элементов. Результаты решения выводятся в наглядной графической форме. Поэтому программой могут пользоваться и те, кто не имеет глубоких познаний в математике и программировании. При использовании VisSim'a не требуется владеть программированием на языках высокого уровня или ассемблере. В то же время, специалисты, владеющие программированием, могут создавать собственные блоки, дополняя ими богатую библиотеку стандартных блоков VisSim'a.

Интерфейс программы – это совокупность средств, позволяющих человеку общаться с ней:

- а) вводить и получать данные,
- б) контролировать ход выполнения компьютером программы,
- в) подавать управляющие воздействия и наблюдать реакцию на них программы и т. п.

Программа VisSim предоставляет исследователю графический интерфейс, позволяющий основную часть работы по созданию модели выполнить с помощью мыши, а параметры элементов ввести с клавиатуры. Интерфейс VisSim состоит из главного окна, имеющего меню и ряд кнопок управления и так называемого рабочего пространства, в котором строится

и корректируется модель, наблюдаются результаты ее работы. С точки зрения исследователя интерфейс программы VisSim представляет собой интерактивный виртуальный лабораторный стенд, обеспечивающий построение моделей из отдельных блоков, запуск процесса моделирования, управление им и контроль результатов [23].

Меню VisSim позволяет выполнять, наряду с другими, следующие важные действия:

а) File (работа с файлами) – открывать новую диаграмму, сохранять, а затем при необходимости вновь открывать созданную диаграмму;

б) Edit (Редактирование) – разворачивать на 180 градусов отдельные, выделенные блоки, а также создавать составные блоки и отменять последние изменения, внесенные в диаграмму;

в) Simulate (Моделировать) – запускать процесс моделирования и изменять важные параметры этого процесса: время работы модели и величину шага интегрирования;

г) Blocks (Блоки) – выносить на рабочее пространство элементарные блоки, из которых составляется модель, и надписи;

д) Analyze (Анализировать) – строить частотные характеристики выделенных фрагментов линейной части модели;

е) View (Вид) – устанавливать шрифт надписей и красивый вид блоков;

ж) Help (Справка) – получать справочный материал по программе и отдельным ее блокам.

Диаграммой в VisSim'e называется совокупность связанных, а также автономных блоков и надписей, помещенных на рабочее пространство и способных в известном смысле функционировать при запуске процесса моделирования. Диаграмма может быть сохранена в виде отдельного файла и, при необходимости, открыта вновь.

Модель VisSim'a – это часть диаграммы, содержащая виртуальный аналог реальной или проектируемой системы. Диаграмма может содержать несколько моделей.

Исходными данными для построения модели в VisSim'e являются структурно-функциональная схема моделируемой системы, процесса или объекта и описывающие их дифференциально-алгебраические уравнения.

Вместо таких уравнений могут быть заданы операторы или функции, характеризующие отдельные элементы моделируемой системы, например, передаточные функции для линейных элементов и статические характеристики для нелинейных элементов.

Реальные системы и объекты состоят из отдельных, связанных и взаимодействующих друг с другом элементов. И для всей системы в целом, и для ее отдельных элементов можно указать место приложения воздействия, которое можно назвать входом, и место их реакции на входное воздействие, называемое выходом. И воздействие, и реакция – это некоторые физические величины, являющиеся функциями времени. Модели систем и объектов в программе VisSim строятся из отдельных

элементов – блоков. Блок – это виртуальный аналог физического элемента реальной системы.

Взаимодействие между блоками отображается так называемыми линиями связи, указывающими направление передачи воздействий (сигналов) от одного блока к другому.

Взаимодействие между блоками моделируется сигналами – функциями времени, передаваемыми между блоками по линиям связи. Сигналы в модели могут быть измерены с помощью виртуальных измерительных устройств или рассмотрены и изучены с помощью виртуального осциллографа.

Принцип построения модели в VisSim'е состоит в вынесении на рабочее пространство моделей реальных элементов (блоков) и соединении их в соответствии с заранее составленной структурно-алгоритмической схемой моделируемой системы. Такое построение модели из виртуальных блоков очень похоже, с определенной степенью условности, на построение реальной системы из настоящих блоков в производственных условиях или на лабораторном стенде [22, 23].

#### ***2.4 Анализ электронных и электрических схем (Pspice)***

Программа Pspice (рисунок 2.72) рассчитывает следующие характеристики электронных цепей [24]:

- а) режим цепи по постоянному току в «рабочей точке» (Bias Point);
- б) режим по постоянному току при вариации источников постоянного напряжения или тока, температуры и других параметров цепи (DC Sweep);
- в) чувствительность характеристик цепи к вариации параметров компонентов в режиме по постоянному току (Sensitivity);
- г) малосигнальные передаточные функции в режиме по постоянному току (Transfer Function);
- д) характеристики линеаризованной цепи в частотной области при воздействии одного или нескольких сигналов (AC Sweep);
- е) спектральную плотность внутреннего шума (Noise Analysis);
- ж) переходные процессы при воздействии сигналов различной формы (Transient Analysis);
- з) спектральный анализ (Fourier Analysis);
- и) статистические испытания по методу Монте-Карло и расчет наихудшего случая (Monte Carlo/Worst Case);
- к) многовариантный анализ при вариации температуры (Temperature) и других параметров (Parametric).

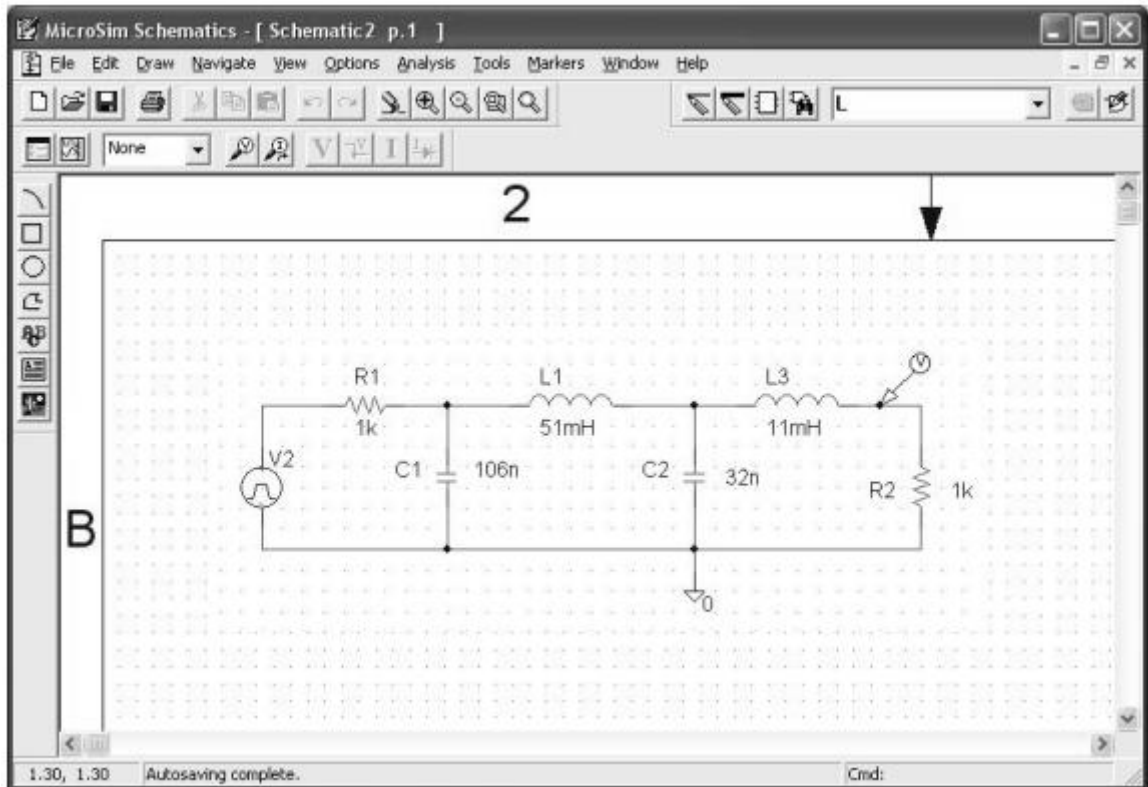


Рисунок 2.72 – Общий вид пакета Pspice

## 2.5 Моделирование систем управления (*Model vision studium*)

Model Vision Studium (MVS) – это интегрированная графическая оболочка для быстрого создания интерактивных визуальных моделей сложных динамических систем и проведения вычислительных экспериментов с ними. Пакет разработан исследовательской группой «Экспериментальные объектные технологии» в Санкт-Петербургском Государственном Техническом Университете (рисунок 2.74).

Пакет предназначен для численного моделирования гибридных систем. Гибридная система – это специальная математическая модель реальных объектов, обладающих одновременно «непрерывными» и «дискретными» свойствами. Такие системы достаточно трудно формализовать, так как необходим единый язык для описания как непрерывных, так и дискретных аспектов поведения, но еще труднее корректно численно решить возникающую при этом математическую задачу.

Проектировщики MVS исходили из того, что современный язык моделирования должен быть в максимальной степени визуальным и поддерживать технологию объектно-ориентированного моделирования.

Корректное численное решение должно получаться автоматически. Таким образом, анализ свойств решаемой математической задачи и выбор

и настройка метода решения должны выполняться именно пакетом, а не пользователем.

Пользователь должен иметь возможность активно вмешиваться в ход вычислительного эксперимента, и, при необходимости, получать о решении как можно больше дополнительной информации.

Работа пакета MVS основана на использовании нового типа объекта – активного динамического объекта и специальной формы наглядного представления гибридного поведения – карты поведения. Разработчики стремились создать простой и наглядный инструмент для моделирования гибридных систем, ориентированный на широкий круг прикладных пользователей.

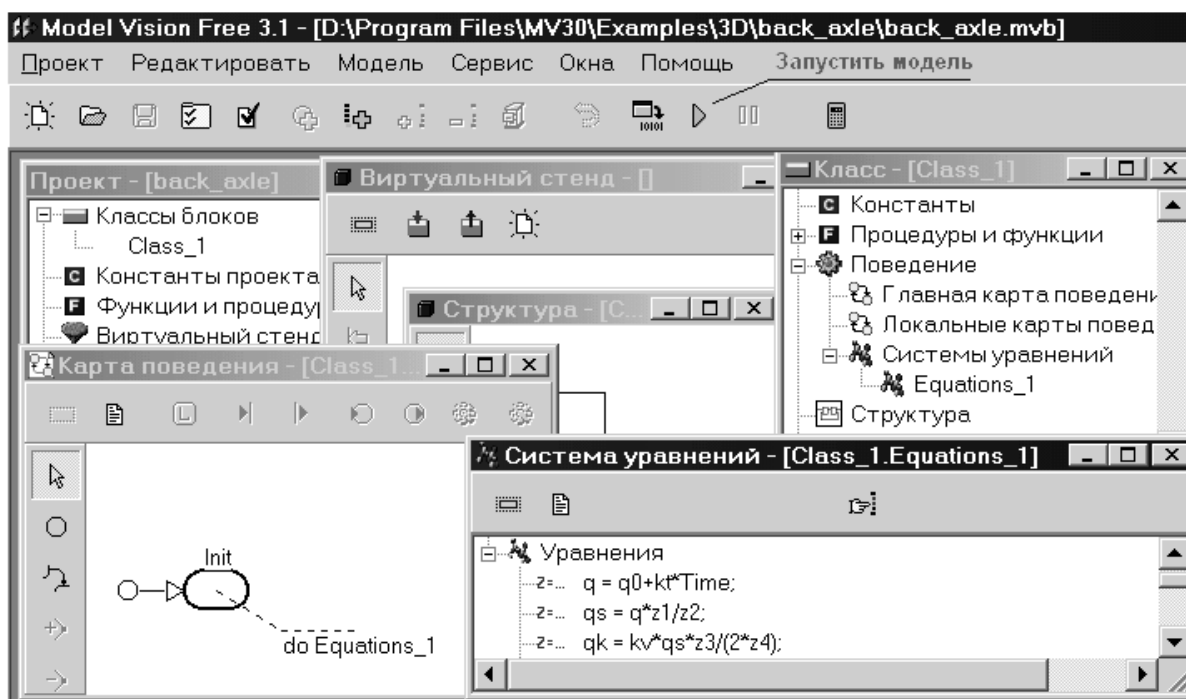


Рисунок 2.74 – Общий вид окна MVS

Как и в большинстве других современных графических инструментов моделирования, в основе технологии MVS лежит понятие виртуального лабораторного стенда. На этом стенде размещаются различные виртуальные блоки моделируемой системы – вновь создаваемые и стандартные, такие как «генераторы сигналов», «измерительные приборы», «устройства отображения», соединенные виртуальными «кабелями». Вся виртуальная аппаратура функционирует независимо и параллельно, подобно ее физическим двойникам «в металле».

Для получения виртуального стенда необходимо описать моделируемую систему на входном языке пакета и создать соответствующий этому описанию программный код, выполнение которого компьютером и будет восприниматься как работа стенда.



Интегрированная оболочка пакета представляет собой многооконную среду, позволяющую редактировать проект, автоматически преобразовывать графическое описание модели в текстовое и текстовое в графическое, подключать библиотеки классов, создавать свои библиотеки классов, создавать выполняемые модели и запускать их, а также запускать специальные подсистемы (оптимизации и символического анализа).

### 2.5.1 Задание на самостоятельную работу №6

**Тема:** моделирование систем управления с помощью пакетов MVR и Simulink.

**Цель работы:** ознакомиться с пакетом моделирования систем MVR, произвести моделирование системы управления и выполнить сравнительный анализ особенностей моделирования в пакетах MVR и Simulink.

#### *Теоретические сведения*

Model Vision Studium (MVS) – компьютерная лаборатория для моделирования и исследования сложных динамических систем. Model Vision Studium - это интегрированная графическая оболочка для быстрого создания интерактивных визуальных моделей сложных динамических систем и проведения вычислительных экспериментов с ними.

Пакет MVS позволяет описать модель на специальном графическом языке, а затем автоматически построить программу для воспроизведения ее поведения, использующую для этого современные численные методы. Пакет предназначен для исследования гибридных систем.

Вводимый класс устройств может наследовать свойства другого класса. Все устройства являются потомками предопределенного класса CDevice, которому приписаны все предопределенные соглашения о взаимодействии с исполняющей системой MVS. При наследовании, вы можете вводить новые параметры, фазовые переменные, константы, алгоритмические процедуры и функции, локальные поведения, локальные устройства и связи, узлы и переходы в карте поведения, анимационные окна и анимационные компоненты, а также переопределять и перегружать алгоритмические процедуры и функции, поведения, узлы и переходы в карте поведения, анимационные окна и анимационные компоненты.

Классы устройств обладают свойством полиморфизма: вместо экземпляра класса-предка может использоваться экземпляр класса-потомка.

К проекту могут быть присоединены ранее созданные библиотеки классов, и при создании модели можно использовать уже готовые классы

устройств. Любой проект может быть превращен в библиотеку классов и любой класс из проекта пользователя может быть добавлен в существующую библиотеку. Пакет поставляется со стандартной библиотекой классов SysLib, содержащей набор наиболее типовых линейных блоков, нелинейных блоков, и источников сигналов.

### ***Индивидуальное задание***

Загрузить программу Model Vision Studium и ознакомиться с программной оболочкой, главным меню, библиотекой компонентов.

Создать согласно варианту (таблица 2.38) динамическую систему и изучить особенности функционирования модели. Изучить элементы, компоненты и блоки, используемые для моделирования объектов. Привести таблицы наследования классов, дифференциальные уравнения, графические зависимости исследуемого объекта. Выполнить сравнительный анализ особенностей моделирования в рассматриваемом пакете и пакете Simulink Matlab.

*Таблица 2.38 – Варианты моделей*

№	Тип модели
1, 14	Электродвигатель постоянного тока
2, 15	Бьющееся сердце
3, 16	Десятичный счетчик
4, 17	Система терморегулирования дома
5, 18	Прыгающий мяч
6, 19	Маятник с подвижным подвесом
7, 20	Простой алгебраический цикл
8, 21	Двойной алгебраический цикл
9, 22	Следящий электропривод с дискретным PD-регулятором
10, 23	следящий электропривод с аналоговым PD-регулятором
11, 24	Бак с жидкостью
12, 25	Система катапультирования
13, 26	Электродвигатель переменного тока

### ***Контрольные вопросы***

1 Понятие функций и процедур, карты связи, иерархия классов, наследование. Принципы объектно-ориентированного программирования в пакете MVS.

2 Иерархия блоков и функциональные связи. Библиотека классов.

3 Математическая основа и особенности реализации численных методов пакета MVS.

## 2.6 Создание виртуальных панелей управления (LabView)

LabView – универсальная среда для разработки систем сбора, обработки данных и управления экспериментом, включающая обширную библиотеку функций, методов анализа (спектральный и корреляционный анализ, вейвлетный анализ, методы фильтрации, статистическая обработка), библиотеки драйверов устройств. Основой среды является графическое программирование на простом и наглядном языке блок-диаграмм, состоящих из функциональных узлов и связей между ними.

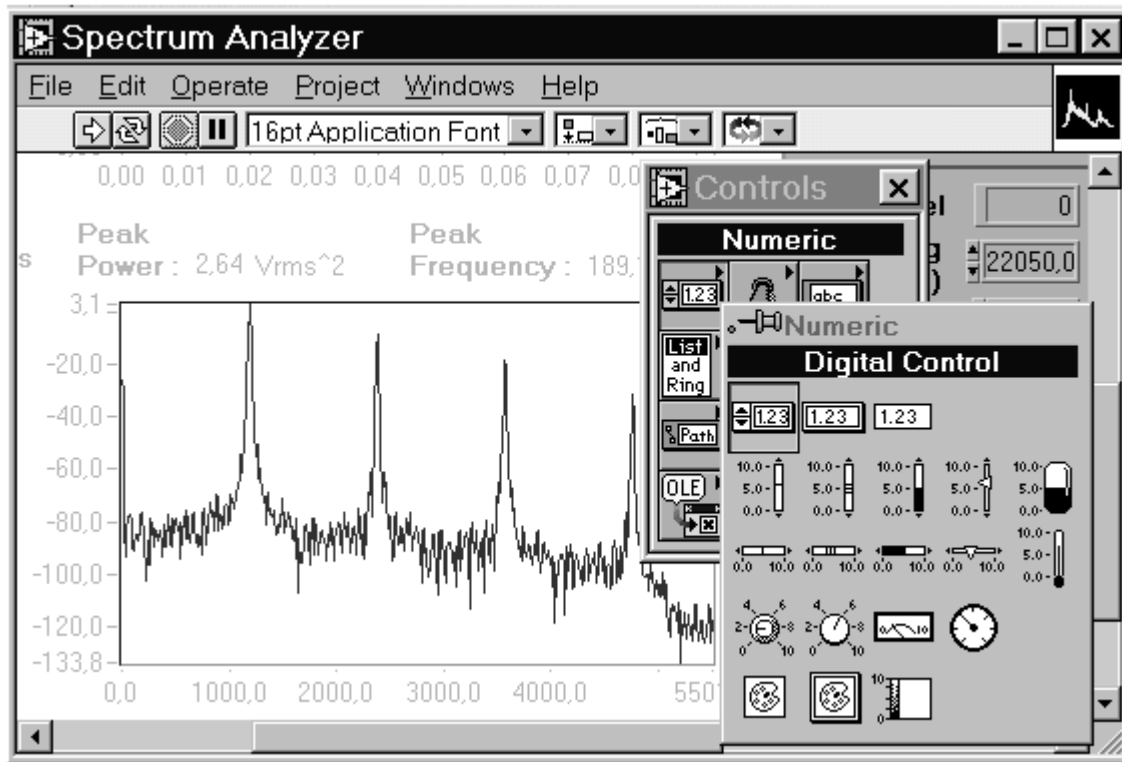
Внешний вид и функциональность LabView повторяет традиционный физический прибор, такой как осциллограф или мультиметр. В LabView проектируется лицевая панель виртуального прибора из набора различных управляющих элементов и индикаторов. После создания лицевой панели определяется функциональность виртуального прибора путем набора блок-схемы [25].

LabView удобно использовать с различными аппаратными средствами, встраиваемыми в персональный компьютер или подключаемыми к нему.

LabView – программная среда, использующая язык графического программирования G (рисунок 2.75). На языке G программа представляется не в виде семантического текстового описания, а в виде графического представления основных операторов программирования на блок-диаграмме и проводников потока данных, определяющих последовательность выполнения программы.

В LabView интерфейс пользователя создается с использованием палитры Элементов (Controls). Для соблюдения формальной аналогии с реальным прибором интерфейс пользователя называется лицевой панелью. Обычно лицевая панель имеет серый фон рабочего пространства. Собственно программирование осуществляется на панели диаграмм с использованием графических представлений узлов и функций. Исходный код на языке G называется блок-диаграммой. Программист использует для создания блок-диаграммы палитру Функций (Function), из которой извлекает, по мере необходимости, графические представления функций [25].

Программная среда LabView полностью поддерживает аппаратные средства, имеющие аппаратные интерфейсы типа GPIB, VXI, PXI, RS-232, RS-485, а также DAQ устройства. LabView также поддерживает сетевые протоколы TCP/IP и протоколы обмена данными между программами ActiveX.



*Рисунок 2.75 – Общий вид окна LabView*

С помощью программной среды LabView можно разрабатывать программно-аппаратные комплексы для тестирования, измерения, ввода данных, анализа и управления внешним оборудованием. LabView – это 32-х разрядный компилятор, который создает как автономные модули (.EXE), так и совместно используемые динамические библиотеки (.DLL).

### ***2.7 Интегрированная среда разработки ATMEL AVR Studio***

AVR Studio – это интегрированная отладочная среда разработки приложений для микроконтроллеров AVR компании Atmel [26].

AVR Studio (рисунок 2.76) содержит:

- а) средства создания и управления проектом;
- б) редактор кода на языке ассемблер;
- в) транслятор языка ассемблера (Atmel AVR macroassembler);
- г) отладчик (Debugger);
- д) программное обеспечение верхнего уровня для поддержки внутрисхемного программирования (In-System Programming, ISP) с использованием стандартных отладочных средств Atmel AVR.

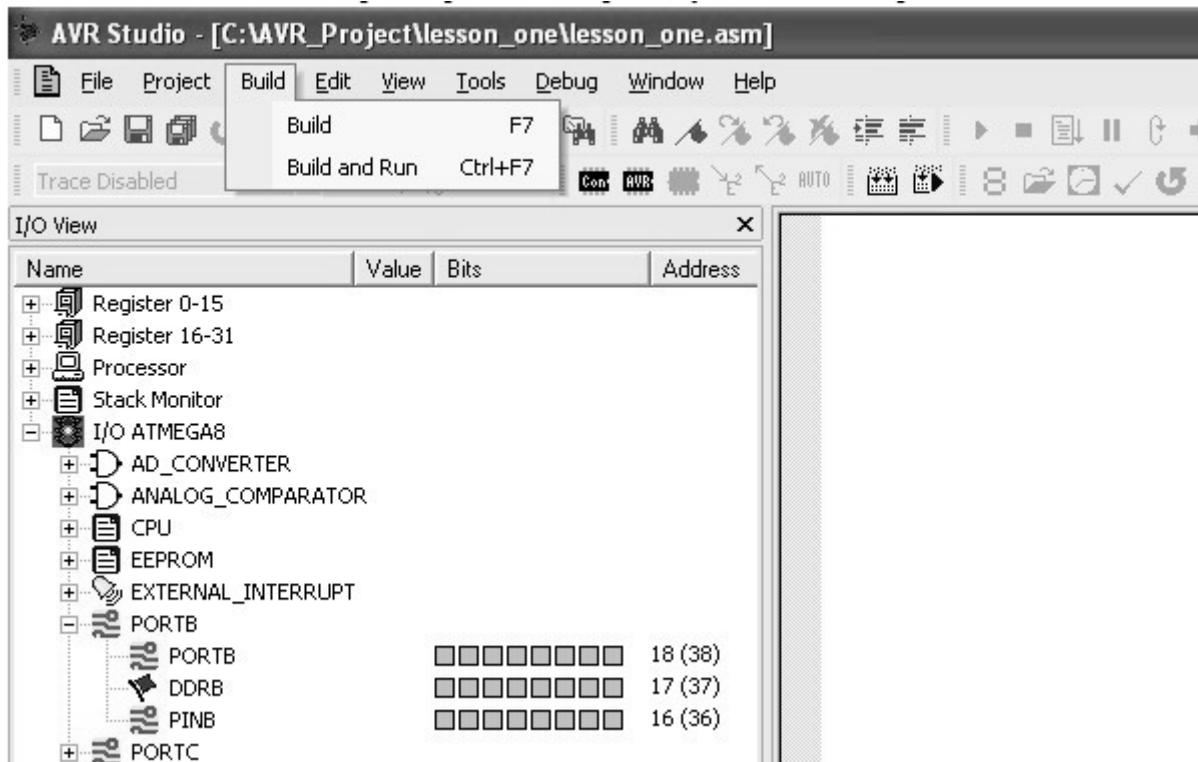


Рисунок 2.76 – Общий вид окна AVR Studio

Работа с AVR Studio начинается с создания проекта. При создании проекта необходимо указать используемый микроконтроллер и платформу, на которой будет производиться отладка программы.

Написание программы производится в окне редактора текста программы. Для использования символических имен регистров специального назначения вместо их адресов необходимо подключить (директива `.include`) к проекту файл определения регистров специального назначения (например, `m16def.inc` для ATmega16). Включаемые файлы входят в прикладное программное обеспечение AVR Studio и при инсталляции помещаются в папку `Appnotes` в директории установки AVR Studio. Примеры программ доступны в большом количестве в качестве приложений к руководствам по применению микроконтроллеров AVR.

Написание программы в AVR Studio производится на языке ассемблер. Система команд микроконтроллера описана в упомянутых выше руководствах по применению в документе AVR Instruction Set либо в файле справки, встроенном в AVR Studio (меню `Help \ AVR Tools User Guide \ AVR Assembler`), в котором содержатся достаточно подробные комментарии к каждой команде.

Последние версии AVR Studio содержат тестовую версию AVR ассемблера второй версии, который в дополнение к стандартному ассемблеру поддерживает новые директивы ассемблера, Си - подобные директивы препроцессора, создание переменных определенного типа. Более подробную информацию можно найти в файле справки.

## 2.8 Общие понятия и структура SCADA-систем

В настоящее время SCADA (Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных) является наиболее перспективной технологией автоматизированного управления во многих отраслях промышленности [27].

В последние несколько десятилетий за рубежом резко возрос интерес к проблемам построения высокоэффективных и высоконадежных систем диспетчерского управления и сбора данных.

С одной стороны, это связано со значительным прогрессом в области вычислительной техники, программного обеспечения и телекоммуникаций, что увеличивает возможности и расширяет сферу применения автоматизированных систем.

С другой стороны, развитие информационных технологий, повышение степени автоматизации и перераспределение функций между человеком и аппаратурой обострило проблему взаимодействия человека-оператора с системой управления. Расследование и анализ большинства аварий и происшествий в промышленности и на транспорте, часть из которых привела к катастрофическим последствиям, показали, что, если в 60-х годах XX века ошибка человека являлась первоначальной причиной лишь 20% инцидентов, то в 90-х годах доля «человеческого фактора» возросла до 80%, причем, в связи с постоянным совершенствованием технологий и повышением надежности электронного оборудования и машин, доля эта может еще возрасти (рисунок 2.77).

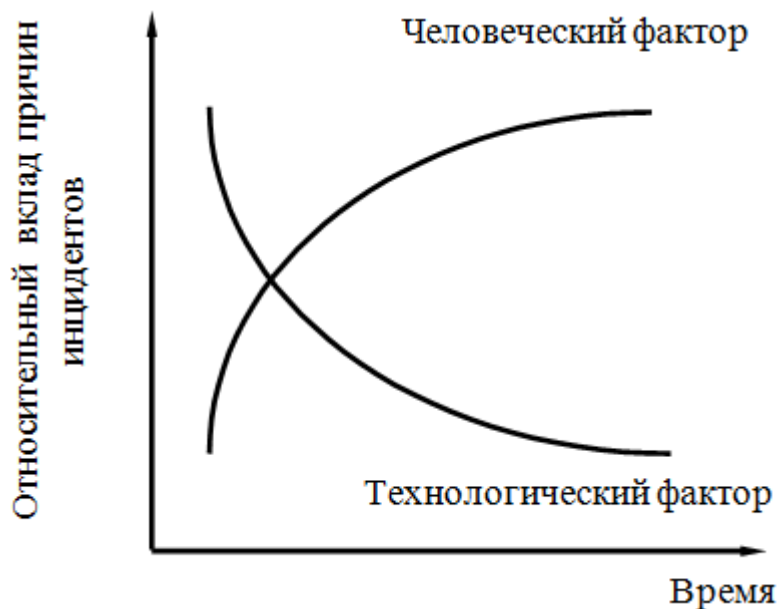


Рисунок 2.77 – Тенденции причин аварий в сложных автоматизированных системах

Основной причиной таких тенденций является старый традиционный подход к построению АСУ, который применяется часто и в настоящее

время: ориентация в первую очередь на применение новейших технических (технологических) достижений, стремление повысить степень автоматизации и функциональные возможности системы и, в то же время, недооценка необходимости построения эффективного человеко-машинного интерфейса (HMI - Human-Machine Interface), т.е. интерфейса, ориентированного на оператора.

Возникла необходимость применения нового подхода при разработке таких систем, а именно, ориентация в первую очередь на человека-оператора (диспетчера) и его задачи. Реализацией такого подхода и являются SCADA-системы, которые иногда даже называют SCADA/HMI.

Управление технологическими процессами на основе SCADA-систем стало осуществляться в передовых западных странах в 80-е годы XX века. В странах СНГ переход к управлению на основе SCADA-систем стал осуществляться несколько позднее, в 90-е годы.

SCADA-системы наилучшим образом применимы для автоматизации управления непрерывными и распределенными процессами, какими являются нефтегазовые технологические процессы. Кроме нефтяной и газовой промышленности, SCADA-системы применяются в следующих областях [28]:

- а) управление производством, передачей и распределением электроэнергии;
- б) промышленное производство;
- в) водозабор, водоочистка и водораспределение;
- г) управление космическими объектами;
- д) управление на транспорте (все виды транспорта: авиа, метро, железнодорожный, автомобильный, водный);
- е) телекоммуникации;
- ж) военная область.

В мире насчитывается не один десяток компаний, активно занимающихся разработкой и внедрением SCADA-систем. Программные продукты многих из этих компаний представлены на российском рынке.

SCADA – это процесс сбора информации реального времени с удаленных объектов для обработки, анализа и возможного управления этими объектами.

В SCADA-системах в большей или меньшей степени реализованы основные принципы, такие, как работа в режиме реального времени, использование значительного объема избыточной информации (высокая частота обновления данных), сетевая архитектура, принципы открытых систем и модульного исполнения, наличие запасного оборудования, работающего в «горячем резерве» и др [27].

Все современные SCADA-системы включают три основных структурных компонента (рисунки 2.78).

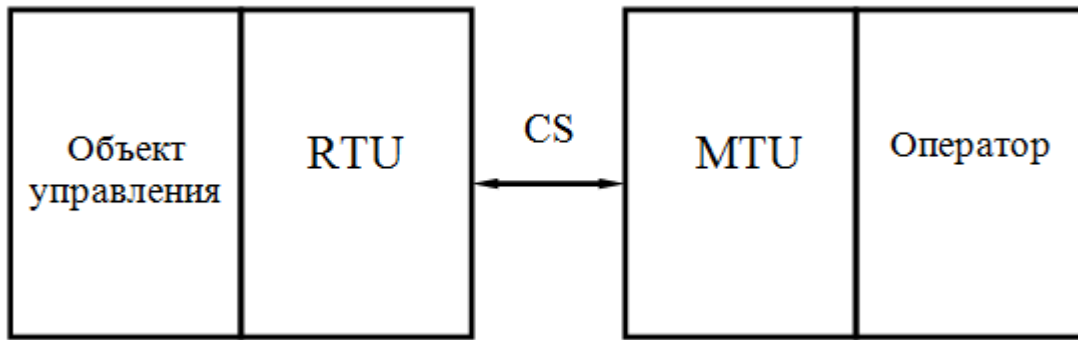


Рисунок 2.78 – Основные структурные компоненты SCADA-системы

Remote Terminal Unit (RTU) - удаленный терминал, осуществляющий обработку задачи (управление) в режиме реального времени.

Системы реального времени бывает двух типов: системы жесткого реального времени и системы мягкого реального времени.

*Системы жесткого реального времени* не допускают никаких задержек

Спектр воплощения RTU широк - от примитивных датчиков, осуществляющих съем информации с объекта, до специализированных многопроцессорных отказоустойчивых вычислительных комплексов, осуществляющих обработку информации и управление в режиме жесткого реального времени. Конкретная его реализация определяется конкретным применением. Использование устройств низкоуровневой обработки информации позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом.

Master Terminal Unit (MTU) - диспетчерский пункт управления (главный терминал); осуществляет обработку данных и управление высокого уровня, как правило, в режиме мягкого реального времени. Одна из основных функций - обеспечение интерфейса между человеком-оператором и системой. MTU может быть реализован в самом разнообразном виде - от одиночного компьютера с дополнительными устройствами подключения к каналам связи до больших вычислительных систем и/или объединенных в локальную сеть рабочих станций и серверов.

Communication System (CS) - коммуникационная система (каналы связи), необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на RTU.

В названии SCADA присутствуют две основные функции, возлагаемые на системы этого класса:

- а) сбор данных о контролируемом процессе;
- б) управление технологическим процессом, реализуемое ответственными лицами на основе собранных данных и правил (критериев), выполнение которых обеспечивает наибольшую эффективность технологического процесса.



SCADA-системы обеспечивают выполнение следующих функций [28]:

- а) прием информации о контролируемых технологических параметрах от контроллеров нижних уровней и датчиков;
- б) сохранение принятой информации в архивах;
- в) обработка принятой информации;
- г) графическое представление хода технологического процесса, а также принятой и архивной информации в удобной для восприятия форме;
- д) прием команд оператора и передача их в адрес контроллеров нижних уровней и исполнительных механизмов;
- е) регистрация событий, связанных с контролируемым технологическим процессом и действиями персонала, ответственного за эксплуатацию и обслуживание системы;
- ж) оповещение эксплуатационного и обслуживающего персонала об обнаруженных аварийных событиях, связанных с контролируемым технологическим процессом и функционированием программно-аппаратных средств АСУТП с регистрацией действий персонала в аварийных ситуациях;
- з) формирование сводок и других отчетных документов на основе архивной информации;
- и) обмен информацией с автоматизированной системой управления предприятием;
- к) непосредственное автоматическое управление технологическим процессом в соответствии с заданными алгоритмами.

Данный перечень функций, выполняемых SCADA-системами, не является абсолютно полным, более того, наличие некоторых функций и объем их реализации сильно варьируется от системы к системе.

Существует два типа управления удаленными объектами в SCADA-системах: автоматическое и инициируемое оператором системы.

Процесс управления в современных SCADA-системах имеет следующие особенности:

- а) процесс SCADA применяется в системах, в которых обязательно наличие человека (оператора, диспетчера);
- б) процесс SCADA был разработан для систем, в которых любое неправильное воздействие может привести к отказу объекта управления или даже катастрофическим последствиям;
- в) оператор несет, как правило, общую ответственность за управление системой, которая при нормальных условиях только изредка требует подстройки параметров для достижения оптимальной производительности;
- г) активное участие оператора в процессе управления происходит нечасто и в непредсказуемые моменты времени, обычно в случае наступления критических событий (отказы, нештатные ситуации и пр.);
- д) действия оператора в критических ситуациях могут быть жестко ограничены по времени (несколькими минутами или даже секундами).

В настоящее время на российском рынке представлено несколько десятков зарубежных и отечественных SCADA-продуктов. Некоторые зарубежные SCADA-системы, известные в мире, на российском рынке пока не представлены (например, Cube, Panorama, Cimview и др.). Но эта ситуация может в любой момент измениться, как это произошло, например, с австралийской SCADA-системой Citect: система завоевала рынок Юго-Восточной Азии и Америки, в Европе рост ее продаж составил 30% в год, а в России она была неизвестна, пока российская фирма-дистрибьютор RTSoft не начала распространение системы на российском рынке.

В странах СНГ сегодня наиболее популярны следующие зарубежные SCADA-пакеты:

- а) In Touch (Wonderware, США);
- б) iFIX (Intellution, США);
- в) SIMATIC WinCC (Siemens, Германия);
- г) Citect (Ci technologies, Австралия);
- д) RTAP/plus (HP, Канада);
- е) Wizcon (PC Soft International, Израиль-США);
- ж) Sitex и Phocus (Jade SoftWare, Великобритания);
- з) Real Flex (BJ Software Systems, США);
- и) Factory Link (US Data Corp., США);
- к) View Star 750 (AEG, Германия);
- л) PlantScape (SCAN 3000) (Honeywell, США).

### ***Контрольные вопросы***

- 1 Определение и общая структура SCADA.
- 2 Функциональная структура SCADA.
- 3 Особенности SCADA как процесса управления.
- 4 Основные требования к SCADA-системам.
- 5 Основные возможности современных SCADA-пакетов.
- 6 Тенденции развития аппаратных и программных средств SCADA-систем.

### ***2.9 Сетевые технологии в электроприводе***

Говоря о роли электропривода в автоматизированных распределенных системах, следует иметь в виду прежде всего системы АСУТП, поскольку именно в них электропривод является основным средством осуществления необходимых операций по приведению в движение разнообразных рабочих органов технологического оборудования и обрабатываемых материалов [29].

Автоматизированная система управления технологическими процессами (АСУТП) – совокупность аппаратно-программных средств,

осуществляющих контроль и управление производственными и технологическими процессами, поддерживающих обратную связь и активно воздействующих на ход процесса при отклонении его от заданных параметров, обеспечивающих регулирование и оптимизацию управляемого процесса [30].

Электропривод успешно применяется не только в качестве основного источника энергии, с помощью которой происходит механическая обработка, но выполняет также множество вспомогательных функций по требуемому перемещению рабочих органов, осуществляющих другие виды обработки (например, лазерная обработка, покраска, производство печатной продукции и т. п.). Не очень заметны, но, безусловно, необходимы электроприводы многочисленных насосных и вентиляционных установок, которые являются первичными источниками энергии для пневмо- и гидроагрегатов, поддерживают требуемые условия окружающей среды в промышленных, административных и жилых помещениях. Дистанционное управление различными заслонками, задвижками, вентилями также осуществляется исполнительными механизмами на базе электроприводов.

Постоянно расширяется сфера применения регулируемых электроприводов, с помощью которых удается повысить качество выпускаемой продукции, экономить материальные и энергетические ресурсы. Переход от нерегулируемого к регулируемому электроприводу, как правило, увеличивает объем информационного потока между электроприводами, оператором и внешними управляющими устройствами. Наличие в силовом канале регулируемого электропривода электронного преобразователя, управляемого достаточно мощным микроконтроллером, создает условия для более качественного управления, контроля и диагностики как собственно ЭП, так и технологического оборудования.

Развитие систем управления электроприводом происходит как в части совершенствования алгоритмов регулирования координат электромеханического преобразователя (токи, момент, скорость), так и за счет расширения вспомогательных функций по защите, диагностике, автоматической настройке регуляторов. Обеспечивается несколько уровней человеко-машинного интерфейса, позволяющих не только контролировать основные параметры работы электропривода, но и осуществлять протоколирование аварийных ситуаций, визуализацию процессов, дистанционный контроль и управление.

Наблюдается тенденция встраивания в системы управления (СУ) электроприводами (ЭП) все более развитых микропроцессорных средств, способных выполнять функции управления, возлагаемые обычно на ПЛК и системы ЧПУ – формирование требуемых законов изменения задающих воздействий на входы регуляторов координат электропривода.

Появляется возможность реализации в той или иной мере функций технологических контроллеров – регуляторов технологических величин. И, наконец, СУ ЭП оснащаются встроенными средствами для обмена

данными с внешними устройствами по промышленным коммуникационным сетям, что позволяет интегрировать современный ЭП в распределенные системы АСУТП [31]. Можно говорить, что ЭП становится интеллектуальным устройством в составе АСУТП, совмещающим множество различных функций.

В классификации автоматизированных систем производства ЭП является элементом нижних уровней и, следовательно, может взаимодействовать как непосредственно с датчиками различных технологических величин, так и с другими ЭП, исполнительными механизмами и системами, чем обеспечивается согласованная работа всех составных частей технологической установки.

Для нижнего уровня производственных систем характерны следующие общие типовые задачи:

- а) хранение и выполнение общего алгоритма управления;
- б) локальное аналоговое управление;
- в) локальное цифровое управление;
- г) связь с оператором;
- д) связь с компьютером верхнего уровня.

Наиболее сложными и многообразными являются задачи аналогового управления непрерывной частью объекта, которые удобно разделить на четыре уровня [1].

Первый уровень – задачи сбора, первичной обработки и преобразования информации, т. е. предварительная аналоговая или цифровая фильтрация сигналов, нормализация и масштабирование, аналого-цифровое преобразование, функциональные преобразования, вычисление спектральных характеристик сигналов и т. п. Важность задач этого уровня определяется еще и тем обстоятельством, что именно от эффективного решения задач на этом уровне зависит сохранение точности дальнейших преобразований и обработки, погрешности измерения могут существенно исказить результаты вычислений. Одной из устойчивых тенденций является приближение обработки информации к точкам ее получения.

Алгоритмы решения задач на этом уровне (особенно связанные с цифровой фильтрацией и спектральной обработкой) предъявляют очень высокие требования к вычислительным средствам, в части требуемых объемов памяти и быстродействия.

Второй уровень – задачи геометрических преобразований систем координат, связанных с различными функциональными пространствами; определения текущих параметров измеренных процессов, представленных в цифровой форме (например, скоростей и ускорений его протекания); совместного решения линейных и нелинейных алгебраических уравнений, в том числе и рекуррентного типа; решения систем дифференциальных уравнений (в том числе в частных производных); прогнозирования хода управляемых процессов; интерполирования и экстраполирования вычисляемых функций; поиска данных в таблицах; цифроаналоговых

преобразований и т. п. Часто совокупность задач, решаемых на этом уровне, характеризуют одним термином – вторичная обработка информации.

Третий уровень – задачи оптимизации, т. е. задачи поиска экстремума функции (функционала) одной или нескольких переменных, на которые наложены определенные ограничения. Методы решения таких задач составляют предмет раздела прикладной математики, называемого математическим программированием. В зависимости от характера целевой функции (критерия оптимизации) и ограничений, наложенных на управляемые переменные, эти задачи решаются методами линейного, нелинейного, стохастического, целочисленного и динамического программирования. Решение этих задач, как правило, связано с предъявлением высоких требований к вычислительным ресурсам, особенно для задач большой размерности.

Четвертый уровень – задачи представления информации в виде, удобном для восприятия человеком – оператором. На основе этой информации он может либо принимать решение по введению в систему управляющих воздействий, либо наблюдать за правильностью функционирования технических средств, составляющих систему или системы. Характер используемых методов решения таких задач зависит от конкретного состава и принципов организации устройств отображения информации и конструкции органов управления. Не все из указанных задач целесообразно решать встроенными микропроцессорными средствами ЭП. Многие производители комплектных ЭП выпускают специальные опции, предназначенные для автономной работы небольших установок и совмещающие в себе функции контроллера управления движением и собственно системы управления ЭП.

## 3 СТРУКТУРНЫЙ АНАЛИЗ И МОДЕЛЬ ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ СИСТЕМ

### 3.1 Понятие и особенности системного анализа

Создание более сложных информационных и технических систем требует развития и совершенствования методов их изучения.

Системный анализ – инструмент научных обоснований принимаемых решений, повышающий их качество путем создания сложных многовариантных решений [8, 9].

Основной принцип системного анализа - концепция целостности, предполагающая наличие системного эффекта, т.е. у системы появляются свойства, не присущие ее частям. Причиной системного эффекта является наличие связей между элементами.

Применение методов системного анализа направлено на формализацию информации об объектах для хранения и обработки на ЭВМ. Движущей силой развития технических систем есть противоречия. Они содержатся в любой системе и разнообразны по форме и содержанию, имеют исторически приходящий характер. Поэтому систему необходимо рассматривать в развитии, что дает возможность ставить задачи прогноза. Противоречия возникают между потребностями общества и данной тех. системой — внешние; между частями и элементами системы — внутренние.

Противоречия разделяют [6]:

- а) административные – возникают при формулировке задачи;
- б) технические – возникают между частями системы при изменении одной из них;
- в) физические – формулируются путем предъявления к частям объекта противоположных требований.

Существует несколько видов описаний по уровням и аспектам осмысления подхода к описаниям.

Параметрическое описание – это простейшая форма описания, включающая описание свойств, признаков, характеристик, отношений объектов, основанное на эмпирических наблюдениях.

Морфологическое описание – это определение поэлементного состава объекта или системы, его строения, определение взаимосвязей параметров, выявленных в процессе предыдущего описания.

Элемент – минимальная неделимая часть системы, получаемая ее разделением в условиях данной задачи. Функции и роль элемента определяются его местом и порядком включения в систему.

Функциональное описание – переход к определению функциональных зависимостей между параметрами (функционально-параметрическое описание), между элементами (функционально-морфологическое описание), между параметрами и строением объекта.

Функция – взаимосвязь, определяющее назначение и порядок включения части в целое.

Описание жизни объекта – это наиболее полный вид описания, выявление общей целостности жизни объекта, условий, определяющих состояния системы, смену режимов работы системы.

Основные понятия системного анализа: декомпозиция, иерархичность.

Декомпозиция – разделение объекта на составляющие элементы в рамках задачи, создание иерархии элементов, выделение функциональных элементов.

Принцип декомпозиции – разбиение представлений каждого уровня на ряд составных частей – блоков - для отдельного проектирования.

Принцип иерархичности – структурирование представлений по степени детализации описания. В дереве "И" на верхнем уровне расположен объект; на первом уровне – N – взаимосвязанных элементов. Каждый элемент в свою очередь рассматривается как система, состоящая из M-элементов. Дополнительно к дереву можно построить графы для каждого уровня.

### ***3.2 Блочнo – иерархический подход к проектированию систем***

Создание новых систем относится к задачам структурного синтеза. При рассмотрении таких задач в системах автоматизированного проектирования необходимо решать вопросы представления информации о структуре системы, а также вопросы, связанные с поиском средств улучшения технических решений для элементов системы. Наиболее наглядным и эффективным способом являются графы и деревья, например, И-ИЛИ деревья. Обычно И деревья применяются для описания структуры конкретных систем.

И-деревья представляют собой совокупность вершин, сгруппированных на различных иерархических уровнях, то есть представляется множеством вершин и связывающих их ребер. Каждый иерархический уровень представляет собой проектируемую систему с различной степенью детализации. Нулевой иерархический уровень (система) является наиболее абстрактным, а последний – наиболее детализированным.

Отображение структуры целого класса подобных систем осуществляется с использованием И-ИЛИ деревьев, которые выражают накопленный опыт в разработке систем определенного типа. Построение общего И-ИЛИ дерева технических решений обычно производят путем объединения И деревьев нескольких конкретных систем.

И-ИЛИ дерево представляет собой однонаправленный граф с одной корневой вершиной. Каждая вершина, кроме корневой, подчинена какой-либо одной и только одной вершине, расположенной на более высоком уровне. Вершины бывают двух типов: И, ИЛИ. Выделяют также висячие вершины («листья» дерева). Вершины типа

И, ИЛИ имеют две или более подчиненных вершин. Висячие вершины подчиненных не имеют.

Уровни (ярусы), на которых расположены вершины И и ИЛИ, должны чередоваться. Каждая вершина является техническим решением для одного из элементов объекта. Подчиненные вершины являются составными частями вышестоящей, если это вершина типа И, или ее возможными вариантами, если это вершина типа ИЛИ.

Разбиение объекта на элементы производится в зависимости от задач, стоящих перед разработчиком. Выделение элементов может быть выполнено [32]:

- а) по функциональному назначению;
- б) по конструктивным признакам;
- в) по технологическим признакам;
- г) по последовательности сборочных операций;
- д) по другим критериям.

Наиболее часто при создании новых систем и объектов используется выделение функционального назначения элементов, а затем выбор их конструктивного исполнения.

При анализе существующих технических объектов (ТО) для построения иерархии элементов используют их конструктивное исполнение в соответствии с их функциональным назначением.

Узлом ТО называется множество конструктивно связанных деталей, в совокупности выполняющих хотя бы одну функцию по обеспечению работы других функциональных элементов ТО. Крупные, сложные узлы можно разделить на более простые подузлы для выполнения подчиненных функций и так далее с образованием иерархии элементов и функций.

Деталью ТО называется отдельное тело (элемент) из однородного материала, выполненное без использования сборочных операций, имеющее множество определенных форм и выполняющее хотя бы одну функцию в системе.

В зависимости от решаемой задачи деталь также может быть представлена в виде совокупности элементов ("разрезана" на "неделимые элементы").

Неделимым элементом называется часть детали (или сама деталь), имеющая минимальное число функций по обеспечению работы других элементов, при любом делении которой появляются элементы, не имеющие самостоятельной функции или с одинаковыми функциями (шарик в подшипнике). Следует отметить, что степень деления и выделения элементов объекта связаны с решаемой задачей. Вид задачи определяет уровень абстракции, на котором рассматривается объект и степень детализации его элементов при разбиении и изучении.

Наиболее часто используемые в ТО функциональные объекты, инвариантные большинству отраслей техники: несущие элементы, элементы связи, элементы передачи, двигатели, элементы управления,



элементы гашения скоростей и ускорений, элементы формирования объемов и потоков, движители.

Несущие элементы. Функции: обеспечение определенной формы ТО и (или) взаиморасположения и (или) движения в пространстве (балки, каркасы, корпуса, кронштейны и т.д.).

Элементы связи. Функция: обеспечение определенной степени свободы движения одних элементов ТО относительно других (от 0 до 6). К ним относятся соединения разъемные и неразъемные, элементы, реализующие кинематические пары, муфты, стержни, нити.

Элементы передачи. Функции: передача на расстояние механической энергии, движения или статических сил и моментов с одновременным преобразованием скоростей (угловых скоростей), моментов, сил и их направлений. К ним относятся гибкие валы, карданные валы, зубчатые и другие передачи, редукторы и коробки передач, гидравлические и другие передачи. Элементы передач образуют силовые цепи различной конфигурации.

Двигатели. Функция: получение необходимой мощности в результате преобразования заданного вида энергии в механическую. К ним относятся различные двигатели - паровые, гидравлические, электрические, турбины, двигатели внутреннего сгорания и т.п.

Элементы управления. Функции: сбор, хранение, обработка информации для выработки информации об управляющем воздействии и передача ее исполнительным органам, например двигателям, которые через элементы передач воздействуют на рабочие органы ТО. Используются в объектах с переменными во времени процессами. К ним относятся органы чувств и мозг человека, датчики регистрации и измерения различных сигналов, усилители и т.д. Комплекс таких элементов - система управления.

Элементы гашения скоростей и ускорений. Функции: уменьшение скоростей движения и (или) амплитуды колебаний твердых и других тел и веществ. К ним относятся амортизаторы, тормоза.

Элементы формирования объемов и потоков. Функции: хранение и (или) транспортировка необходимых по величине и форме веществ и их смесей. К ним относятся баки, трубопроводы, емкости, цистерны и т.д.

Движители. Функции: преобразование работы двигателя или другого источника энергии в работу на преодоление сопротивления движению ТО. К ним относятся колесные, гусеничные и др. Движители транспортных средств, насосы, шнеки, ленты транспортеров и др.

В ряде случаев элементы ТО могут одновременно выполнять несколько функций.

### **3.3 Аспекты объектно - ориентированного анализа и проектирования систем**

В настоящее время наиболее часто используется структурная (функциональная) и объектно-ориентированная методологии разработки программного обеспечения.

Объектно-ориентированный подход (ООП) является эволюционным развитием структурного подхода и обеспечивает более естественную структуризацию системы, снижение сложности ее представления и облегчает нахождение технических решений при проектировании.

Для анализа и документирования предметной области разработан и стандартизован специализированный графический язык UML, который используется для создания метамodelей программных систем. Разработана и унифицированная последовательность этапов проектирования, обеспечивающая стадийность и итеративность процесса их выполнения [34, 35].

В области проектирования сложных программных систем ООП предоставляет системные диаграммные методики, обеспечивающие разносторонние аспекты рассмотрения ПС, которые в совокупности формируют их архитектуру.

Представление информации требует использования наглядных диаграммных методик. Существует множество различных средств визуализации информации, которые применяются на практике. Для структурного анализа используют диаграммы потоков данных (DFD), диаграммы сущность-связь (ERD) и диаграммы переходов состояний (STD) [8,9], которые дают полное описание системы независимо от её новизны.

Целью анализа является построение логической функциональной спецификации – подробное описание того, что должна делать система, без рассмотрения путей реализации (чёткое представление о конечных результатах). Выбор этих средств зависит от решаемой задачи, поэтому рассмотрим ряд наиболее часто используемых для указанных задач диаграмм и методик соответственно:

а) DFD (Data Flow Diagrams) – диаграммы потоков данных; совместно со словарями данных и спецификациями процессов (миниспецификация) иллюстрируют функции, которые система должна выполнять;

б) ERD (Entity Relationship Diagrams) – диаграммы "сущность- связь" показывают отношения между данными;

в) STD (State Transition Diagrams) – диаграммы перехода состояний показывают зависящее от времени поведение системы (аспекты реального времени).

Основой документирования метамодели системы при функциональной декомпозиции является диаграмма потоков данных (DFD), элементами которой являются внешние сущности (пользователи,

другие программные системы), процессы и потоки данных между ними. Переходной формой представления метамодели является технология SADT (IDEF0), которая различает входы-выходы, управление, исполнителей, что аналогично представлению системы в виде черного ящика. SADT упрощает переход к объектной модели, т.к. за каждой активностью (функцией) можно закрепить реализующий ее объект [36].

При проектировании на основе ООП прежде всего выполняется анализ предметной области, выделение основных понятий и целей проектирования, что не противоречит и структурно-функциональному подходу. При ООП с использованием универсального языка моделирования UML строят диаграммы прецедентов использования, на которых выделяют границу системы, пользователей и доступные им функции. Последовательность использования функций не рассматривается, так как ООП предполагает большое количество вариантов их использования. Отображение временной последовательности работы ПС осуществляется на специализированных диаграммах поведения системы [34, 35].

Кроме DFD-диаграмм для представления функций системы могут применяться SADT-диаграммы (IDEF0-модель). SADT является технологией для анализа систем как множества взаимосвязанных активностей или функций. Эту технологию целесообразно использовать на ранних этапах жизненного цикла системы для понимания сущности функций и взаимосвязей в системе.

SADT-диаграммы полнее, чем DFD, описывают функциональный аспект системы, так как они определяют исполнителей процесса, а также правила, в соответствии с которыми выполняются процессы.

SADT-диаграммы позволяют получить ясную картину того, какие процессы предметной области должны быть отражены при проектировании программной системы, и какая требуется структура программной системы для их моделирования.

SADT-технология обеспечивает единое понимание процессов в системе и является средством для представления идей и проектных решений. SADT-диаграммы устраняют неопределенности трактовки процессов и являются способом обмена информацией между программистами и поддержки взаимосвязи с пользователями систем.

Чисто функциональная ориентация этих диаграмм важна, так как функции системы рассматриваются независимо от объектов, которые их выполняют, что, в свою очередь, позволяет четко отделить сущность (значение) системы от ее реализации [36]. Представленная на рисунке 3.1 контекстная SADT-диаграмма определяет входные данные для разрабатываемого программного комплекса, выходные данные, требования к программному комплексу (ограничения, правила, протоколы), а также аппаратные и программные ресурсы, необходимые для работы программного комплекса.

Для основной активности (функционального блока), представленного на контекстной SADT-диаграмме (рисунок 3.2), может быть проведена декомпозиция, т.е. активность «Автоматизация конструкторской подготовки производства» может быть представлена в виде графа (детализирующей SADT-диаграммы), составленного из активностей более низкого уровня с внутренними связями. Полученные активности, в свою очередь, также могут быть представлены в виде более простых (элементарных) активностей и связей. Детализирующая SADT-диаграмма состоит из нескольких функциональных блоков, расположенных в порядке доминирования.

Количество уровней декомпозиции диаграмм определяется сложностью задачи. На одном уровне желательно представлять не более 4-6 активностей.

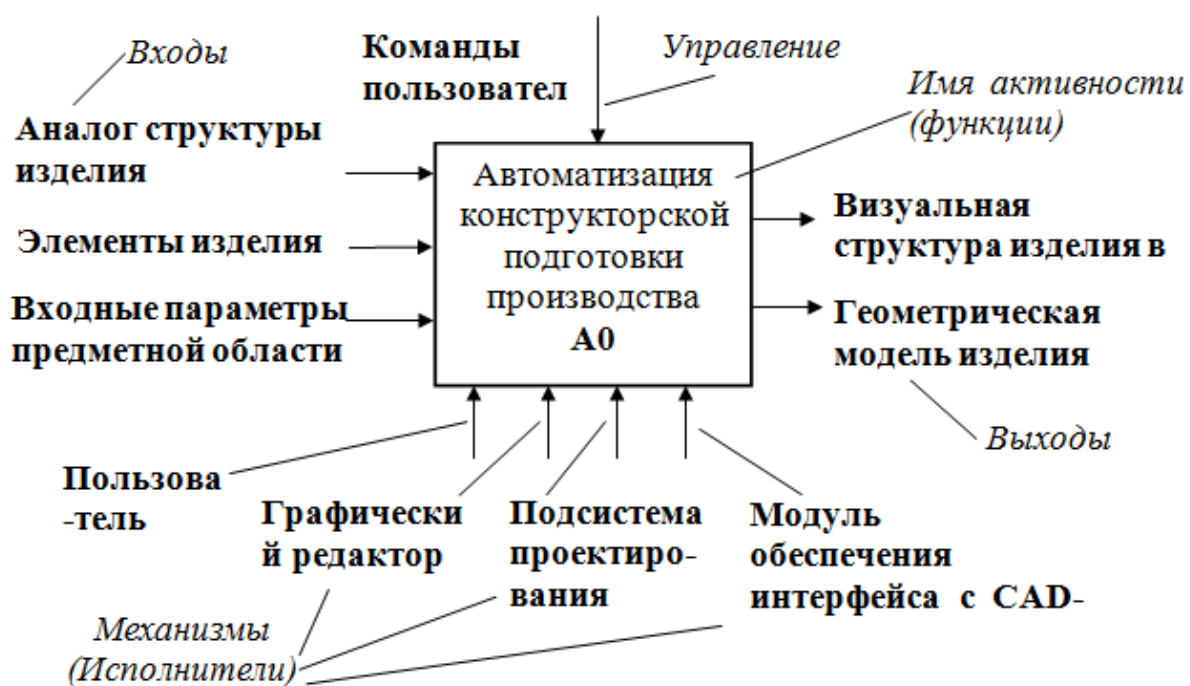


Рисунок 3.1 – SADT-диаграмма уровня 0, демонстрирующая функциональный (активный) блок и интерфейсные дуги (связи)

### 3.3.1 Диаграмма переходов состояний

При структурном подходе для целей моделирования систем вообще и структурного анализа в частности используют три группы инструментальных средств, иллюстрирующих:

- а) функции, которые система должна выполнять;
- б) отношения между данными;
- в) зависящее от времени поведение системы (аспекты реального времени).

Перечисленные средства дают полное описание системы независимо от её новизны. Производится построение логической функциональной спецификации, то есть подробное описание того, что должна делать система, без рассмотрения путей реализации (обеспечивается чёткое представление о конечных результатах).

Диаграмма переходов состояний (STD) отражает спецификации управления, которые предназначены для моделирования и документирования поведенческих аспектов систем, зависящих от времени или реакции на событие.

Диаграммы переходов состояний позволяют осуществлять декомпозицию управляющих процессов и описывают отношения между входными и выходными управляющими потоками при работе управляющего процесса.



Рисунок 3.2 – Детализирующая SADT-диаграмма первого уровня

STD моделирует последующее функционирование системы на основе ее предыдущего и текущего функционирования. Моделируемая система в любой момент времени находится в одном из конечного

множества заданных состояний. Система меняет свое состояние при определенных условиях (при возникновении заданных событий), причем все переходы должны быть точно определены. Таким образом, диаграмма состояний связывает события и состояния. Состояние определяется набором значений своих атрибутов. События передают информацию.

Диаграмма переходов состояний отражает последовательность событий, которое может иметь место при работе конкретной системы, т.е. иметь характер сценария работы системы.

Система изменяет свое состояние при выполнении определенных условий, т.е. имеет место переход из одного состояния в другое, который характеризуется как событием, обеспечивающим смену состояний, так и действием, которое представляет собой мгновенную операцию, связанную с событием.

Диаграммы переходов состояний (STD) предназначены для моделирования и документирования реакций системы при ее функционировании во времени.

STD позволяют осуществлять декомпозицию управляющих процессов в системе. STD описывают отношения между входными и выходными управляющими потоками на управляющем процессе. STD моделируют последующее функционирование системы на основе ее предыдущего и настоящего функционирования.

К объектам STD относятся: состояние, цель, переход, событие, условие.

Состояние - условие устойчивости для системы: способность системы сохранять свои функции без их произвольного изменения. Имя состояния - отражает реальную ситуацию, в которой находится система.

Цель - определяет будущее состояние по прошлому и текущему.

Переход - определяет перемещение системы из одного состояния в другое. Имя перехода - идентифицирует события, которые являются причиной перехода.

Событие - состоит из какого-либо управляющего потока (внешнего, внутреннего) и происходит при выполнении некоторого условия. При этом следует отметить: не все события вызывают переходы, события не всегда вызывают переходы, события не всегда вызывают переход в одно и то же состояние.

Условие - событие, вызывающее переход и названное именем перехода.

С переходом из одного состояния в другое может связываться действие или совокупность действий.

Действие при переходе - операция, которая выполняется при переходе. Действие может быть или физическим, или управляющим потоком.

Пример STD диаграммы для работы банкомата приведен на рисунке 3.3.

На диаграмме элементы нотации обозначаются следующим образом:

- а) состояния – как узлы (например, Ожидание);
- б) переходы – как дуги (например, Корректный пароль);
- в) условия - идентифицируются именем перехода (например, Корректный пароль);
- г) действия - отклики на события, которые "привязываются" к переходам, записываются под условием (например, Обеспечить требуемый сервис).

STD имеет только одно начальное состояние. Но система может иметь большое количество завершающих состояний.

Рекомендации. При построении STD желательно выполнять правила:

- а) строить диаграмму на наиболее высоком уровне;
- б) детализировать (обеспечивать иерархичность диаграмм);
- в) использовать те же термины (имена событий, действий, потоков, что и для DFD).

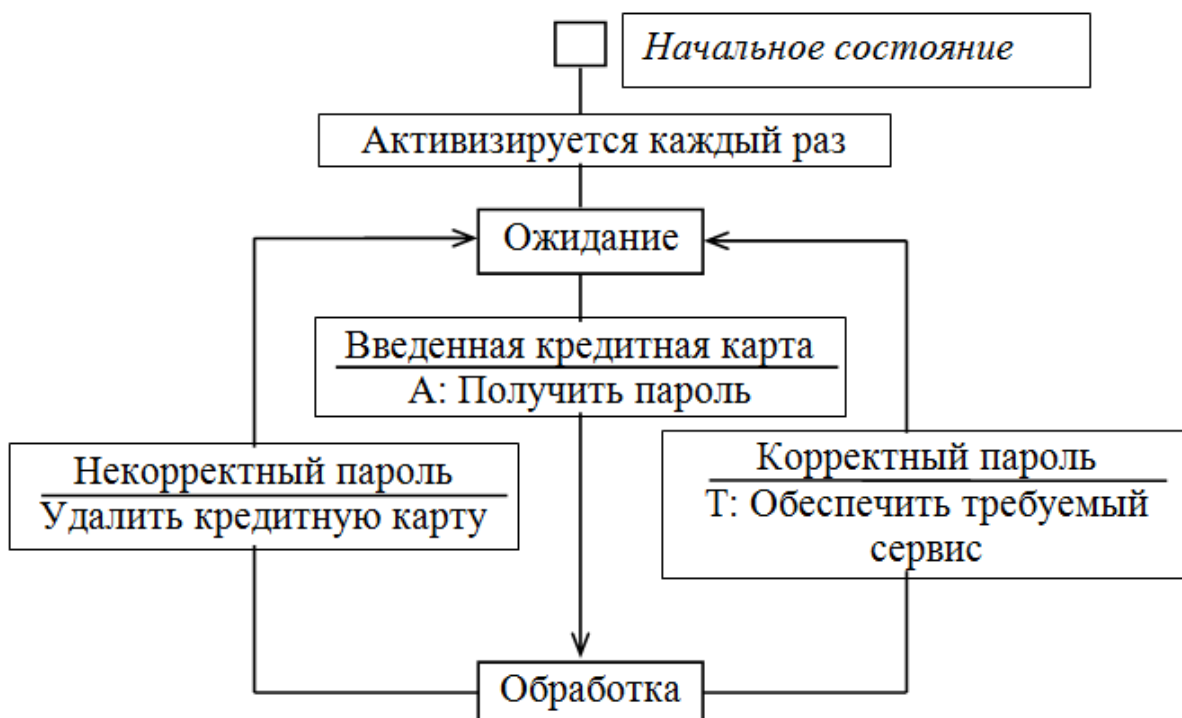


Рисунок 3.3 - Диаграмма переходов состояний (STD) для банкомата

Кроме STD диаграмм, для представления той же информации можно использовать таблицы переходов состояний (таблица 3.1).

Контроль STD диаграмм осуществляется по типу метода контрольных вопросов:

- а) все ли состояния определены и имеют имя?
- б) все ли состояния достижимы?

Для каждого состояния определяют:

- а) реагирует ли система соответствующим образом на все возможные условия, особенно ненормальные;
- б) все ли входные и выходные потоки управляющего процесса отражены в условиях и действиях на STD.

*Таблица 3.1 – Таблица переходов состояний для работы банкомата*

Текущее состояние	Условие	Действие	Следующее состояние
Начальное состояние	Активизируется каждый раз	-	Ожидание
Ожидание	Введенная кредитная карта	Получить пароль	Обработка
Обработка	Некорректный пароль	Удалить кредитную карту	Ожидание
Обработка	Корректный пароль	Обеспечить требуемый сервис	Ожидание

### 3.3.2 SADT – технология

Технология структурного анализа и проектирования SADT (Structured Analysis and Design Technique) - используется при решении широкого спектра задач: телефонные сети, системная поддержка и диагностика, долгосрочное и стратегическое планирование, автоматизированное производство и проектирование конфигурации компьютерных систем, обучение, встроенное программное обеспечение для оборонных систем, управление финансами и материально-техническим снабжением и др.

Рассмотрим активностную модель SADT, которая нашла наиболее широкое применение. В данной технологии используются понятия:

- а) предметы системы (как совокупности данных);
- б) активности системы (функции системы).

На диаграммах блоки означают активности и сопровождаются текстами на естественном языке, описывающим эти активности. Активности (блоки) выражают свои отношения через предметы системы (связи) (рисунок 3.4) [6].



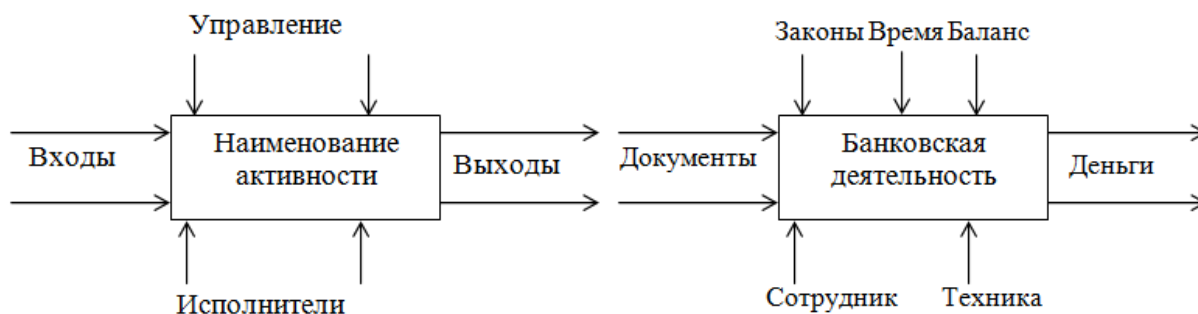


Рисунок 3.4 – Назначение надписей на блоках диаграммы и пример SADT-блока

Модель SADT объединяет диаграммы в иерархические древовидные структуры. В диаграмме 3-6 блоков, вместо одной громоздкой модели используются несколько взаимосвязанных моделей (декомпозиция), что обеспечивает структуризацию проблемы (системы).

Каждая сторона блока имеет своё назначение и показывает принципы функционирования блока:

- а) входы – преобразуются в выходы;
- б) управление – ограничивает и предписывает условия выполнения деятельности;
- в) исполнители – описывают, за счёт чего (кем) выполняются преобразования.

Дуги означают наборы предметов и сопровождаются текстом на естественном языке. Дуги могут соединяться и разветвляться.

Предметы состоят в четырех возможных отношениях (рисунок 3.5) с активностями (Вход, Выход, Управление, Исполнитель). Таким образом, стороны блока графически сортируют предметы, изображаемые дугами.

Размещение блоков на диаграмме производится по ступенчатой схеме в соответствии с их доминированием - влиянием одного блока на другой. Номера блоков проставляются обычно в соответствии с доминированием. Взаимовлияние проявляется в пересылке Выхода одной активности к другой - для дальнейшего преобразования, либо в выработке управляющей информации, которая предписывает, что должна делать другая активность.

Используется пять типов взаимосвязей (отношений) между активностями системы (рисунок 3.5).

При использовании SADT варианты диаграмм разрабатываются несколько раз для выбора лучшего.

Достоинства SADT – совмещение графа (диаграммы) (“активность - предмет”); отражение управления, обратной связи, исполнителей и одновременный показ доминирования.

На рисунке 3.6 приведен пример SADT диаграммы для системы «гидравлический пресс» [6].

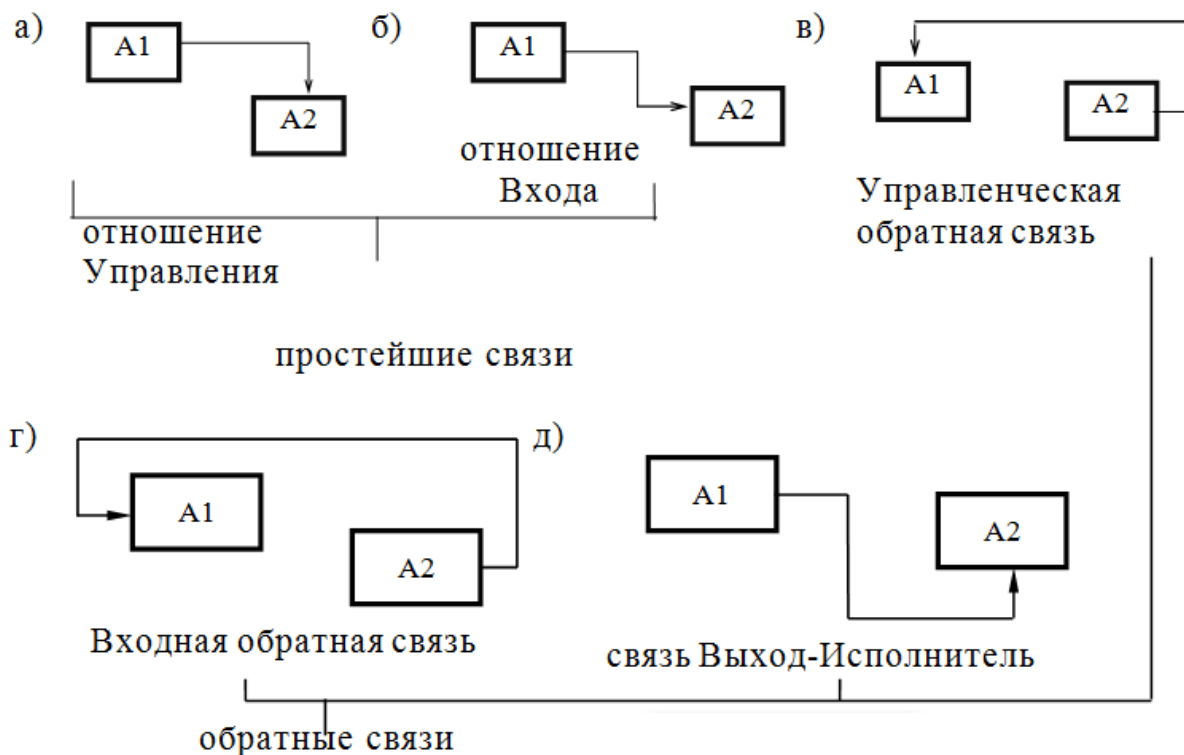


Рисунок 3.5 – Типы взаимосвязей активностей (A1, A2) системы

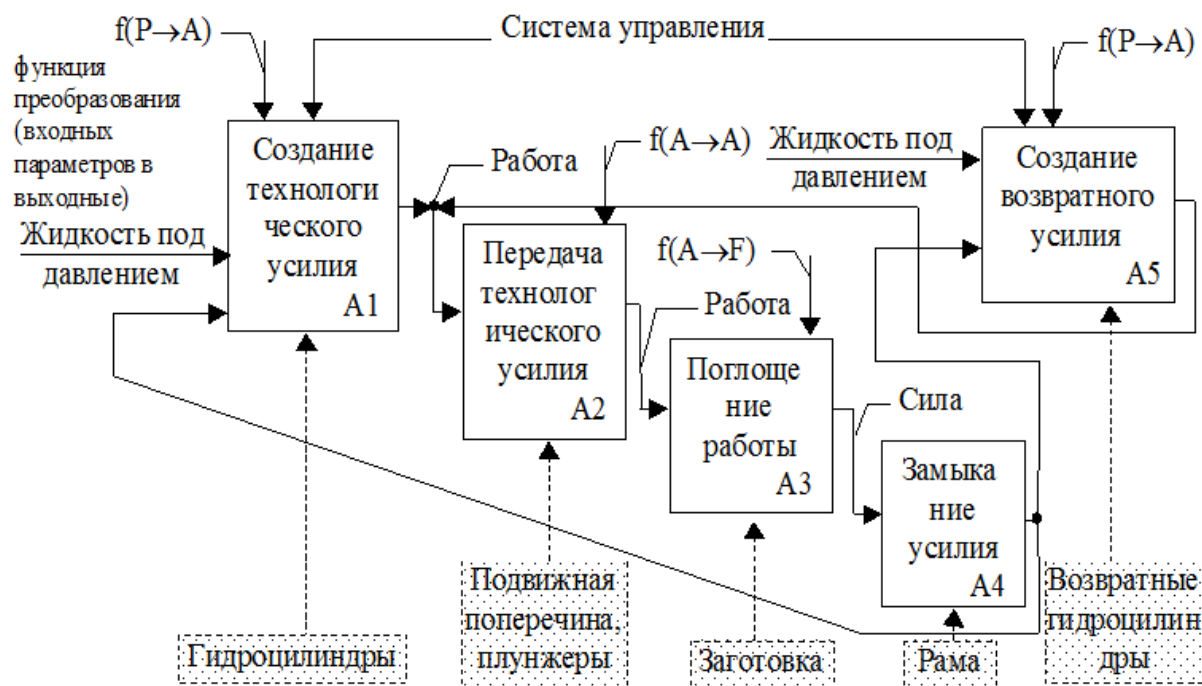


Рисунок 3.6 – Пример SADT диаграммы для гидравлического пресса

На диаграмме (рисунок 3.6) приведены основные активности A1-A5 (функции элементов) системы, а также предметы (жидкость, работа, сила). Преобразование предметов и их параметров

осуществляется в соответствии с заданными функциональными зависимостями  $f$ . Реализация активностей осуществляется «исполнителями», в данном случае это конструктивные решения (гидроцилиндры, рама и т.д.). SADT – диаграмма деятельности компании, осуществляющей распределения товаров по заказам, приведена на рисунке 3.7 [6].

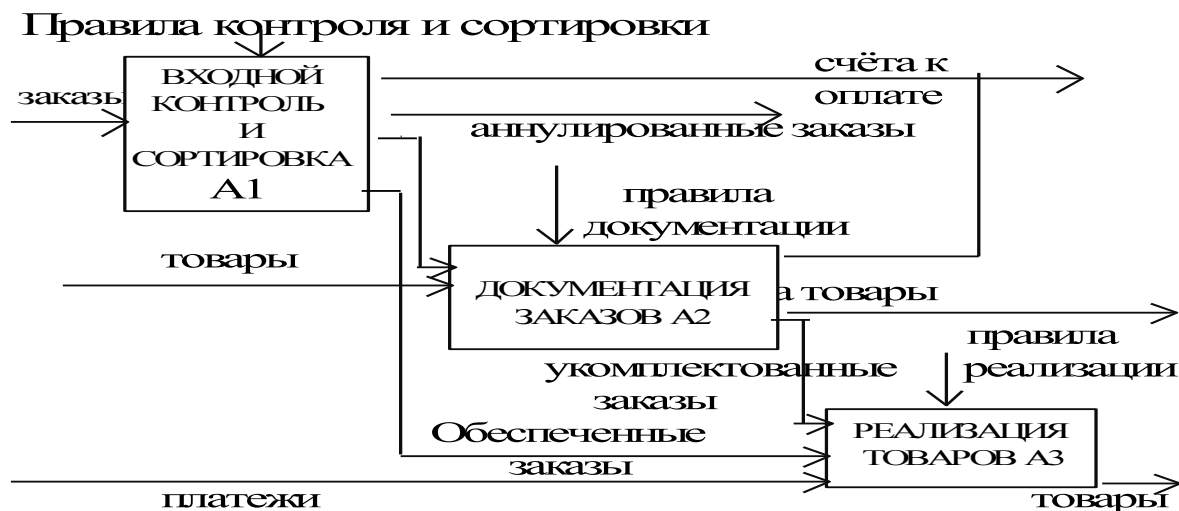


Рисунок 3.7 - Деятельность компании распределения товаров по заказам

### 3.3.3 Диаграммы потоков данных

Диаграммы потоков данных DFD (Data Flow Diagrams) – являются основным средством моделирования функциональных требований к проектируемой системе. С их помощью эти требования разбиваются на функциональные компоненты (процессы), связанные потоками данных и представляются в виде графа. Главная цель таких средств - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Для изображения DFD традиционно используют следующую нотацию (таблица 3.2) [34, 35].

Важную роль в модели играет специальный вид DFD - контекстная диаграмма, которая моделирует систему наиболее общим образом (на самом высоком уровне абстракции). Контекстная диаграмма моделирует (отражает) интерфейс связи системы с внешним миром, а именно - информационные потоки между системами и внешними сущностями, с которыми она должна быть связана. Она идентифицирует внешние сущности, а также, как правило, единственный процесс, отражающий главную цель или природу системы. Каждый проект имеет только одну контекстную диаграмму (0-го уровня)[36].

Таблица 3.2 – Обозначения на диаграмме потоков данных

Наименование и назначение элемента	Обозначение
<p><i>Поток данных</i> - механизм, использующийся для моделирования передачи информации (или даже физических компонентов) из одной части системы в другую. Изображаются именованными стрелками с ориентацией по направлению передачи (движения) информации. Движение возможно в одном направлении – однонаправленный поток, или в двух с обработкой и возвратом: моделируются двумя потоками или одним двунаправленным.</p>	<p>Имя →</p> <p>Имя ←→</p>
<p><i>Процесс</i> - продуцирует выходные потоки из входных в соответствии с действием, задаваемым именем процесса.</p> <p><i>Имя процесса</i> - содержит глагол в неопределенной форме с последующим дополнением (например, “Вычислить максимальную высоту”).</p> <p><i>Номер</i> - уникальный номер процесса для ссылок на него внутри DFD. Может использоваться совместно с номером диаграммы для получения уникального индекса модели.</p>	<p>Имя</p> <p>номер</p>
<p><i>Хранилище</i> (накопитель данных) - позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами.</p> <p><i>Имя хранилища</i> - идентифицирует его содержание и является существительным.</p> <p>Если поток данных входит или выходит из хранилища и соответствует его структуре, он должен иметь то же имя, которое имеет хранилище и это имя не нужно отражать на диаграмме возле потока данных.</p>	<p>Имя</p>
<p><i>Внешняя сущность</i> (терминатор) - представляет сущность вне контекста системы, являющуюся источником или приемником системных данных.</p> <p><i>Имя</i> - должно содержать существительное, (например, “Склад товаров”). Предполагается, что объекты, представленные такими узлами, не участвуют ни в какой обработке информации.</p>	<p>Имя</p>

Рекомендации для построения иерархии DFD:

- а) целесообразно на каждой диаграмме рассматривать от 3 до 6-7 процессов;
- б) не загромождать диаграмму несущественными на данном уровне деталями;
- в) декомпозицию потоков производить параллельно декомпозиции процессов (одновременно);
- г) выбирать ясные, отражающие суть дела имена процессов и потоков (без сокращений);
- д) однозначно определять функционально идентичные процессы только на самом верхнем уровне, где это необходимо, и ссылаться на эти определения на нижних уровнях;
- е) пользоваться простейшей диаграммной техникой;
- ж) отделять управляющие структуры (т.е. процессы) от обрабатывающих процессов, локализовать управляющие структуры.

### 3.3.4 Задание на лабораторную работу №34

**Тема:** Правила и последовательность описания технических объектов на этапе анализа.

**Цель работы:** научиться выполнять различные виды описаний технических объектов на этапе анализа.

#### *Теоретические сведения*

Существует несколько видов описаний по уровням и аспектам осмысления подхода к описаниям: параметрическое, морфологическое, функциональное, описание жизни объекта.

Основной принцип системного анализа - концепция целостности, предполагающая наличие системного эффекта, т.е. появление у системы свойств, не присущих ее частям. Причиной системного эффекта является наличие связей между элементами.

Возможны различные аспекты описания проектируемых объектов, разнообразные варианты декомпозиции объекта по характеру отображаемых свойств, т.е. декомпозиция производится с точки зрения различных аспектов описаний: функциональный, конструктивный, технологический.

Функциональный аспект связан с отображением основных принципов функционирования системы. Рассматривается характер физических, информационных и др. процессов в объекте. В качестве графического отображения используют принципиальные, функциональные, структурные, кинематические схемы и сопровождающие документы (спецификации).

Конструктивный аспект связан с реализацией результатов функционального проектирования, а именно - с определением геометрических форм элементов и их взаиморасположением в пространстве. В качестве графического изображения используют деревья «И-ИЛИ», графы.

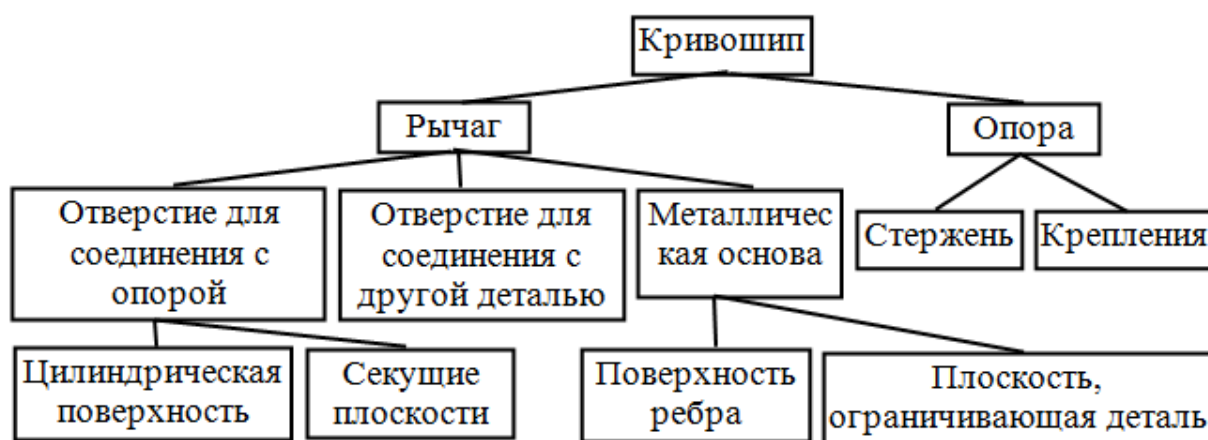
Технологический аспект относится к реализации результатов конструкторского проектирования, связан с определением методов и средств для подготовки изготовления и изготовления объектов. В качестве графического сопровождения используют деревья, графы, технологические карты.

Могут использоваться и другие аспекты. Всегда желательно выделять принцип разделения системы, а затем перечислять возможные варианты.

Ниже приведен пример описания кривошипа.

Параметрическое описание. Объект гладкий, металлический, серого цвета, без вкуса и запаха, средних размеров, имеет металлический блеск.

Морфологическое описание. Объект состоит из рычага и опоры. Рычаг имеет два отверстия. Первое предназначено для соединения с опорой, второе - для соединения с другими деталями. Опора представляет собой два металлических крепления, в отверстия которых вставлена ось для обеспечения вращения рычага. Дерево, отражающее конструктивный аспект описания, приведено на рисунке 3.8.



*Рисунок 3.8 – Дерево конструктивного исполнения объекта проектирования*

Функциональное описание. Функциональное описание объекта удобно представлять в виде графа (рисунок 3.9).



Рисунок 3.9 – Граф функционального описания объекта

Описание жизни объекта. В исходном состоянии рычаг не нагружен; когда к нему приложена внешняя нагрузка, рычаг передает ее, приходя при этом в движение (если нагрузка слишком мала, то рычаг остается в исходном состоянии); при снятии нагрузки кривошип возвращается в исходное состояние.

### Индивидуальное задание

Выполнить последовательно виды описаний для заданного объекта проектирования в соответствии с индивидуальным заданием (таблица 3.3): параметрическое, морфологическое, функциональное, описание жизни объекта.

Таблица 3.3 – Индивидуальные задания

№	Объект	№	Объект
1	Двигатель постоянного тока	11	Телефон
2	Автомобиль	12	Компьютер
3	Магнитофон	13	Телевизор
4	Жесткий диск	14	Лунный самоходный аппарат
5	Прокатный стан	15	Манипулятор "мышь"
6	Станок с ЧПУ	16	Кофейный автомат
7	Принтер	17	Микроволновая печь
8	Сканер	18	Башенный кран
9	Редуктор	19	Буровая установка
10	Видеокамера	20	Кухонный комбайн
11	Стиральная машина	21	Самолет-истребитель
12	Бумагоделательная машина	22	Экстендер
13	Шахтный подъемник	23	Гравировальная машина
14	Кондиционер	24	Токарный станок
15	Холодильная установка	25	Драглайн
16	Навигационный аппарат	26	Машина непрерывного литья заготовок

## ***Контрольные вопросы***

- 1 Какие существуют аспекты описаний объектов? Их характеристики.
- 2 Какие существуют виды описаний объектов? Их характеристики.

### 3.3.5 Задание на лабораторную работу №35

**Тема:** Анализ объекта проектирования как системы, построение дерева «И». Разработка графа связей элементов.

**Цель работы:** научиться представлять объект в виде совокупности взаимосвязанных элементов, образующих различные уровни иерархии.

### ***Теоретические сведения***

Разбиение объекта на элементы производится в зависимости от задач, стоящих перед разработчиком. Выделение элементов может быть выполнено:

- а) по функциональному назначению;
- б) по конструктивным признакам;
- в) по технологическим признакам;
- г) по последовательности сборочных операций;
- д) по другим критериям.

Наиболее часто при создании новых систем и объектов используется выделение функционального назначения элементов, а затем выбор их конструктивного исполнения.

При анализе существующих технических объектов (ТО) для построения иерархии элементов используют их конструктивное исполнение в соответствии с их функциональным назначением.

Узлом ТО называется множество конструктивно связанных деталей, в совокупности выполняющих хотя бы одну функцию по обеспечению работы других функциональных элементов ТО. Крупные, сложные узлы можно разделить на более простые подузлы для выполнения подчиненных функций и так далее с образованием иерархии элементов и функций.

Деталью ТО называется отдельное тело (элемент) из однородного материала, выполненное без использования сборочных операций, имеющее множество определенных форм и выполняющее хотя бы одну функцию в системе.

В зависимости от решаемой задачи деталь также может быть представлена в виде совокупности элементов ("разрезана" на "неделимые элементы").

Неделимым элементом называется часть детали (или сама деталь), имеющая минимальное число функций по обеспечению работы других



элементов, при любом делении которой появляются элементы, не имеющие самостоятельной функции или с одинаковыми функциями (шарик в подшипнике). Следует отметить, что степень деления и выделения элементов объекта связаны с решаемой задачей. Вид задачи определяет уровень абстракции, на котором рассматривается объект и степень детализации его элементов при разбиении и изучении.

В ряде случаев элементы ТО могут одновременно выполнять несколько функций.

Проектирование ТО - процесс, заключающийся в преобразовании исходного описания объекта в окончательное описание на основе выполнения комплекса работ исследовательского, расчетного и конструкторского характера. Проектирование ТО начинается с разработки задания на проектирование, которое отражает потребности общества в получении некоторого технического изделия. Задание представляется в виде тех или иных документов и является исходным (первичным) описанием объекта. ТЗ, как правило, содержит общие требования к разрабатываемому объекту, цель и задачи разработки, технико-экономическое обоснование проекта, календарный план и основные этапы разработки, ожидаемый результат работы.

Результатом проектирования, как правило, служит полный комплект документации, содержащий достаточные сведения для изготовления объекта в заданных условиях. Документация представляет собой окончательное описание объекта.

Преобразование исходного описания в окончательное порождает промежуточные описания, которые являются предметом рассмотрения с целью определения этапов и направлений дальнейшей работы. Процесс разработки ТО, как и все решения в жизни, учебе, производится поэтапно. Такие описания называются проектными решениями.

Проектирование, при котором все или часть проектных решений получают в процессе взаимодействия человека и ЭВМ, называют автоматизированным.

При проектировании сложных объектов и систем используют принципы системного анализа, основными из которых являются декомпозиция и иерархичность описания, многоэтапность и итерационность проектирования, типизация и унификация проектных решений и средств проектирования.

Описания ТО должны быть согласованы по сложности с возможностями восприятия человеком и оперирования описаниями в процессе их преобразования с помощью имеющихся средств проектирования.

Выполнить общее описание и преобразовать информацию, чтобы получить более качественное решение без выявления структуры объекта, очень трудно. Элемент системы усовершенствовать всегда проще, чем систему в целом.

Структурирование описаний и расчленение представлений о проектируемых объектах приводит к блочно-иерархической модели представления данных, а также выделению аспектов описаний.

Такое структурирование позволяет распределить (распараллелить) работы по проектированию между подразделениями проектных организаций (программистами при создании программных продуктов), способствует повышению производительности труда за счет специализации и, (отчасти, снижения требований к профессиональной подготовке).

В основе блочно-иерархического подхода к проектированию лежит выделение иерархических уровней (уровней абстрагирования) в представлениях об объекте.

На каждом уровне используются свои понятия системы и элементов. На верхнем уровне расположен сложный объект проектирования  $S$ , который рассматривается как система  $n$  взаимосвязанных и взаимодействующих элементов  $S_i$  ( $i=1...n$ ). Каждый элемент  $S_i$  в свою очередь рассматривается как состоящий из  $m$  взаимосвязанных и взаимодействующих элементов  $S_{ij}$  на следующем уровне ( $j=1...m$ ).

Такое разбиение продолжается до тех пор, пока оно не становится достаточным в рамках рассматриваемой задачи. Элементы, которые не подлежат делению, называют базовыми.

Принцип иерархичности - структурирование представлений об объектах проектирования по степени детализации описания.

Принцип декомпозиции (блочности) - разбиение представлений каждого уровня на ряд составных частей (блоков) для отдельного проектирования.

Блочно-иерархическая структура представлений об объектах в машиностроении:

- а) базовые элементы (детали - вал, шестерня ...);
- б) сборочные единицы (узлы - вал с подшипниками и шестерней...);
- в) агрегаты (станки, прессы...);
- г) комплексы (поточные линии, летательный аппарат...).

Кроме рассмотренной выше декомпозиции используют также декомпозицию описаний по характеру отображаемых свойств объекта. Такая декомпозиция приводит к появлению ряда аспектов (функциональный, конструкторский, технологический), которым соответствуют аналогичные виды проектирования.

Могут использоваться и другие аспекты описаний: например функциональный аспект разделить на электрический, механический, гидравлический, химический и др. В данном случае деление выполнено по физическим основам явлений в объекте. Всегда желательно выделять принцип деления, а затем перечислять возможные варианты.

В качестве графического представления описания объекта могут использоваться графы (деревья). Для автоматизации обработки

информации, представленной в виде графов, последние могут задаваться в виде матриц (матрицы смежности и матрицы инцидентий).

Матрица смежности для графа  $G(n)$  имеет вид  $M = [m_{ij}]$ ,  $(i, j = 1 \dots n)$ , где  $n$  - число узлов графа. При наличии связи между  $i$ -м и  $j$ -м узлом значение  $m_{ij} = 1$ , иначе  $m_{ij} = 0$ .

Матрица инцидентий имеет вид  $B = [bij]$   $(i = 1 \dots n-1, j = 1 \dots k)$ , где  $n$  - число узлов графа,  $k$  - количество ребер графа. Каждая строка соответствует одному узлу, кроме одного, который называется базовым (в качестве базового может быть выбран произвольный узел). Каждый столбец соответствует одному ребру. Если ребро  $j$  направлено из узла  $i$ , то  $b_{ij} = (+1)$ . Если ребро  $j$  направлено к узлу  $i$ , то  $b_{ij} = (-1)$ .

### **Индивидуальное задание**

Согласно предметной области технического объекта (согласно индивидуальному заданию таблицы 3.3) выполнить следующие пункты:

- а) словесное описание технического объекта;
- б) выделить элементы объекта: главный элемент и элементы, принадлежащие различным уровням иерархии (абстракции);
- в) указать, к какому типу функциональных элементов относятся выделенные элементы;
- г) обозначить и охарактеризовать связи между элементами;
- д) представить граф, содержащий элементы объекта на различных уровнях иерархии (И - дерево);
- е) построить матрицу смежности и инцидентий для полученного графа;
- ж) составить таблицу функционального описания элементов объекта (таблица 3.4).

*Таблица 3.4 – Функциональное описание элементов объекта*

Функции	Элементы системы			
	а	б	в	.....
	+	(+)		

Примечание.

- + - элемент выполняет функцию;
- (+) - элемент может выполнять функцию.

### **Контрольные вопросы**

- 1 В чем заключается сущность процесса проектирования ТО?
- 2 В чем заключается сущность иерархического подхода при описании объекта проектирования?

3 Перечислите и охарактеризуйте основные подходы к выделению элементов при декомпозиции.

4 Раскройте следующие понятия: «узел ТО», «деталь ТО», «неделимый элемент».

5 Перечислите и охарактеризуйте основные виды элементов при функциональном подходе к декомпозиции.

6 Что такое И-ИЛИ дерево технических решений?

7 Для чего используется дерево технических решений?

8 Какие бывают типы вершин И-ИЛИ дерева? Поясните смысл каждого типа вершин.

9 Что такое вариант технического решения? Как его получить из И-ИЛИ дерева технических решений?

10 Что такое мощность вариантов дерева решений? Как ее определить?

11 Опишите алгоритм определения мощности вариантов дерева решений.

### 3.3.6 Задание на лабораторную работу №36

**Тема:** Ассоциативные методы решения творческих (изобретательских) задач.

**Цель работы:** изучить методы преодоления психологической инерции, приобрести навыки создания новых, нестандартных ассоциаций и использования их при решении творческих задач.

#### *Теоретические сведения*

Формулировка задач, требующих творческого подхода для их решения, обычно не содержит указания на способ решения. Постановка задачи содержит задание, которое необходимо выполнить, но которое одновременно не может быть выполнено. Противоречие, содержащееся в поставленной задаче, называется административным.

Решение таких задач осуществляется путем раскрытия административного противоречия, выявления технического противоречия, а затем – физического противоречия и их устранения.

Техническое противоречие возникает, если при улучшении одной из частей системы другая недопустимо ухудшается.

Физическое противоречие возникает, если к одной и той же части системы предъявляются взаимопротивоположные требования.

Для уяснения задачи и раскрытия противоречий необходима специальная обработка условий задания, т.е. переформулировка задачи «для себя». Следует отметить, что такой этап в различных формах

присутствует во всех существующих методиках решения изобретательских задач: мозговой штурм, синектика, АРИЗ и др. Задача, как правило, формулируется в специальных терминах, что позволяет сократить изложение условий. Однако термины существенно сужают область поиска ответов, так как ассоциируются с относительно небольшим количеством стандартных конструктивных решений. Причины этого следует искать в особенностях мышления человека, в частности, образования ассоциаций. Установлено, что:

а) чем чаще объекты встречаются вместе, тем сильнее ассоциативная связь;

б) чем ярче эти объекты воспринимаются впервые, тем сильнее ассоциация;

в) знания, получаемые в организованной форме, усваиваются легче, так как упрощается образование ассоциаций.

Расширение поискового поля при решении задач требует освободиться от стандартных ассоциаций, преодолеть психологическую инерцию мышления.

Для преодоления психологической инерции необходимо:

а) помнить о ее существовании, что, однако, не всегда позволяет устранить ее влияние;

б) устранить из формулировки задачи специальные термины и использовать общие понятия;

в) выяснить типовые стандартные решения, обусловленные действием психологической инерции, и сознательно отказаться от этих направлений решения задачи;

г) фиксировать ход решения.

Существуют и другие причины, которые мешают поиску новых идей решения. Это - поспешное желание оценить решение, критиковать его, боязнь выделиться, недостаточная уверенность в своих способностях. Об этих причинах, обусловленных эмоциональным состоянием, а также о влиянии окружающих условий необходимо помнить и стараться устранить их воздействие на процесс решения задачи.

Решение задачи, как правило, представляется в виде конструкции, выполняющей требуемые по условиям задачи функции.

Системный анализ предполагает представление конструкции объектов в виде совокупности элементов, связанных между собой. Графическим изображением объекта является граф.

Постановка задачи обычно производится в неопределенной ситуации, когда существенные факторы приводятся в условиях задачи вместе с несущественными. Выделение существенных факторов связано с описанием параметров, структуры, функций элементов системы, определением внутренних связей между элементами.

Ассоциативные методы предназначены для решения творческих (изобретательских) задач путем создания и развития новых ассоциаций.

Создание новых ассоциаций и поиск на этой основе идей решения являются "обратной стороной" преодоления психологической инерции, которая связана со стандартными ассоциациями. К ассоциативным методам относятся:

- а) метод фокальных объектов;
- б) метод гирлянд случайностей;
- в) метод каталога и др.

Сущность этих методов состоит в поиске отдаленных нестандартных ассоциаций в разных областях знаний, переносе этих знаний и применении их при решении данной задачи. Приемы, используемые в эвристике, включают различные виды ассоциаций:

- а) аналогия - сходство предметов, явлений процессов по каким-либо признакам. Для создания аналогий используют ассоциации по сходству;
- б) инверсия - изменение обычного порядка на противоположный, обращение, выворачивание. Используют ассоциации по контрасту;
- в) преобразование в пространстве и времени - ассоциации по смежности;
- г) установление причинно-следственных связей в объекте, между объектом и средой. Ассоциации по смыслу.

Существуют также другие приемы, связанные с изменением основных показателей технических объектов:

- а) адаптация - приспособление известных принципов решения, свойств, процессов к заданным условиям;
- б) неология - перенос и использование таких новых свойств, показателей объектов, которые ранее не использовались в данной области техники;
- в) мультипликация - умножение, увеличение основных показателей;
- г) дифференциация - разделение функций объекта, его показателей, дробление и т.д.;
- д) интеграция - сближение, смешивание, соединение показателей;
- е) динамизация - изменение во времени основных показателей: веса, температуры, размеров и др.;
- ж) импульсация - импульсное, скачкообразное изменение показателей;
- з) идеализация - приближение показателей объекта к идеальным.

Для поиска ярких ассоциаций и генерирования новых идей применяют также различные метафоры. Ассоциативные методы позволяют быстро находить новые модификации известных решений, но редко дают результат при создании принципиально новых объектов. Развитие ассоциаций можно осуществить путем составления гирлянд ассоциаций, случайностей, метафор, используя ассоциативные связи между различными понятиями. Увеличения количества идей решения добиваются выполнением нескольких ассоциативных шагов.

### ***Индивидуальное задание***

Согласно заданному объекту предметной области (таблица 3.2) выполнить следующие пункты:

а) выполнить ассоциативный переход между заданными понятиями за несколько шагов, предложите несколько вариантов;

б) усовершенствовать конструкцию технического объекта с использованием метода фокальных объектов, а именно: выбрать три-четыре случайных объекта; составить списки признаков случайных объектов; присоединить полученные признаки к фокальному объекту, указать найденные идеи решения; развить полученные сочетания признаков путем дополнительных ассоциативных шагов (переходов); оценить полученные идеи, отобрать полезные решения.

в) проанализировать ход решения задачи.

г) указать какие противоречия устраняются в предложенных полезных решениях;

д) указать, в чем состояла сложность постановки задачи;

е) сформулировать задачу так, чтобы формулировка содержала техническое или физическое противоречие.

### ***Контрольные вопросы***

- 1 Что такое ассоциация? Какие типы ассоциаций вы знаете?
- 2 В чем сущность ассоциативных методов активизации творческого мышления?
- 3 Перечислите приемы усовершенствования технических объектов, основанные на использовании ассоциаций. В чем заключается сущность этих приемов?
- 4 Как образуются гирлянды ассоциаций, метафор?
- 5 С какой целью используют гирлянды ассоциаций?
- 6 Что такое ассоциативный переход?
- 7 Какие пункты выполнения работы относятся к методу фокальных объектов?
- 8 В чем сущность методу фокальных объектов?

#### 3.3.7 Задание на лабораторную работу №37

**Тема:** Функциональная декомпозиция объекта на основе SADT технологии (технология структурного анализа и проектирования).

**Цель работы:** научиться анализировать функциональную структуру деятельности заданных объектов, производить декомпозицию на основе активностной модели SADT технологии,

строить SADT-диаграммы для деятельности объектов, а также для элементов этих объектов.

### ***Индивидуальное задание***

Для заданного технического объекта (таблица 3.2) или вида деятельности выполнить следующие пункты:

- а) определить наборы предметов, предметы и активности (3-6 штук) системы;
- б) обосновать типы взаимосвязей активностей, иерархию соподчиненности;
- в) описать правила преобразований предметов;
- г) на основе проведенного анализа построить SADT диаграмму всей деятельности;
- д) выбрать один из блоков диаграммы (активность) и детализировать его – построить SADT диаграмму для выбранной активности.

### ***Контрольные вопросы***

- 1 Что представляет собой SADT технология, каковы ее особенности?
- 2 Что такое активности и предметы в понимании SADT, как они взаимосвязаны?
- 3 Опишите основные принципы построения диаграмм по технологии SADT?
- 4 Опишите и поясните типы взаимосвязей активностей. Каково их графическое представление на диаграммах?
- 5 Что является объектом применения SADT?
- 6 Достоинства и недостатки технологии SADT.

### 3.3.8 Задание на лабораторную работу №38

**Тема:** Разработка логической модели технического объекта. Диаграмма переходов состояний (State Transition Diagrams – STD).

**Цель работы:** научиться разрабатывать логические модели систем с представлением информации в виде STD диаграмм.

### ***Индивидуальное задание***

Для технического объекта (таблица 3.3) и видов его деятельности выбрать какой-либо управляющий процесс и описать его при помощи STD диаграммы и соответствующей ей таблицы переходов.



### ***Контрольные вопросы***

- 1 Перечислите методики графического представления структурной информации при разработке программного обеспечения.
- 2 Охарактеризуйте диаграммную методику STD и сравните ее SADT – технологией. Для чего используются эти методики?
- 3 Для чего предназначены диаграммы переходов состояний?
- 4 Перечислите объекты, которые выделяют при ее построении.
- 5 Опишите процесс построения и контроля STD-диаграмм.

#### 3.3.9 Задание на самостоятельную работу №7

**Тема:** Разработка логической модели технического объекта. Диаграмма потоков данных (Data Flow Diagrams – DFD).

**Цель работы:** научиться разрабатывать логические модели систем с представлением информации в виде DFD диаграмм.

#### ***Индивидуальное задание***

- 1 Разработать контекстную и одну детализирующую DFD диаграмму первого уровня для программно-методического комплекса по индивидуальному заданию.
- 2 Пронумеровать и описать все элементы диаграмм - потоки данных, процессы, хранилища и внешние сущности.
- 3 Выбрать какой-либо управляющий процесс и описать его при помощи STD диаграммы и соответствующей ей таблицы переходов.

### ***Контрольные вопросы***

- 1 Перечислите методики графического представления структурной информации при разработке программного обеспечения.
- 2 Охарактеризуйте диаграммную методику DFD и сравните с SADT – технологией. Для чего используются эти методики?
- 3 Перечислите и опишите объекты и понятия, которыми оперируют при построении DFD диаграмм.

## ЗАКЛЮЧЕНИЕ

Эффективная работа инженера в настоящее время немыслима без вычислительной техники, компьютерных технологий и развитых телекоммуникационных средств.

Квалифицированный пользователь, владеющий одним из языков программирования, может самостоятельно разработать и отладить комплекс программ, позволяющий реализовать на компьютере алгоритм поставленной задачи. В ряде случаев разработанная программа является узкоспециализированной, и такой подход требует больших трудозатрат на разработку и внедрение, сокращая долю творческого труда по решению конкретной технической задачи.

Для сокращения времени программирования создано большое количество программных продуктов, область применения которых значительна и для наиболее эффективного использования вычислительной техники необходимо правильно выбрать наилучший пакет программ еще на стадии решения прикладной задачи.

В пособии рассматривается структура компьютерных технологий по их использованию для различных технических задач в системах электропривода и электромеханики. Приведены описания современных программ, которые применяются при проектировании и эксплуатации электропривода.

При исследовании электромеханических систем, систем автоматического регулирования, вычислительных математических задач целесообразно применение пакета Matlab с обширными предметно-ориентированными библиотеками и инструментом визуального моделирования Simulink. Для визуального моделирования и моделирования совместно с аппаратурой исследователю-электромеханику удобен Vissim. Аналитические преобразования и их автоматизацию позволяет выполнять пакет Mathcad. Использование интегрированных программных систем автоматизации математических расчетов (Matlab, Maple, Mathematica, Mathcad и др.) позволяют решать поставленные задачи на входном языке, который максимально приближен к естественному математическому языку поддерживаются и развиваются за счет появления новых версий, введения новых модулей, блоков. Значительная часть программных продуктов позволяет проектировать и исследовать электрическое и механическое оборудование (Pspice, LabView, MVS, AVR и др.), сокращая временные и материальные затраты.

Рассматриваются также технологии объектно-ориентированного проектирования технических объектов, приводятся рекомендации по выполнению анализа объектов автоматизированного проектирования, разработке компонентов информационной модели и алгоритмов проектирования, а также по применению современных диаграммных методик представления информации о логической модели программной системы.

Современные программные средства позволяют провести весь цикл математического исследования электромеханических систем любой сложности: от непосредственного решения задачи (аналитического или численного) до подготовки технической документации к печати, повышают культуру инженерного мышления, имея большое научное и прикладное значения.

## СПИСОК ЛИТЕРАТУРЫ

- 1 **Доманов, А. В.** Компьютерные технологии в электроприводе / А.В. Доманов. – Ульяновск : УлГТУ, 2006. – 112 с.
- 2 **Каймин, В.А.** Информатика: учебник. – 2-е изд., перераб. и доп. – М.: ИНФРА-М, 2001. – 272 с. – ISBN 5-16-000612-5.
- 3 Информатика. Базовый курс. 2-е изд. / под. ред. С. В.Симоновича. – СПб.: Питер, 2005 – 640 с.: ил. – ISBN 5-94723-752-0.
- 4 **Мозговой, М.В.** Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. / М.В. Мозговой – СПб.: Наука и техника, 2006. – 320 с.: ил. – ISBN 5-94387-224-8.
- 5 Информатика / А.В. Могилев, Н.И. Пак, Е.К. Хённер. – М.: Академия, 2004. – 848 с.
- 6 Методические указания к курсовому и дипломному проектированию «Анализ объекта автоматизированного проектирования, разработка информационной модели и алгоритмов проектирования» / Сост. А.Ф.Тарасов. – Краматорск: ДГМА, 2001. – 31 с.
- 7 **Колесников, Л.А.** Основы теории системного подхода / Л.А. Колесников. – К.: Наук. думка, 1980. – 176 с.
- 8 Теоретические основы САПР / Корячко В.П., Курейчик В.М., Норенков И.П. – М.: Энергоатомиздат, 1987. – 400 с.
- 9 **Овчинников, В.В.** Архитектуры распределения информационно-вычислительных систем / В.В. Овчинников. – М.: Энергоатомиздат, 1988. – 382 с.
- 10 **Петренко, А.И.** Основы построения САПР / А.И. Петренко, О.И. Семенов. – К.: Выща шк., 1984. – 296 с.
- 11 **Дьяконов, В.П.** МАТЛАБ 5.0/5.3. Система символьной математики / В.П. Дьяконов, И.В. Абраменкова. – М.: Нолидж, 1999. – 640 с.
- 12 **Дьяконов, В. П.** Matlab 6: учебный курс. – СПб.: Питер, 2001. – 592 с.: ил.
- 13 **Половко, А.М.** Matlab для студента / А.М. Половко, П.Н. Бутусов. – СПб.: БХВ-Петербург, 2005 – 320 с.: ил. – ISBN 5-94157-595-5.
- 14 **Черных И. В.** Simulink: Инструмент моделирования динамических систем / И. В.Черных.– М.: Диалог-МИФИ, 2003. – 496 с.
- 15 Matlab в інженерних та наукових розрахунках: монографія / О.Ф. Дащенко, В. Х. Кирилов, Л.В. Коломієць, В. Ф. Оробей. – Одеса: Астропринт, 2003. – 214 с. – ISBN 966-549-016.
- 16 Методичні вказівки до лабораторних робіт з курсу «Введення в інформатику» (для студентів спеціальності 6.0922.03 «Електромеханічні системи автоматизації і електропривод» очно-заочної форми навчання)/ Укл.: О.І. Толочко, Г.С. Чекавський, О.В. Песковатська. – Донецьк: ДонНТУ, 2006. – 60 с.

- 17 Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М. – СПб.: Физматлит. – 2001. – 630с.
- 18 Розв'язування задач електромеханіки в середовищі пакетів MathCAD і MATLAB: навчальний посібник / А.О. Лозинський, В.І. Мороз, Я. С. Паранчук. – Львів: Видавництво Державного університету “Львівська політехніка”, 2000. – 166 с.
- 19 **Дьяконов, В.П.** MathCAD 2000: учебный курс / В.П. Дьяконов. – СПб: Питер 2000. – 673 с.
- 20 **Гурский, Д.А.** Вычисления в MathCAD / Д.А. Гурский. – Минск: Новое знание, 2003. – 814 с.
- 21 **Дьяконов, В. П.** MathCAD 8/2000: специальный справочник / В. П. Дьяконов. – СПб: Питер, 2001. – 592 с.
- 22 **Клиначев, Н. В.** Моделирование обыкновенных линейных систем: руководство к лабораторным работам в пакетах VisSim и Workbench / Н. В. Клиначев.– Челябинск, 2001. – 35 с.
- 23 **Клиначев, Н. В.** Моделирование систем в программе VisSim / Н. В. Клиначев. – 2001. – 50 с.
- 24 B2 Spice A/D 4.2. User`s manual: 2003. – 256 с.
- 25 LabView 7 Express. Вводный курс – М.: Приборкомплект, 2003. – 42 с.
- 26 **Курилин А. И.** AVR – микроконтроллеры: семь ярких лет становления. Что дальше? Часть 3 / А. И. Курилин, Р. Н. Золотуха // Компоненты и технологии. – 2005, №1. – 45 с.
- 27 **Андреев, Е.** SCADA-системы: взгляд изнутри / Е. Андреев. – М.: РТСофт, 2004. – 176 с.
- 28 **Пьявченко, Т.А.** Проектирование АСУТП в SCADA-системе / Т.А. Пьявченко. – Таганрог: Технологический институт ЮФУ, 2007 – 84 с.
- 29 Электропривод и сетевые технологии: доклады научно-практического семинара. – М.: изд-во МЭИ, 2003. – 144 с.
- 30 **Федоров, Ю.Н.** Справочник инженера по АСУТП: проектирование и разработка / Ю.Н. Федоров, М.: Инфра Инженерия, 2008. – 928 с. – ISBN 978-5-9729-0019-0.
- 31 **Нестеров, А.Л.** Проектирование АСУТП / А. Л. Нестеров. – М.: ДЕАН, 2006. – 552 с. – ISBN 5-93630-530-9.
- 32 **Калянов, Г.Н.** CASE. Структурный системный анализ. Автоматизация и применение / Г.Н. Калянов. – М.: ЛОРИ, 1996. – 242 с.
- 33 **Буч, Г.** Объектно-ориентированное проектирование / Г. Буч. – Киев: Диалектика, 1992. – 519 с.
- 34 **Ларман, К.** Применение UML и шаблонов проектирования / К. Ларман. – 2-е изд. – М.: Изд. дом «Вильямс», 2002. – 624 с.
- 35 **Маклаков, С.В.** BPWin и EPWin CASE средства разработки информационных систем / С.В. Маклаков. – М.: Диалог-МИФИ, 1999. – 346 с.

36 Буч, Г. Язык UML: Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон; Пер. с англ. – М.: ДМК Пресс, 2001. – 432 с.

## ПРИЛОЖЕНИЕ А

### *Требования к оформлению лабораторных работ*

Отчеты по выполненным работам оформляются в соответствии с методическими указаниями “Структура и правила оформления текстовых документов” на основе ДСТУ 3008.95 “Документация, отчеты в сфере науки и техники. Структура и правила оформления”. Отчеты оформляются на отдельных листах формата А4 (с текстовой информацией, набранной на компьютере и распечатанной на одной стороне листа). Листы должны иметь рамку, согласно стандарту, быть пронумерованы. В конце семестра после защиты всех работ отчеты сшиваются с титульным листом и предоставляются на кафедру.

### *Содержание отчета по лабораторной работе*

- 1 Фамилия, имя, отчество и группа студента.
- 2 Номер и название работы.
- 3 Цель работы.
- 4 Индивидуальное задание.
- 5 Описание пунктов выполненного задания с поясняющими блок-схемами, диаграммами, таблицами, программным кодом (Matlab), распечаткой рабочей области (Mathcad), графиками и рисунками.
- 6 Результаты работы программы.
- 7 Выводы.

Пример оформления первого листа отчета по лабораторной работе приведен на рисунке А.1. Второй лист должен содержать рамку, но без верхней шапки.

ДГМА	Факультет автоматизации машиностроения и информационных технологий	Группа ЭСА11-1
Кафедра ЭСА	Дисциплина: «Современные программные средства в электроприводе»	Выполнил: <i>Ф.И.О. студента</i>
Дата выполнения:	Дата защиты:	Рейтинг:

### *Лабораторная работа № 1*

**Тема:** Численное решение оптимизационных задач.

**Цель работы:** изучить особенности оптимизации инженерных задач, приобрести навыки решения оптимизационных задач в среде программного пакета Matlab.

#### *Индивидуальное задание (Вариант 1)*

Найти и вывести на печать координаты и значение минимума функции двух переменных с учетом координат начальной точки поиска  $f(x,y) = 2x^3 - y + 5$ .

#### *Ход работы:*

- 1 .....
- 2.....
- 3.....

**Вывод:** в ходе выполнения лабораторной работы было выполнено....., были приобретены навыки .....

					ЭСА11-1.СПС.01.ЛР1	Лист
						1
Изм.	Лист	№ докум.	Подпись	Дата		

*Рисунок А.1 – Пример оформления первого листа лабораторной работы*