

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Транспортные и технологические машины»

ИНФОРМАТИКА

*Методические рекомендации к лабораторным работам
для студентов специальности
1-37 01 02 «Автомобилестроение (по направлениям)»
очной формы обучения*



УДК 004.4
ББК 32.973
И74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой ТТМ «08» октября 2019 г., протокол № 3

Составитель ст. преподаватель В. И. Семчен

Рецензент канд. техн. наук, доц. А. Е. Науменко

Методические рекомендации разработаны на основании рабочей программы по дисциплине «Информатика» для обучающихся по специальности 1-37 01 02 «Автомобилестроение (по направлениям)» очной формы обучения и предназначены для использования при выполнении лабораторных работ.

Учебно-методическое издание

ИНФОРМАТИКА

Ответственный за выпуск

И. В. Лесковец

Редактор

А. А. Подошевка

Компьютерная верстка

Н. П. Полевничая

Подписано в печать .Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. .Уч.-изд. л. . Тираж 56 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, Могилев.

© Белорусско-Российский
университет, 2019



Содержание

Введение.....	4
1 Технические средства ПК.....	5
2 Графический интерфейс ОС Windows. Создание, копирование, перемещение и удаление каталогов и файлов. Хранение и защита данных.....	6
3 Основы алгоритмизации инженерных задач.....	9
4 Программный комплекс MATLAB	10
5 Запись выражений на языке MATLAB. Приоритет операций. Стандартные функции языка.....	13
6 Формирование векторов и матриц с требуемыми свойствами. Операции с элементами и блоками элементов матриц	14
7 Создание простейших программ. Реализация линейного алгоритма на языке MATLAB. Организация диалога с пользователем. Отладка программы. m-файлы	16
8 Программирование разветвляющегося алгоритма на языке MATLAB ...	17
9 Программирование циклических алгоритмов на языке MATLAB.....	20
10 Построение графиков в декартовых и полярных координатах	22
11 Организация вычислений с использованием функций пользователя	25
12 Создание презентации по изученному материалу.....	27
13 Оформление технического документа с использованием текстового редактора. Включение специальных объектов. Составление оглавления документа	29
14 Решение инженерных задач с использованием электронных таблиц. Визуализация результатов. Оформление таблицы	32
15 Решение систем ЛАУ численными методами. Реализация методов Гаусса и LU-разложения.....	37
16 Решение нелинейных алгебраических уравнений и систем. Оценка точности результата	38
17 Интерполяция и аппроксимация экспериментальных данных методом наименьших квадратов.....	40
18 Вычисление определенного интеграла методом трапеции и методом Симпсона.....	41
19 Решение ОДУ и их систем. Визуализация результатов вычислений	44
20 Численное дифференцирование функции одной переменной. Моделирование случайных величин с заданными свойствами. Визуализация результатов.....	45
21 Создание презентации по изученному материалу.....	47
Список литературы	48



Введение

Информатика входит в цикл общепрофессиональных и специальных дисциплин и обеспечивает подготовку обучающихся по использованию современных базовых компьютерных технологий в процессе обучения и в последующей профессиональной деятельности.

Одним из важных видов учебной деятельности обучающихся, предусмотренных учебным планом дисциплины, является выполнение лабораторных работ.

Целью выполнения лабораторных работ является получение практических навыков:

- в использовании программных средств общего назначения;
- построения математических моделей;
- разработки алгоритмов и программ на алгоритмическом языке для решения задач.

В лабораторных работах решаются задачи вычислительного характера.

Цель методических рекомендаций – оказание помощи обучающимся при выполнении лабораторных работ. В методических рекомендациях изложено содержание каждой работы, даны краткие теоретические сведения по решаемой задаче и порядок выполнения работы.



1 Технические средства ПК

Цель работы: изучить структуру, назначение и характеристики аппаратных средств ПК.

1.1 Основные положения

Конфигурация ЭВМ – особенности ее конструкции, включая архитектуру, состав и характеристики основных ее составных частей и вспомогательных (периферийных) средств, а также организацию связей между ними.

Архитектура ЭВМ – общий принцип построения и организации работы, включая определение функционального состава основных узлов и блоков, а также структуры управляющих и информационных связей между ними, обеспечивающих реализацию заданных целей и характеристик.

Конфигурация ЭВМ как при их проектировании, так и выборе определяется составом и сложностью задач, на которые они рассчитаны.

Основные принципы логического устройства современных ЭВМ определил фон Нейман (рисунок 1.1).

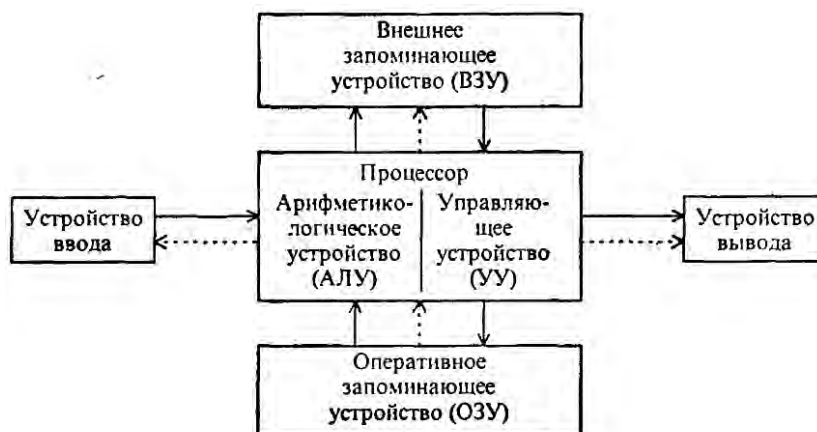


Рисунок 1.1 – Общая структура ЭВМ

Понятие «минимальная (базовая) конфигурация ЭВМ» связывается с конкретным типом центрального процессора, минимальными размерами внутренней и внешней памяти, клавиатурой и монитором, минимальным набором устройств, обеспечивающих работу с ЭВМ.

Основой электроники системного блока является материнская плата, характеристики и функциональность которой определяются набором микросхем системной логики – чипсетом. Процессор получает команды, выполняет предписанные каждой командой действия и затем переходит к выборке следующей команды, обменивается данными с внешними устройствами и связан с другими устройствами шинами.

Оперативное запоминающее устройство (ОЗУ), или оперативная память, относится к классу энергозависимой, выступает в качестве буфера между

накопителями и процессором и предназначена для размещения запущенных программ, хранящаяся в ОЗУ информация изменяется в процессе работы. Постоянное запоминающее устройство (ПЗУ) – энергонезависимая память в рабочем режиме допускает только считывание информации, предназначена для хранения одной и той же заранее определенной информации (программы).

Накопители используются для хранения информации после выключения компьютера.

Устройство ввода-вывода – это компонент типовой архитектуры компьютера, предоставляющий компьютеру возможность взаимодействия с пользователем.

1.2 Порядок выполнения работы

- 1 Выполнить запуск ЭВМ.
- 2 Определить конфигурацию ЭВМ, на которой выполняется работа.
- 3 Определить доступные параметры основных устройств.
- 4 Изучить основные принципы использования устройств ввода.
- 5 Завершить работу с ЭВМ.

Контрольные вопросы

- 1 Какие элементы включает общая структура ЭВМ?
- 2 Что такое архитектура ЭВМ?
- 3 Что такое конфигурация ЭВМ?
- 4 Что такое ОЗУ, для каких целей оно служит?
- 5 Что такое ПЗУ, для каких целей оно служит?
- 6 Каковы основные устройства ЭВМ и их характеристики?

2 Графический интерфейс ОС Windows. Создание, копирование, перемещение и удаление каталогов и файлов. Хранение и защита данных

Цель работы: изучить структуру и основные элементы интерфейса, получить практические навыки настройки и работы с ОС Windows и файловой системой, средствами хранения и защиты данных.

2.1 Основные сведения

Операционная система (ОС) – это совокупность программных средств, осуществляющих управление ресурсами ЭВМ, запуск прикладных программ и их взаимодействие с внешними устройствами и другими программами, а также обеспечивающих диалог пользователя с компьютером.



К основным элементам интерфейса ОС Windows относят рабочий стол, панель задач и главное меню «Пуск».

Структура окна программы в ОС Windows включает строку заголовка с кнопками управления окном, главное меню, панели инструментов, рабочую область и строку состояния.

Основные настройки элементов интерфейса и параметров операционной системы вызываются командой «*Параметры*» на вкладке «*Файл*» и дублируются в контекстных меню, вызываемых щелчком правой кнопкой мыши на любом объекте. Состав контекстного меню зависит от объекта, на котором оно вызвано.

В Windows имеется программа «*Проводник*», с помощью которой можно управлять файлами и папками.

Выделение файлов или папок:

- щелкнуть мышью по имени;
- нажать клавишу «*Ctrl*» и, удерживая ее, щелкнуть все нужные объекты;
- щелкнуть первый файл диапазона, нажать клавишу «*Shift*» и, удерживая ее, щелкнуть последний файл диапазона;
- диапазон файлов указать выделением рамкой с помощью мыши.

Снятие выделения файлов и папок:

- щелкнуть мышкой по имени выделенного файла;
- нажать клавишу «*Ctrl*» и, удерживая ее, щелкнуть все нужные объекты, с которых снимается выделение;
- щелкнуть мышью в поле проводника для отмены выделения всех файлов.

Для создания, копирования, удаления, перемещения и изменения файлов и папок необходимо:

- выделить файлы и папки (для создания шаг пропустить);
- выбрать необходимую команду в меню окна, в контекстном меню, панели инструментов, нажать соответствующие клавиши на клавиатуре, воспользоваться перетаскиванием;
- выполнить дополнительные действия, если это необходимо.

Если при удалении удерживать клавишу «*Shift*», объект будет удален без помещения в папку «*Корзина*».

Для восстановления удаленного файла или папки:

- раскрыть папку «*Корзина*»;
- выделить файлы или ярлыки, которые требуется восстановить;
- в выбрать команду «*Восстановить*».

Windows позволяет работать одновременно с несколькими приложениями. В каждый момент времени только одно окно является активным, заголовок активного окна выделен. Для переключения между окнами:

- щелкнуть мышью в любом месте видимого окна;
- щелкнуть мышью по кнопке приложения на панели задач;
- нажать клавишу «*Alt*» или «*Win*» и, удерживая ее, нажимать клавишу «*Tab*», пока не появится название нужной программы.



Информационной безопасностью называют меры по защите информации от неавторизованного доступа, разрушения, модификации, раскрытия и задержек в доступе.

Защита информации – это деятельность по предотвращению утраты и утечки защищаемой информации.

Информационная безопасность обеспечивает конфиденциальность критической информации, целостность информации и связанных с ней процессов, доступность информации, когда она нужна, учет всех процессов, связанных с информацией.

Средства защиты информации:

- контроль доступа с помощью пароля и идентификации личности;
- разграничения доступа в зависимости от полномочий;
- дублирование каналов связи и подключение резервных устройств;
- криптографические преобразование информации.

Учетная запись пользователя – это запись, которая содержит сведения, необходимые для идентификации пользователя при подключении к системе, а также информацию для авторизации и учета.

2.2 Порядок выполнения работы

1 Изменить настройки рабочего стола, панели задач, меню пуск.

2 Закрепить в меню «Пуск» значок программ Word, Excel, MATLAB. Поместить в меню «Пуск» – «Все программы» папку, названную по фамилии исполнителя работы.

3 Создать ярлык на рабочем столе для файла методических указаний по лабораторным работам и программ Word, Excel, MATLAB.

4 В окне проводника создать на доступном для записи диске папку по фамилии автора, скопировать в эту папку материалы по дисциплине, просмотреть содержимое и уровни иерархии выбранной папки, изменить режим вывода, размеры окон и столбцов проводника, переименовать созданную папку, добавив номер группы.

5 Создать на рабочем столе папку, названную по фамилии автора, и в нее скопировать созданную на диске папку.

6 Запустить Word, Excel, MATLAB. Изменить размеры и положение окон, выполнить переключение между окнами с помощью различных методов.

7 Запустить диспетчер задач, завершить работу Excel, MATLAB.

Контрольные вопросы

1 Каковы основные функции ОС?

2 Что относят к интерфейсным элементам управления ОС?

3 Какова структура окна приложения в Windows?

4 Как производится настройка интерфейса Windows?

5 Какова структура и основные приемы работы с проводником?

6 Каковы цели и средства информационной безопасности?



3 Основы алгоритмизации инженерных задач

Цель работы: получить практические навыки по составлению различных типов алгоритмов для решения задач.

3.1 Общие сведения

Алгоритм – это краткое и точное предписание о порядке выполнения действий над исходными данными для получения требуемого результата.

Наиболее распространены следующие формы представления алгоритмов:

- словесная (записи на естественном языке);
- графическая (изображения из графических символов).

В графическом представлении алгоритмы изображаются в виде блок-схемы, дополненной элементами словесной или математической записи.

Схема алгоритма включает блочные символы, соединенные между собой линиями, указывающими порядок выполнения операций. Блочные символы стандартизированы и различаются по типу выполняемых действий (ГОСТ 19.002–80 и 19.003–80, ИСО 2636–73 или ИСО 1028–73).

Шаги алгоритма должны читаться слева направо и сверху вниз независимо от направления потоков информации. Логика алгоритма должна опираться на минимальное число достаточно простых управляющих базовых структур. При разработке схем алгоритмов необходимо соблюдать некоторые требования:

- все линии быть замкнутыми или иметь разрывы, помеченные соединителями;
- должны четко прослеживаться потоки информации, не допускается пересечения линий, при передаче управления в схеме «снизу–вверх» или «справа–налево» линии обязательно помечают стрелками;
- не допускается передача управления в никуда.

Линейными называют алгоритмы, в которых операции выполняются последовательно одна за другой в единственном порядке следования.

Алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений, называют разветвляющимися. Каждое возможное направление вычислений называется ветвью.

Алгоритм циклической структуры предусматривает многократное повторение действий в одной и той же последовательности по одним и тем же зависимостям (тело цикла), но при разных значениях исходных данных.

Специально изменяемый по заданному закону параметр, входящий в тело цикла, называется переменной цикла. Переменная цикла используется для подготовки очередного повторения цикла и отслеживания условий его окончания.

Циклы, в теле которых нет разветвлений и встроенных в них циклов, называют простыми, в противном случае их относят к сложным. Циклические алгоритмы делят на детерминированные и итерационные.



Циклы, в которых число повторений заранее известно из исходных данных или определено в ходе решения задачи, называют детерминированными.

Циклы, в которых число повторений зависит от некоторого логического условия, изменяющегося в ходе решения задачи, называют итерационными.

3.2 Порядок выполнения работы

- 1 Получить индивидуальное задание по вариантам.
- 2 Составить словесные описания алгоритмов.
- 3 Составить алгоритмы с помощью языка блок-схем.

Контрольные вопросы

- 1 Что такое алгоритм?
- 2 Какие способы описания алгоритмов Вам известны?
- 3 Какие требования предъявляются к алгоритмам?
- 4 Перечислите основные правила построения алгоритмов.
- 5 Назовите основные типы графических элементов.
- 6 Какие основные виды структуры алгоритмов Вам известны?

4 Программный комплекс MATLAB

Цель работы: изучить структуру окна системы MATLAB, получить представление об основных возможностях системы MATLAB, приобрести навыки выполнения вычислений в командном режиме работы MATLAB.

4.1 Общие сведения

В состав MATLAB входят интерпретатор команд, графическая оболочка, редактор-отладчик, библиотеки команд, компилятор, символьное ядро для проведения аналитических вычислений, математические библиотеки MATLAB, генератор отчетов и инструментарий.

Интерфейс (рисунок 4.1) многооконный и имеет ряд средств прямого доступа к различным компонентам системы. Вся работа организуется через командное окно («*Command Window*»), которое появляется при запуске программы. В процессе работы данные располагаются в памяти («*Workspace*») в виде матриц.

В начале работы с системой MATLAB следует выбрать активную папку, которая будет использоваться при выполнении работы с файлами. С этой целью необходимо мышью нажать на кнопку «*Browse for folder*».

Командное окно разделено на две зоны: зону просмотра и зону редактирования. Зона редактирования находится в строке командного окна, отмеченной знаком «>>», после которого записывается вычисляемое выражение.



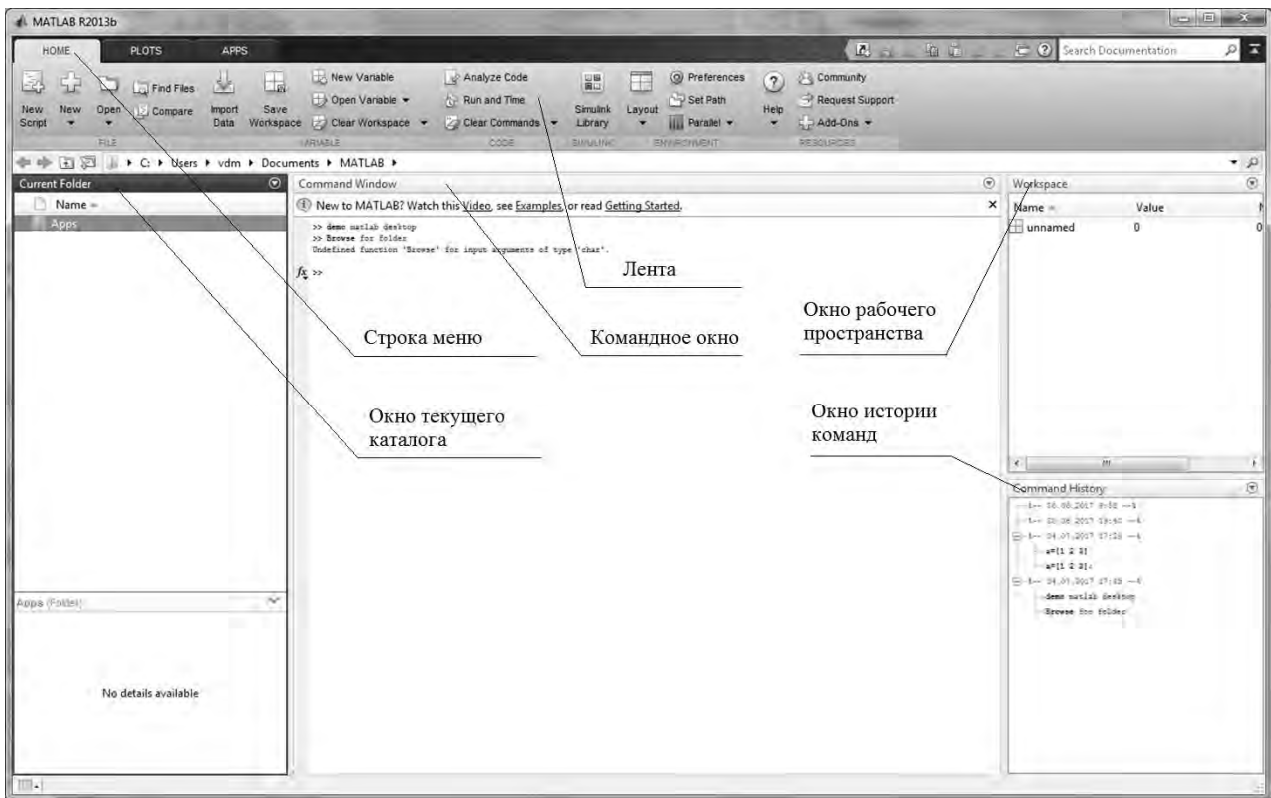


Рисунок 4.1 – Интерфейс программы MATLAB

Нажатие клавиши «*Enter*» заставляет систему вычислить выражение и показать результат. Если запись оператора не заканчивается символом «*;*», то результат выводится в командное окно, в противном случае – не выводится.

Если оператор не содержит знака присваивания «*=*», то значение результата присваивается системной переменной «*ans*».

Клавиши-стрелки курсора «*Вверх*» и «*Вниз*» используются для подстановки после маркера строки ввода «*>>*» ранее введенных строк из стека команд для их исправления, дублирования или дополнения.

Стек команд – это область оперативной памяти ПК, отведенная для хранения выполненных в данном сеансе работы команд.

Все значения переменных, вычисленные в течение текущего сеанса работы, сохраняются в специально зарезервированной области памяти компьютера, называемой рабочим пространством системы MATLAB («*Workspace*»).

Для просмотра значения любой переменной из текущего рабочего пространства системы достаточно набрать ее имя и нажать клавишу «*Enter*».

Для сохранения рабочего пространства в файле с расширением «**.mat*» нужно выполнить команду «*Save Workspace*».

Команды ***clc*** и ***clear*** отвечают за очистку командного окна и буфера команд с рабочим пространством соответственно.

При работе в среде MATLAB для упрощения записи выражений целесообразно применять стандартные функции, являющиеся частью языка, их назначение и принципы использования определены разработчиком языка. Имена этих функций недопустимо указывать в качестве имен переменных.

Для представления чисел на экране имеются разные форматы. Нужный формат может быть определен в меню («*File/Preferences*») либо при помощи команды «*Format*».

Алфавит MATLAB включает:

- прописные и строчные буквы латинского алфавита;
- арабские цифры от 0 до 9;
- специальные символы.

Переменная – именованная область памяти, в которой хранятся данные определенного типа, которые можно изменить в любой момент работы программы. Имя переменной должно начинаться с буквы, за ней могут идти буквы, цифры и символ подчеркивания. Допустимы имена любой длины, но MATLAB идентифицирует их по первым 31 символу и различает прописные и строчные буквы.

Арифметические выражения представляют собой последовательности чисел, констант, переменных и функций, объединенных знаками арифметических операций. Операции одинакового приоритета выполняются в порядке слева направо. Умножение и деление имеют более высокий приоритет перед вычитанием и сложением, но круглые скобки могут изменить порядок.

4.2 Порядок выполнения работы

- 1 Изучить структуру и интерфейс окна MATLAB.
- 2 Командами «*Command Window*», «*Current Directory*», «*Workspace*» из группы «*Environment*» отработать отображение основных окон MATLAB.
- 3 Вычислить значения выражений по заданию преподавателя, повторить расчет, изменив исходные данные в истории команд.
- 4 Сохранить результаты из рабочего пространства в файл.
- 5 Изменить формат представления результатов.

Контрольные вопросы

- 1 Как обозначена строка ввода, как повторить ранее введенную команду?
- 2 Каковы основные системные переменные, стандартные функции и специальные символы MATLAB и их назначение?
- 3 Каковы основные элементы интерфейса MATLAB?
- 4 Что такое переменная?
- 5 Как MATLAB представляет введенные значения и переменные?
- 6 Какие существуют правила именования переменных в MATLAB?
- 7 Каков состав алфавита языка MATLAB?
- 8 Для чего предназначены команды `clc` и `clear`?
- 9 Какое выражение называют арифметическим?



5 Запись выражений на языке MATLAB. Приоритет операций. Стандартные функции языка

Цель работы: ознакомиться с языком программирования системы MATLAB, изучить структуру интерфейса редактора/отладчика m-файлов, научиться записывать выражения с использованием стандартных функций MATLAB, изучить программный режим работы, научиться создавать файлы сценариев.

5.1 Общие сведения

Работать в MATLAB можно в командном и программном режимах.

В командном режиме исправить ошибки без повторного введения последовательности команд после нажатия клавиши «Enter» уже нельзя.

Работа в программном режиме осуществляется путем вызова имени программы, написанной пользователем на языке программирования MATLAB.

Достоинство программного способа в том, что он позволяет сохранять разработанные вычислительные алгоритмы и благодаря этому повторять вычисления при других исходных данных.

Записывать программы, их редактировать и отлаживать необходимо в m-файлах, которые делятся на два класса:

- файлы-сценарии, не имеющие входных параметров;
- файлы-функции, имеющие входные параметры.

Для создания файла сценария выполняется команда «New script» вкладки «File», получаем окно редактора m-файла. Редактор позволяет создать m-файл.

Редактор дает возможность устанавливать в тексте файла специальные метки, именуемые точками прерывания (*breakpoints*), для оценки промежуточных результатов. Строки программы в нем нумеруются в последовательном порядке. Редактор является многооконным. Окно каждой программы оформляется как вкладка.

Редактор/отладчик m-файлов выполняет синтаксический контроль программного кода по мере ввода текста. При этом используются следующие цветовые выделения: ключевые слова – синий цвет; операторы, константы и переменные – черный цвет; комментарии – зеленый цвет; символьные переменные – фиолетовый цвет; синтаксические ошибки – красный цвет.

m-файл по умолчанию имеет название «Untitled» (Безымянный). Чтобы дать ему имя, необходимо в меню «File» выполнить команду «Save as» и в диалоговом окне указать папку и имя этого файла.

Запуск файла сценария производится кнопкой «Выполнить» или командой, содержащей имя m-файла. Результат выполнения появится в окне команд.

Файл-сценарий имеет следующую структуру:

- % Основной комментарий
- % Дополнительный комментарий
- Тело программы



Для использования m-файла необходимо его вызвать в командном окне по имени. Для блокировки вывода на экран промежуточных результатов в теле программы все операторы завершаются символом «;».

Переменные, использованные в m-файле, для вычисления выражения должны быть определены перед его вызовом в командном окне при работе программы с различными исходными данными либо в тексте m-файла, что ведет к выполнению m-файла только с этими исходными данными.

5.2 Порядок выполнения работы

- 1 Создайте программу на языке MATLAB для вычисления выражений при заданных исходных данных по заданию преподавателя.
- 2 Сохраните результат в m-файл.
- 3 Установите точку прерывания перед выполнением расчета по формуле.

Контрольные вопросы

- 1 Каковы преимущества программного режима работы в MATLAB?
- 2 Как создать, сохранить, открыть, выполнить m-файл?
- 3 Как использовать m-файл для расчета с разными исходными данными?
- 4 Для чего используют комментарии? Как записать комментарий?
- 5 Какие переменные, выражения и стандартные функции MATLAB использованы в программе?
- 6 Как идентифицируются строки программы в окне редактора?

6 Формирование векторов и матриц с требуемыми свойствами. Операции с элементами и блоками элементов матриц

Цель работы: получить навыки работы с массивами, векторами и матрицами в программном режиме.

6.1 Общие сведения

Все данные MATLAB представляет в виде массивов. Массив – упорядоченная, пронумерованная совокупность однородных данных.

MATLAB не требует от пользователя предварительного задания размерности массива. Пользователь может вводить элементы массива последовательно, при этом MATLAB будет динамически перестраивать его структуру.

Массивы в MATLAB состоят из элементов типа «double».

Под вектором в MATLAB понимается одномерный массив чисел, а под матрицей – двумерный массив.

Для создания одномерных массивов используют:

– операцию конкатенации: $z = [1\ 2\ 3]$ – строка, $b = [1;2;3]$ – столбец;



– операцию индексации для экономии ресурсов присваивание значений элементам массива начинают с последних: $z(3) = 3$; $z(2) = 2$; $z(1) = 1$;

– вызов специальных функций:

zeros(m,n) – создает матрицу размером $M \times N$ с нулевыми элементами;

ones(m,n) – создает матрицу размером $M \times N$ с единичными элементами;

eye(m,n) – создает единичную матрицу размером $M \times N$;

rand(m,n) – создает матрицу $M \times N$ из случайных чисел, равномерно распределенных в диапазоне от 0 до 1;

– специальную операцию, обозначаемую двоеточием: $z = 1:1:3$.

При необходимости изменить элемент сформированного одномерного массива используют операцию индексации и операцию присваивания: $z(3) = 78$.

Запись несуществующего элемента массива означает добавление нового элемента к уже существующему массиву: $z(4) = 7$.

Удлинение массива z можно выполнить и с помощью операции конкатенации: $z = [z \ 7]$. Можно подвергнуть конкатенации и несколько массивов: $y = [z \ z \ 9 \ z]$.

Для создания двумерного массива (матрицы) в MATLAB используют операции конкатенации, индексации и вызов специальных функций.

Значения элементов матрицы вводятся в квадратных скобках по строкам, при этом элементы строки матрицы разделяются пробелом или запятой, а строки отделяются одна от другой точкой с запятой: $X = [1 \ 2; \ 3 \ 4; \ 5 \ 6]$.

Двумерные массивы можно создать с помощью операции индексации, прописывая по отдельности элементы необходимыми числовыми значениями: $X(3, 3) = 1$; $X(3,2) = 2$ и т. д.

Выделение блоков матриц осуществляется индексацией при помощи двоеточия: $X1 = X(2:3, 2:3)$.

Для чтения из массивов используют операцию индексации справа от знака присваивания: $y = z(2)$.

Для выделения из матрицы столбца или строки следует в качестве одного из индексов использовать номер столбца или строки матрицы, а другой индекс заменить двоеточием: $x = X(2, :)$.

При выделении блока до конца матрицы можно не указывать его размеры, а использовать слово `end`: $x = X(2, 2:end)$.

Для удаления строк или столбцов присваивают им пустой массив: $[]$.

6.2 Порядок выполнения работы

1 Создать по заданию преподавателя массивы в командном режиме, произвести их обработку с помощью программы–скрипта на языке MATLAB.

2 Сохранить файл на диске. Запустить его на выполнение. Просмотреть результаты в Workspace.



Контрольные вопросы

- 1 Какие способы создания матриц Вам известны?
- 2 Как заполнить массив различными способами?
- 3 Как создать дополнительные элементы вектора или матрицы?
- 4 Как удаляются строки и столбцы матриц?
- 5 Как создать пустой массив, выделить из матрицы строку или столбец?
- 6 Какие стандартные функции по работе с массивами Вам известны?
- 7 Назовите основные поэлементные действия над матрицами.

7 Создание простейших программ. Реализация линейного алгоритма на языке MATLAB. Организация диалога с пользователем. Отладка программы. m-файлы

Цель работы: получить навыки работы в режиме диалога с пользователем и решения задач на основе линейного алгоритма в программном режиме работы.

7.1 Общие сведения

При составлении программ с линейным алгоритмом одним из основных операторов является оператор присваивания:

Имя_переменной = Выражение,

где Имя_переменной – имя простой переменной, вектора, матрицы, функции;
Выражение – арифметическое, логическое или строковое выражение.

Тип имени переменной определяется системой автоматически по типу выражения.

Работа оператора: вычисляется выражение справа от «=» и результат присваивается переменной слева от «=».

В MATLAB в качестве оператора ввода используется функция `input`, которую принято называть оператором ввода:

Имя_переменной = **input**(Символьная константа),

где Символьная константа – набор символов, заключенный в апострофы.

Пока пользователь не ввел значение в ответ на запрос программы, блокируется дальнейшее выполнение программы.

Если необходимо вывести данные на экран дисплея, применяется функция `disp`, которую принято называть оператором вывода:

disp(Выражение),



где Выражение – это арифметическое, логическое или символьное выражение.

Символы и текстовые строки в MATLAB всегда вводятся при помощи простых кавычек: $s = \text{'Привет'}$.

Над строками и символьными данными выполняется операция конкатенации: $h = [s + \text{'от MATLAB'}]$, s – символьная переменная, 'от MATLAB' – строковое значение.

Если в строковом выражении, создаваемом конкатенацией, необходимо использовать число, его преобразуют в строку, используя функции преобразования.

7.2 Порядок выполнения работы

1 Создать m-файл для решения задач, реализующих линейный алгоритм, по вариантам, используя режим диалога с пользователем.

2 Организовать вызов m-файла для расчета с другими исходными данными.

Контрольные вопросы

- 1 Какой оператор (операция) реализует линейный алгоритм?
- 2 Что происходит при операции присвоения?
- 3 Какой тип данных используют в аргументе функций input и disp?
- 4 Как объединить строковые значения в одно выражение?
- 5 Для чего используют функции преобразования?

8 Программирование разветвляющегося алгоритма на языке MATLAB

Цель работы: изучить операции отношения, логические операции и условные операторы, приобрести навыки их использования при реализации разветвляющихся алгоритмов.

8.1 Общие сведения

Разветвляющийся алгоритм – алгоритм, содержащий хотя бы одно условие, обеспечивающий ветвление основного потока программы.

Для организации ветвлений в MATLAB используются условные операторы if и оператор переключения switch.

В качестве условий в условных операторах используют логические выражения, которые строятся из арифметических выражений, операндов (значений, констант, переменных, функций) логических операций и отношений, результатом вычисления логических выражений являются логические значения «ложь» (false) и «истина» (true).



Полная форма 1:

```

if Условие 1
    Инструкции_1
elseif Условие 2
    Инструкции_2
else
    Инструкции_3
End

```

Полная форма 2:

```

if Условие
    Инструкции_1
else
    Инструкции_2
end

```

Сокращенная форма:

```

if Условие
    Инструкции_1
end

```

Инструкции_1, Инструкции_2, Инструкции_3 – операторы и функции для обработки данных

Работа оператора if:

- если условие возвращает логическое значение «истина», выполняются инструкции_1 и управление передается оператору, следующему за оператором if;
- если условие возвращает логическое значение «ложь», проверяется условие 2 (если есть), выполняются инструкции_2 и управление передается оператору, следующему за оператором if и т. д.;
- в сокращенной форме, если условие возвращает значение «ложь», то инструкции не выполняются, а управление передается оператору, следующему за оператором if.

Для осуществления множественного выбора (или ветвления) используется конструкция с переключателем типа switch.

Полная форма:

```

switch Выражение селектор
    case список констант 1, Список инструкций_1
    case список констант 2, Список инструкций_2
    ...
    case список констант n, Список инструкций_n
otherwise Список инструкций_N+1
end

```

Сокращенная форма:

```

switch Выражение селектор
    case список констант 1, Список инструкций_1
    case список констант 2, Список инструкций_2

```



...
case список констант n, Список инструкций_n
end

Работа оператора switch:

- вычисляется выражение селектор;
- последовательно проверяется совпадение значения селектора со значениями списка констант, если совпадение с каким-либо списком констант есть, то выполняются инструкции, соответствующие этому списку констант, при этом следующие далее списки констант не проверяются и управление передается оператору следующему за switch, если совпадений ни с одним списком констант нет, то выполняются инструкции, следующие за словом otherwise, и управление передается оператору, следующему за switch;
- в сокращенной форме, если совпадений ни с одним списком констант нет, то управление передается оператору, следующему за switch.

8.2 Порядок выполнения работы

1 Создать m-файл для решения задач с использованием условного оператора по вариантам, ввод исходных данных выполнить в режиме диалога с пользователем.

2 Создать m-файл для решения задач с использованием оператора выбора по вариантам, ввод и вывод данных выполнить в режиме диалога с пользователем.

Контрольные вопросы

- 1 Для чего используются условные операторы?
- 2 В чем отличие операторов if и switch?
- 3 Какой тип выражений используют в качестве условий?
- 4 Каковы основные логические операции и отношения?
- 5 Какие значения может принимать результат логического выражения?
- 6 Каковы синтаксис и работа использованных операторов?



9 Программирование циклических алгоритмов на языке MATLAB

Цель работы: приобрести навыки использования операторов повтора при организации вычислений в циклических алгоритмах.

9.1 Общие сведения

Циклический алгоритм – алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Многократно повторяющиеся участки называются циклами или телом цикла.

В MATLAB используют:

- оператор цикла с параметрами **for**;
- оператор цикла с условием **while**.

Оператор цикла **for** предназначен для программирования циклических алгоритмов с заданным числом повторений.

Синтаксис при явно заданном шаге изменения переменной цикла:

```
for Имя ПЦ= НПЦ : ШПЦ : КПЦ
    Инструкции
End
```

Синтаксис при шаге изменения переменной цикла, равном 1:

```
for Имя ПЦ = НПЦ : КПЦ
    Инструкции
End
```

где Имя ПЦ – идентификатор переменной цикла;

НПЦ – начальное значение переменной цикла;

КПЦ – конечное значение переменной цикла;

ШПЦ – шаг изменения переменной цикла при повторении цикла;

Инструкции – оператор или группа операторов, составляющих тело цикла.

Работа оператора **for**:

- 1) проверяется условие Начальное значение ПЦ \leq Конечное значение ПЦ;
- 2) если условие не выполняется, то оператор цикла прекращает свою работу и выполняется следующий за ним оператор программы;
- 3) если условие выполняется, то переменной цикла присваивается ее начальное значение (или принимается следующее с заданным шагом);
- 4) выполняется тело цикла (инструкции);
- 5) переменная цикла увеличивается на шаг и выполняется шаг 4;



б) шаг 5 повторяется, пока значение ПЦ \leq КПЦ, в противном случае оператор цикла прекращает свою работу и выполняется следующий за ним оператор программы.

Корректное задание параметров цикла определяется выражением $\text{НПЦ} \leq \text{КПЦ}$ при $\text{ШПЦ} > 0$, $\text{НПЦ} \geq \text{КПЦ}$ при $\text{ШПЦ} < 0$.

Количество повторений цикла определяется из зависимости $(\text{КПЦ} - \text{НПЦ}) / \text{ШПЦ} + 1$.

Оператор `while` предназначен для программирования любых циклов, где проверка условия повторения цикла выполняется перед выполнением рабочей части цикла и действий, число которых заранее не известно, но известно условие продолжения цикла:

while ЛВ

 Инструкции

end

где ЛВ – логическое выражение.

Работа оператора `while`:

- 1) проверяется (вычисляется) значение логического выражения;
- 2) пока оно истинно, выполняется тело цикла (инструкции);
- 3) повторяются шаги 1 и 2;
- 4) если логическое выражение принимает ложное значение, то работа оператора прекращается и выполняется следующий за оператором `while` оператор программы.

Особенности использования оператора `while`:

- если переменная цикла явно выражена, нужно присвоить ей начальное значение до оператора цикла;
- необходимо предусмотреть изменение переменной цикла в теле цикла;
- переменная цикла должна использоваться в логическом выражении условия повторения цикла;
- если логическое выражение в заголовке цикла `while` заведомо ложно, то цикл может не выполниться ни разу.

В управляющих структурах циклов используются операторы, влияющие на их выполнение. Для этого в теле цикла предусматривают дополнительное условие, при котором выполняют оператор **break** для досрочного прерывания выполнения цикла или оператор **continue** для передачи управления в следующую итерацию цикла, пропуская операторы, которые записаны за ним, причем во вложенном цикле он передает управление на следующую итерацию основного цикла.

В теле любого оператора цикла могут находиться другие операторы цикла. При этом цикл, содержащий в себе другой, называют внешним, а цикл, находящийся в теле первого, – внутренним (вложенным). Таких вложенных циклов во внешнем цикле может быть несколько. Все операторы внутреннего цикла должны полностью располагаться в теле внешнего цикла.



При вложении циклов `for` внутренний цикл выполняется полностью от начального до конечного значения его параметра цикла при неизменном значении параметра внешнего цикла. Затем значение параметра внешнего цикла изменяется на единицу, и опять от начала и до конца выполняется вложенный цикл. И так до тех пор, пока значение параметра внешнего цикла не станет больше конечного значения параметра внешнего цикла.

Аналогично осуществляется работа циклов `while` за исключением того, что проверка условия выполнения внешнего цикла производится только после всех итераций внутреннего цикла.

Возможна комбинация разных операторов цикла при использовании вложенности, условия их работы определяются комбинированием вышеуказанных правил.

9.2 Порядок выполнения работы

1 Создать программу для вычисления значения суммы ряда по вариантам с использованием операторов `for` и `while` и сравнить результаты вычислений.

2 Изменить созданную программу для досрочного выхода из цикла, когда номер суммируемого элемента ряда станет равен номеру варианта.

3 Создать программу для решения задачи по вариантам с использованием вложенных циклов.

Контрольные вопросы

- 1 Для чего используются операторы повтора?
- 2 Каковы синтаксис и работа операторов `for` и `while`?
- 3 Назовите основные требования к организации циклов `for` и `while`.
- 4 Какие способы управления работой цикла Вы знаете?
- 5 Что такое вложенные циклы, правила работы вложенных циклов?

10 Построение графиков в декартовых и полярных координатах

Цель работы: приобрести практические навыки использования средств MATLAB для представления данных в графической форме.

10.1 Общие сведения

Основной функцией, обеспечивающей построение графиков на экране дисплея, является функция `plot(x1, y1, s1, x2, y2, s2,...)`, где `x1, y1, x2, y2` – векторы значений аргументов и функций, `s1, s2, ...` – символьные переменные, определяющие тип линии, тип точки, цвет линии.



Наиболее простыми частными случаями являются команды:

– **plot**(X, Y) – для построения графика функций одной переменной;

– **plot** (X, Y, S) – тип линии графика можно задавать с помощью строковой константы S;

– для построения нескольких графиков в одних и тех же координатных осях, функция **plot** записывается следующим образом: **plot**(x1,y1,x2,y2).

Если переменные s не указаны, то тип линии – сплошной отрезок прямой, тип точки – пиксел.

Цвет устанавливается в очередности построения графиков: синий, зеленый, красный, голубой, фиолетовый, желтый, черный и белый.

При построении графиков толщина линии задается параметром ('LineWidth', толщина линии в пикселях), который указывается после основных параметров.

Графики в MATLAB выводятся в отдельное графическое окно, которое называют фигурой. Для создания нового графического окна используют функцию **figure**.

Для того чтобы на экране всегда отображалось два окна с нужными графиками, при вызове функции **figure** необходимо в качестве аргумента указывать номер графического окна, которое следует создать или сделать активным, если оно уже создано (**figure**(1), **figure**(2) и т. д.

Отображение двух графиков в одном графическом окне обеспечивает функция: **subplot**(число строк, число столбцов, номер подокна).

Для построения графика в трехмерных координатных осях, используется функция **plot3**(X,Y,Z), где в качестве первых двух аргументов принимаются матрицы с координатами точек по осям O_x и O_y соответственно, а в качестве третьего передается матрица значений точек по оси O_z.

Для получения матрицы координат x и y получающейся прямоугольной сетки применяют функцию **meshgrid**: [X , Y] = **meshgrid**(x , y).

Для отображения графика в виде сетки используют функцию **mesh**(X,Y,Z), при этом получается график, образованный интерполяцией точек массивов X, Y и Z линиями по осям O_x и O_y.

Цветом указывается уровень точки по оси O_z от самого малого значения (синего) до самого большого (красного) и производится удаление «невидимых» линий.

Для управления прозрачностью графика используют команды:

– **hidden off** – прозрачность включена;

– **hidden on** – прозрачность выключена.

В системе MATLAB предусмотрена функция визуализации непрерывной поверхности в трехмерных осях **surf**(X,Y,Z), которая строит каркасную поверхность графика функции и заливает каждую клетку поверхности цветом, зависящим от значения функции в точках, соответствующих углам клетки.

Команда **shading flat** позволяет убрать каркасные линии.

Для получения поверхности, плавно залитой цветом, зависящим от значения функции, предназначена команда **shading interp**.

Вернуться к начальному виду поверхности можно с помощью команды **shading faceted**.

Для изменения цветовой карты используется функция **colormap**(имя карты).

Для управления созданным окном графика применяется команда **set**(object name, 'Property name1', 'Property value1', ...), где object name – имя управляемого объекта, 'Property name1' – строковая константа, содержащая название изменяемого свойства, 'Property value1' – строковая константа, содержащая значение изменяемого свойства.

Для установки титульной надписи предназначена команда **title**(string). Для подписей осей x, y и z применяются команды **xlabel**(string), **ylabel**(string), **zlabel**(string), содержание определяется строковой переменной string.

Пояснение в виде отрезков линий со справочными надписями, размещаемое внутри графика или около него, называется легендой и создается командой:

– **legend**(string1,string2,string3,...) – добавляет к текущему графику легенду в виде строк, указанных в списке параметров;

– **legend**(string1,string2,string3,...,Pos) – помещает легенду в точно определенное место, специфицированное параметром Pos: Pos = 0 – автоматически; Pos = 1 – верхний правый угол; Pos = 2 – верхний левый угол; Pos = 3 – нижний левый угол; Pos = 4 – нижний правый угол; Pos = -1 – справа от графика.

Изменить масштабирование графиков позволяет команда:

– **axis**([xmin xmax ymin ymax]) – установка диапазонов координат по осям x и y для текущего двумерного графика;

– **axis auto** – установка параметров осей по умолчанию.

Для улучшения чтения графиков среда MATLAB позволяет создавать масштабную сетку:

– **grid on** – добавляет сетку к текущему графику;

– **grid off** – отключает сетку.

Для построения наложенных друг на друга графиков в одном и том же окне служит команда управления графическими построениями:

– **hold on** – обеспечивает продолжение вывода графиков в текущее окно, что позволяет добавлять последующие графики к уже существующим;

– **hold off** – отменяет режим продолжения графических построений.

Для вывода текста используется команда **text**(x,y, 'текст'), которая создает текстовую надпись в координатах (x,y).

Изменить свойства линии, вывести на график обозначения осей, легенду, включить сетку можно с помощью свойств графика в вызываемой в меню графического окна команде «*Edit Plot*».

Для построения графиков функций в полярной системе координат используется функция **polar**(Theta,R,S), которая строит график в полярной системе координат, представляющий собой положение конца радиус-вектора с длиной R и углом Theta, строковая константа S задает стиль построения.



10.2 Порядок выполнения работы

- 1 Вычислить вектор у значений функции $y(x)$.
- 2 Построить график функций в декартовых координатах с использованием различных типов и толщины линий, цвета и маркера.
- 3 Вычислить функцию с пределами изменения x от -2π до 2π и y от -2π до 2π и построить трехмерный график.
- 4 Вывести в одно графическое окно функцию в виде трехмерных графиков разного типа (используя три разные команды вывода трехмерных графиков).

Контрольные вопросы

- 1 Какие команды используются для построения графиков?
- 2 Каков порядок построения графиков?
- 3 Как создать новое окно для построения графика?
- 4 Как создать дополнительное подокно для вывода графика?
- 5 Как создать в одном окне наложенные друг на друга графики?
- 6 Как редактируется внешний вид графиков?
- 7 Как использовать греческие буквы и специальные символы?

11 Организация вычислений с использованием функций пользователя

Цель работы: изучить возможности MATLAB для организации вычислений повторяющихся алгоритмов, приобрести практические навыки создания и использования функций пользователя.

11.1 Общие сведения

В программировании встречаются случаи, когда приходится выполнять одни и те же вычисления, но при различных исходных данных. Для исключения одинаковых фрагментов программы повторяющиеся вычисления выделяют в самостоятельную часть – подпрограмму.

Подпрограммой называется автономная часть программы, реализующая определенный алгоритм и допускающая обращение к ней из различных частей общей программы неограниченное число раз по мере необходимости.

Преимущества использования подпрограмм:

- отсутствие дублирования, что сокращает трудоемкость создания программы;
- более удобный процесс отладки и внесения изменений;
- повышение надежности программы.

В языке MATLAB все подпрограммы можно разделить на встроенные и пользовательские.



Встроенные (стандартные) подпрограммы являются частью языка и могут вызываться по имени без предварительного описания.

Пользовательские (нестандартные) подпрограммы пользователя пишутся самим программистом в соответствии с синтаксисом языка.

Передача данных в подпрограмму и возврат результата выполнения осуществляются с помощью параметров. Различают формальные и фактические параметры. Параметры, которые указываются при объявлении (описании) подпрограммы, называются формальными. Параметры, которые указываются при вызове подпрограммы, называются фактическими.

Количество фактических параметров должно соответствовать числу формальных параметров.

В языке MATLAB выделяют два вида функций пользователя – встраиваемые и внешние.

Для создания функции вручную используют окно редактора m-файлов либо команду «*New function*» вкладки «*File*».

Структура описания внешней функции:

```
function[выходные параметры]=Имя функции(входные параметры)
% Тело функции – любые выражения и операторы MATLAB
end
```

Слово **end** допускается не указывать за исключением случая, когда используется вложенность функций.

Вызов функции в основной программе осуществляется по ее имени с указанием фактических входных параметров.

После завершения работы функции фактические значения присваиваются выходным параметрам.

В простейшем случае у функции есть один входной и один выходной параметры. У функции может быть несколько входных и несколько выходных параметров. Список как входных, так и выходных параметров может быть пустым, и получение исходных данных для проведения вычислений и вывод полученных результатов могут осуществляться исключительно с помощью операций ввода-вывода.

Встраиваемая функция создается с помощью процедуры **inline** или анонимной командой **@** непосредственно в командном окне и часто называется **inline**-функцией:

Имя функции=**inline** (выражение, список аргументов);

Имя функции=**@**(аргументы)выражение.

Определяющее функцию выражение и имена аргументов передаются в **inline**-функцию только в виде строковых выражений. У **inline**-функций может быть только один выходной параметр, совпадающий с именем самой функции.

MATLAB допускает два способа описания нескольких функций внутри одного файла:

- основная функция и ее подфункции;
- основная функция и вложенные функции.



m-файл с программой-функцией может содержать описание не одной, а нескольких функций. Имя главной функции должно совпадать с именем файла. При вызове функции можно будет обратиться только к той функции, имя которой совпадает с именем m-файла. Все остальные функции называются внутренними (подфункциями) и вызываются только из главной функции.

Чтобы создать вложенную функцию, нужно разместить ее тело внутри тела другой функции, каждая из функций должна заканчиваться ключевым словом `end`. Количество уровней вложенности не ограничено.

Область видимости – это та часть программного кода, в которой может быть использована данная переменная. Видимость зависит от того, на каком уровне объявлена переменная: на внутреннем (локально), т. е. внутри некоторого блока, или на внешнем (глобально), т. е. вне всех блоков.

Для того чтобы переменная была видна за пределами функции (являлась глобальной), в которой она объявлена, используют обращение с помощью ключевого слова **global**, за которым следует имя глобальной переменной.

11.2 Порядок выполнения работы

- 1 Создать функцию для решения задачи по вариантам.
- 2 Составить программу для ввода исходных данных в режиме диалога, организовать их обработку, вызвав созданную на шаге 1 функцию.
- 3 Решить задачи по вариантам с использованием вложенных функций пользователя; использовать внешнюю и встроенную (`inline`) функции.

Контрольные вопросы

- 1 Что называют подпрограммой?
- 2 Какие виды функций используют в MATLAB?
- 3 Каково определение формальных и фактических параметров?
- 4 Какова структура описания разных типов функций в MATLAB?
- 5 Как вызывают разные типы функции в основной программе?
- 6 Что называют локальными и глобальными переменными?

12 Создание презентации по изученному материалу

Цель работы: изучить аппаратные и программные средства создания презентаций, получить практические навыки создания презентации.

12.1 Общие сведения

Презентация – это целенаправленный информационный процесс, решающий свои задачи, коммуникационный процесс со всеми его основными элементами.

Слайд – автономная информационная структура, содержащая различные



объекты, которые представляются на общем экране в виде единой композиции.

PowerPoint предназначен для создания экранных презентаций и позволяет создавать классические слайды, строить электронные презентации с мультимедийными эффектами. Структура и управление интерфейсом, отвечающие за основные возможности управления документом, аналогичны другим продуктам Microsoft Office.

Создание презентации состоит из следующих этапов: выбор общего оформления; добавление новых слайдов и их содержимого; выбор разметки слайдов; изменение при необходимости оформления слайдов; замена цветовой схемы; применение различных шаблонов оформления; создание эффектов анимации при демонстрации слайдов.

Способы создания новой презентации:

- создание презентации на основе других готовых презентаций;
- создание презентации на основе готовых шаблонов слайдов;
- создание презентации на основе пустых макетов слайдов.

12.2 Порядок выполнения работы

1 Создайте слайд «Главная» – «Слайды» – «Создать слайд».

2 Выберите дизайн для презентации: «Дизайн» – «Темы».

3 Оформите титульный лист презентации в соответствии с темой.

4 Создайте нижний колонтитул с Вашими ФИО, установите дату и время: «Вставка» – «Текст» – «Колонтитулы».

5 Наберите текст слайда с перечнем целей и задач и оформите его (размер шрифта Times New Roman – 22 pt, выравнивание по ширине, абзац – 1,25 см).

6 Измените внешний вид слайда (вкладка «Дизайн») – выберите из представленных вариантов.

7 Далее создайте слайды с соответствующими заголовками, содержащие:

- маркированный список выполненных в семестре работ. Добавьте изображения программных продуктов, использованных при выполнении работ;
- таблицу, содержащую наименование работы и оценку за ее выполнение;
- схему по заданию преподавателя.

8 На вкладке «Анимация» в разделе «Переход к этому слайду» выберите для каждого слайда тип анимации (звук перехода и т. д.).

9 На вкладке «Показ слайдов» в разделе «Настройка» настройте время показа для каждого слайда.

10 Просмотрите презентацию на полном экране (клавиша «F5» либо «Shift-F5» для показа с текущего слайда).

11 Произведите сортировку слайдов, используя окно сортировщика слайдов.



Контрольные вопросы

- 1 Каковы основные этапы создания презентаций?
- 2 Каковы основные возможности PowerPoint по созданию и редактированию презентаций?

13 Оформление технического документа с использованием текстового редактора. Включение специальных объектов. Составление оглавления документа

Цель работы: изучить структуру и основные элементы интерфейса, получить практические навыки настройки и работы с текстовым процессором Word.

13.1 Общие сведения

Microsoft Word – текстовый процессор, предназначенный для создания профессионально оформленных документов и организации удобной среды для совместной работы. Окно Word представлено на рисунке 13.1.

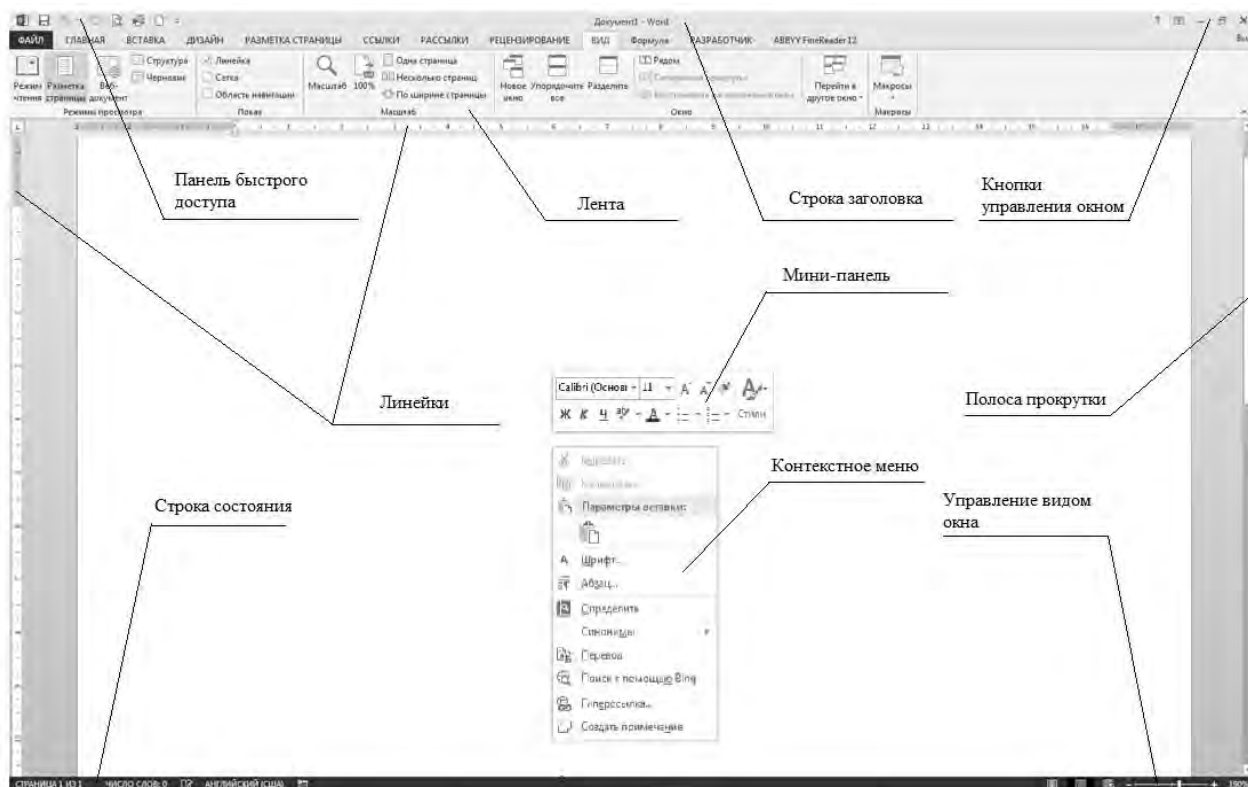


Рисунок 13.1 – Структура окна

К элементам интерфейса относятся также мини-панели инструментов, содержащие наиболее часто используемые элементы для оформления объектов

(появляются автоматически при выделении фрагмента документа).

Лента является главным элементом пользовательского интерфейса и состоит из вкладок. Наборы команд каждой вкладки объединены в группы. При выполнении некоторых задач автоматически появляются контекстные вкладки.

Настройка элементов интерфейса Word осуществляется с использованием команды «*Параметры*» на вкладке «*Файл*».

Создание и сохранение документа выполняется с расширением «*.docx» и использованием соответствующих команд меню «*Файл*».

Ввод текста начинается с позиции курсора. Верстка текста осуществляется автоматически. В конце абзаца нажимается клавиша «*Enter*».

Для добавления в документ новой страницы в конце текущей необходимо воспользоваться клавишей «*Enter*» либо командами «*Пустая страница*», «*Разрыв страницы*» вкладки «*Вставка*».

Для редактирования текста используются кнопки команд на ленте вкладки «*Главная*», а также диалоговые окна «*Разметка Страницы*», «*Шрифт*» и «*Абзац*», вызываемые либо из «*Контекстного меню*», либо щелчком по кнопке соответствующей группы в ленте вкладки «*Главная*».

Для работы с фрагментом текста необходимо его выделить:

- с помощью мыши при нажатой левой клавише;
- с помощью клавиши «*Shift*» и клавиш управления курсором;

Одно слово можно выделить двойным щелчком левой клавиши мыши, а абзац – тройным щелчком.

Для создания таблицы используется команда «*Таблица*» на ленте вкладки «*Вставка*». Для добавления и удаления строк и столбцов, изменения оформления таблиц применяют команды контекстного меню и контекстных вкладок «*Конструктор и макет*».

Изменение высоты строк и ширины столбцов осуществляют с помощью маркера границ в виде двойной стрелки, который виден при установке указателя мыши на границу.

Выделение элементов таблиц:

- с помощью мыши при нажатой левой клавише;
- с помощью клавиши «*Shift*» и клавиш управления курсором;
- один столбец или строку подвести курсор мыши к началу строки или столбца и, когда изображение курсора изменится, нажать левую клавишу мыши;
- всю таблицу подвести курсор к левому верхнему углу таблицы и щелкнуть по значку.

Расчеты в таблицах Word возможны с использованием диалогового окна «*Формула*», вызываемого на вкладке «*Макет*» при работе с таблицей.

Редактор математических выражений Microsoft Equation ориентирован на создание формул в тексте документа для запуска:

- в диалоговом окне «*Вставка объекта*», вызываемом командой «*Объект*» на ленте вкладки «*Вставка*», выбирается тип объекта Microsoft Equation;
- используются команды вкладки «*Формула*».



Для нумерации страниц необходимо вызвать команду «*Номер страницы*» в группе «*Колонтитулы*» на вкладке «*Вставка*».

Для создания оглавления заголовки разделов и подразделов набирают стилем «*Заголовок*» из группы «*Стили*» на вкладке «*Главная*», далее выбирают команду «*Оглавление*» в группе «*Оглавление*» на вкладке «*Ссылки*».

В режиме редактирования следует активировать непечатаемые знаки командой «*Отобразить все знаки*» в группе «*Абзац*» вкладки «*Главная*».

13.2 Порядок выполнения работы

1 Ознакомиться с интерфейсом Word, создать собственную вкладку «*Мои команды*» на ленте, в ней – группу «*Формулы*» и поместить команду вызова редактора формул Microsoft Equation, настроить панель быстрого доступа, на которую поместить кнопки, создания документа, просмотр и печать, кнопку вызова редактора формул.

2 Создать отчет, содержащий титульный лист по образцу, установить параметры страниц документа, стиль – обычный, шрифт – Times New Roman, размер шрифта – 14 pt, для подписи таблиц и рисунков размер шрифта – 12 pt, заголовки – жирный 16 pt. В поле основной надписи «*Лист*» вставить нумерацию страниц. На листе содержания, используя средства Word, составить оглавление по заголовкам пунктов отчета.

3 Вставить в текст документа рисунки с копиями экрана, содержащими созданные в пункте 1 задания элементы интерфейса, и подписать их.

4 Создать таблицу по образцу (рисунок 13.1) с параметрами шрифт – 12 pt, высота строк – 0,6 см, автоподбор ширины столбцов – по содержимому, выравнивание таблицы по центру, выравнивание содержимого – по центру, обрамление ячеек – толщиной 2,25 pt. Заполнить таблицу, задавшись наименованием товара и его себестоимостью. Налог определить равным 13,5 % от себестоимости, цену 122 % от суммы себестоимости и налога. Определить прибыль от реализации каждого товара. Расчеты выполнить с использованием команды «*Формула*» вкладки «*Макет*» и адресации ячеек.

№п.п	Наименование товара	Себестоимость, тыс.руб.	Налог, тыс.руб.	Цена, тыс.руб.	Прибыль, тыс.руб.
1	Авторучка				
...	...				
5	Фломастеры				
	ИТОГО:				

Рисунок 13.1 – Образец оформления таблицы

5 Выполнить набор двух формул по заданию преподавателя: одной с использованием вставки объекта Microsoft Equation, второй – с использованием команды вкладки «*Формула*».

Контрольные вопросы

- 1 Назовите и покажите основные части окна Word.
- 2 Как настраиваются элементы интерфейса?
- 3 Как производится выделение, вставка/удаление фрагментов документа?
- 4 Как форматируются шрифты и абзацы?
- 5 Как создается и редактируется таблица?
- 6 Как задаются параметры страниц их нумерация, создается оглавление?
- 7 Как вставить формулу в текст документа?

14 Решение инженерных задач с использованием электронных таблиц. Визуализация результатов. Оформление таблицы

Цель работы: изучить структуру интерфейса и настройки MS Excel, получить практические навыки обработки данных в MS Excel.

14.1 Общие сведения

Табличный процессор MS Excel – это компьютерная программа для хранения и обработки информации, представленной в табличной форме. С помощью MS Excel можно не только создавать таблицы, но и автоматизировать обработку данных, производить их сортировку, фильтрацию и анализ.

Окно Excel имеет структуру, присущую большинству программ пакета MS Office, но структура интерфейса Excel содержит и элементы, присущие именно этому приложению (рисунок 14.1).

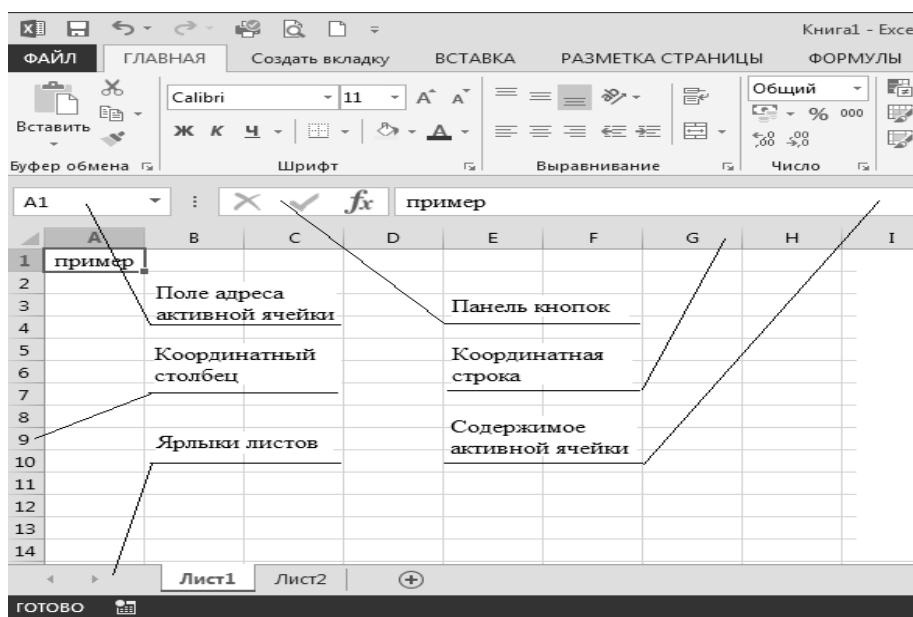


Рисунок 14.1 – Отличительные элементы Excel



Строка формул состоит из поля адреса активной ячейки, панели кнопок и содержимого активной ячейки.

Рабочая область листа состоит из ячеек, стоящих на пересечении столбцов и строк. Столбцы именуются латинскими буквами, а строки – цифрами.

Координатная строка содержит имена столбцов.

Координатный столбец содержит номера строк.

В нижней части расположены ярлыки листов с кнопками прокрутки для перехода по рабочим листам.

Настройки приложения Excel и элементов его интерфейса аналогичны описанным в работе №13.

По умолчанию файлы MS Excel имеют расширение «*.XLSX». Создаваемые в Excel файлы называются рабочими книгами. Рабочая книга содержит один или несколько листов по умолчанию имеет имя Книга1, 2,

Рабочий лист служит для ввода и обработки данных. Имена листов находятся на ярлыках, расположенных в нижней части окна книги. Название текущего листа всегда выделено. Листы можно переименовывать, вставлять, удалять, перемещать или копировать в пределах одной книги или из одной книги в другую.

На пересечении строки и столбца находится ячейка. Каждая ячейка имеет уникальный адрес, в котором указывается имя столбца и номер строки, на пересечении которых она расположена. Ячейка, где находится курсор, называется текущей.

Диапазон (блок) ячеек – это прямоугольник, в котором указываются адреса ячеек левого верхнего и нижнего правого углов, разделенных двоеточием, например «A1:D4».

Чтобы изменить формат ячеек, следует вызвать соответствующее диалоговое окно из контекстного меню командой «*Формат ячеек*». Управлять шириной и высотой ячеек можно, перемещая границу ячейки с помощью мыши при нажатой левой клавише.

Для ввода данных необходимо переместиться в нужную ячейку, набрать данные и нажать клавишу «*Enter*» либо перейти в другую ячейку. Число выравнивается по правому краю ячейки, текст – по левому краю.

Для заполнения таблицы значениями, изменяющимися как арифметическая прогрессия:

– в две смежные ячейки ввести первый и второй член арифметической прогрессии и выделить эти ячейки. Затем установить указатель мыши на маркер заполнения выделенного диапазона и протянуть его в нужном направлении, пока не получится числовой ряд нужной длины;

– в ячейку ввести первый член прогрессии. Выбрать команду «*Заполнить*» в группе «*Редактирование*» на вкладке «*Главная*» и в открывшемся диалоговом окне «*Прогрессия*». В группе «*Расположение*» установить переключатель расположения данных прогрессии, в группе «*Тип*» выбрать тип прогрессии, в поле «*Шаг*» ввести значение шага, в поле «*Предельное значение*» ввести предельное значение прогрессии.



Для редактирования содержимого ячейки нужно выделить ячейку и нажать функциональную клавишу «F2» или дважды щелкнуть по ячейке.

Можно удалять ячейки или очищать их содержимое. При удалении ячейки Microsoft Excel так перемещает соседние ячейки, чтобы они заполнили освободившееся место. При очистке ячейки удаляется ее содержимое (формулы и данные), форматы и примечания, а сама ячейка остается на листе.

Для удаления или очистки выделенных ячеек необходимо воспользоваться командой «Удалить», или командой «Очисть содержимое» в контекстном меню, или клавишей «Del».

Удаление и добавление строк и столбцов производят командами из контекстного меню, вызванного на заголовке строки или столбца.

Чтобы быстро переместить или скопировать данные, следует выделить нужные ячейки и:

- перетащить границу выделенного диапазона с помощью мыши (используя маркер заполнения знак «+»);
- при нажатой клавише «Ctrl» перетащить выделенный диапазон с помощью мыши (рядом с указателем мыши появляется знак «+»);
- использовать команды вкладки «Главная» на ленте, команды контекстного меню или сочетания клавиш на клавиатуре.

Формулой в Excel называется последовательность начинающихся со знака равенства «=» символов, включающая текст и числа, адреса ячеек (ссылки на значения, хранящиеся в ячейках), функции и знаки операций.

Результатом работы формулы является новое значение, которое выводится как результат вычисления формулы по уже имеющимся данным.

В формулах MS Excel используют следующие виды ссылок на ячейки:

- относительные (A1);
- абсолютные (\$F\$1);
- смешанные (\$F1 или F\$1);
- по имени.

В ссылке ячейки на другой лист или книгу перед именем ячейки добавляется название книги и(или) листа, отделенных знаком «!».

Функции в Excel – заранее определенные формулы, которые выполняют вычисления по заданным величинам, называемым аргументами, в указанном порядке.

Структура функции в формуле начинается с имени функции, далее идут открывающаяся скобка, список аргументов, закрывающаяся скобка.

Для добавления стандартной функции Excel в формулу:

- выполнить команду «Вставить функцию» на вкладке ленты «Формулы» или в строке формул;
- выбрать функцию из списка последних использованных функций в панели адреса;
- набрать имя функции вручную.

Для фильтрации данных следует выделить ячейку в строке с заголовками таблицы, затем перейти на вкладку «Данные» группы «Сортировка и фильтр»



и щелкнуть на кнопке «*Фильтр*» (либо «*Главная*» – «*Рецензирование*» – «*Фильтр*»).

При преобразовании диапазона данных в таблицу (выделить диапазон выполнить «*Вставка*» – «*Таблица*») в заголовок автоматически добавляются инструменты фильтрации и сортировки.

В Microsoft Excel имеется возможность графического представления данных в виде диаграммы. Диаграммы связаны с данными листа, на основе которых они были созданы, и изменяются каждый раз, когда изменяются данные на листе. Диаграммы могут использовать данные несмежных ячеек.

Для построения диаграммы функции необходимо сначала построить таблицу ее значений при различных значениях аргумента, причем обычно аргумент изменяется с фиксированным значением шага.

Для создания диаграммы:

- выделить ячейки, содержащие данные, которые должны быть отражены на диаграмме (значения функции);
- вызвать меню построения в группе «*Диаграмма*» на вкладке «*Вставка*» и выбрать команду «*Все типы диаграмм*», при этом появится окно «*Вставка диаграммы*»;
- выбрать тип диаграммы.

Для изменения вида диаграммы следует выделить мышью диаграмму и перейти на вкладку «*Конструктор*» или вкладку «*Формат*». На этих вкладках доступны инструменты для настройки внешнего вида диаграммы, а также ее цвета и типа. Если необходимо изменить данные для построения диаграммы, то нужно выделить диаграмму мышью, перейти на вкладку «*Конструктор*» и нажать на кнопку «*Выбрать данные*», указать другую область; в результате диаграмма автоматически изменится.

Все вышеуказанные команды и диалоговые окна также доступны в контекстном меню, вызванном на элементах диаграммы.

Для нахождения корней уравнения их предварительно нужно их локализовать: вычислить значения функции и построить график функции. Локализованная область определяется по изменению значения функции с положительного на отрицательное и наоборот. Далее на вкладке «*Данные*» в группе «*Работа с данными*» выполнить команду «*Анализ «Что-если»*», в списке выбрать пункт «*Подбор параметра*».

В окне «*Подбор параметра*»:

- в поле «*Установить в ячейке*» ввести ссылку на ячейку, содержащую необходимую формулу со значением, приближенным к значению корня (одна из пары ячеек, где значение функции меняет знак);
- в поле «*Значение*» ввести 0 (значение функции равно 0);
- в поле «*Изменяя значение ячейки*» ввести ссылку на ячейку, значение которой нужно подобрать.

Для работы с массивами данных:

- перед вводом формулы надо выделить область на рабочем листе, куда будет выведен результат вычислений;
- ввести формулу;



– закончить ввод нажатием клавиш «Ctrl» + «Shift» + «Enter», при этом Excel заключит формулу в строке формул в фигурные скобки: { = ... }.

При работе с массивами формула действует на все ячейки диапазона. Нельзя изменять отдельные ячейки в операндах формулы.

В Excel имеются следующие специальные функции для работы с матрицами:

- МОБР (minverse) – обратная матрица;
- МОПРЕД(mdeterm) – определитель матрицы;
- МУМНОЖ (mmult) – матричное произведение двух матриц;
- ТРАНСП (transpose) – транспонированная матрица.

14.2 Порядок выполнения работы

1 Изучить структуру интерфейса Excel и его настройку.

2 Сохранить книгу под своей фамилией. Выполнить удаление и добавление листов, переименовать их.

3 Заполнить и отформатировать таблицу по образцу (рисунок 14.1).

Таблица 14.1 – Образец заполнения таблицы

A	B	C	D	E	F
Марка автомобиля	Тип топлива	Цена 1л топлива	Пробег	Расход горючего на 100 км	Расходы на топливо

4 Ввести не менее десяти записей. Вычислить расходы на топливо. Установить точность отображения чисел до двух знаков после запятой. Выделить шапку таблицы цветом. Выполнить обрамление таблицы толщиной 2-пт. К столбцу C применить «Проверку данных»: вводимые данные должны быть более 1,5 и менее 2. Выполнить сортировку данных таблицы.

5 Скопировать ячейки с расчетами на лист 2 и преобразовать скопированный диапазон в таблицу. Выбрать автомобили с типом топлива ДТ, далее, используя меню таблицы, выполнить сортировку.

6 Вычислить выражения по заданию преподавателя при различных значениях аргументов: на листе 1 задать значения аргумента, на листе 2 вычислить значения и построить в одной системе координат графики функций для заданного на листе 1 аргумента по вариантам: для Y – гистограмму, для Z – график (при вычислении Z использовать логические функции).

7 На листе 3 выполнить решение нелинейного уравнения по вариантам.

8 На листе 4 выполнить вычисления с диапазонами по вариантам.

Контрольные вопросы

1 Табличный процессор Excel. Каковы его функции, структура окна?

2 Понятие адреса и виды ссылок на адреса ячеек.

3 Какие действия можно выполнять с рабочими листами?

4 Как изменять параметры строк и столбцов?

- 5 Каковы особенности ввода формул и данных в ячейки?
- 6 Каков порядок выделения, копирования и перемещения ячеек?
- 7 Что такое формат ячеек. Как он задается?
- 8 Каков порядок построения и редактирования диаграммы?
- 9 Каковы основные этапы вычисления выражений с массивами?

15 Решение систем ЛАУ численными методами. Реализация методов Гаусса и LU-разложения

Цель работы: изучить возможности среды MATLAB по решению систем линейных алгебраических уравнений, получить навыки решения систем ЛАУ, используя программные возможности и стандартные функции среды MATLAB.

15.1 Общие сведения

Алгебраические задачи могут быть классифицированы:

- по числу решаемых уравнений;
- по типу и количеству ожидаемых ответов.

В случае одного уравнения задачу называют линейной, алгебраической или трансцендентной.

Все численные методы решения алгебраических задач делятся на прямые (точные) и итерационные методы.

Прямые методы приводят к решению за конечное число операций, а если при этом не делать округлений, то решение задачи получится точным.

Итерационные методы позволяют находить решение приближенно, в результате построения последовательности, сходящейся к решению.

Для вычисления СЛАУ в MATLAB требуется, чтобы определитель матрицы, составленной из коэффициентов левой части, не был равен нулю: **det**(<имя матрицы>).

Расчет выполняется:

- оператором \backslash – левое деление: $x = A \backslash b$;
- оператором $/$ – правое деление: $x = b / A = (A \backslash b)$;
- функцией: $x = \mathbf{mldivide}(A, b)$;
- перемножением обратной матрицы на матрицу свободных членов: $x = \mathbf{inv}(A) * B$ или $x = A^{-1} * B$.

Расчет с использованием встроенной функции **lu**:

– $[L, U] = \mathbf{lu}(A)$, которая находит LU-разложение для произвольной квадратной матрицы A в виде произведения нижней треугольной матрицы L и верхней треугольной матрицы U , так, что $A = L * U$;

– $[L, U, P] = \mathbf{lu}(A)$ находит разложение для произвольной квадратной матрицы A в виде трех составляющих – нижней треугольной матрицы L , верхней треугольной матрицы U и матрицы перестановок P , так, что $P * A = L * U$.



Линейные алгебраические уравнения, трансцендентные уравнения и их системы можно решать, используя команду:

- **solve**(E1, E2, ..., EN);
- **solve**(E1, E2, ..., EN, var1, var2, ..., varN).

E1, E2, ..., EN – символьные выражения или переменные, в которых они содержатся, var1, ..., varN – переменные, относительно которых следует решить систему уравнений $E1=0, E2=0, \dots, EN=0$.

15.2 Порядок выполнения работы

- 1 Получить индивидуальное задание по вариантам.
- 2 Создать функции, реализующие вычисления методом Гаусса и LU-разложения.
- 3 Вызвать созданные функции в m-файле и выполнить проверку результатов расчета, используя стандартные функции MATLAB.

Контрольные вопросы

- 1 Что называют численными методами?
- 2 Каковы этапы решения инженерной задачи на ЭВМ?
- 3 Как классифицируются задачи и численные методы?
- 4 Что называют СЛАУ?
- 5 Каков порядок решения СЛАУ методами Гаусса и LU-разложения?
- 6 Какие стандартные функции применяются для решения СЛАУ?

16 Решение нелинейных алгебраических уравнений и систем. Оценка точности результата

Цель работы: изучить возможности среды MATLAB по решению систем нелинейных уравнений и их систем, получить практические навыки решения нелинейных уравнений, используя средства программирования и стандартные функции среды MATLAB.

16.1 Общие сведения

Нелинейным называется уравнение вида $\varphi(x) = 0$, где $\varphi(x)$ – некоторая нелинейная функция. Методы решения нелинейных уравнений делятся на прямые и итерационные. Прямые методы позволяют записать корни уравнения в виде точных аналитических формул. Для большинства нелинейных уравнений найти решение в явном виде невозможно, поэтому используются итерационные методы – последовательное приближение к точному решению.

При решении таких задач на ЭВМ в алгоритмах программ реализуют, например, численные методы половинного деления и касательных.



В пакете MATLAB решение нелинейных уравнений и систем выполняют с использованием стандартных функций:

- **roots**(p), p – коэффициенты полинома;
- **fzero**('имя функции', x), 'имя функции' – имя функции в символьном виде, x – интервал поиска или начальное приближение к значению корня;
- **solve**(E1, E2, ..., EN), E1, E2, ..., EN – символьные выражения или переменные;
- **fsolve**(fun,x0), fun – имя функции, которая описывает правую часть системы уравнений в символьной форме, x0 – задает начальное приближение для процедуры поиска решения.

16.2 Порядок выполнения работы

- 1 Получить индивидуальное задание по вариантам.
- 2 Создать функции нахождения корней уравнений с заданной точностью методами половинного деления и касательных.
- 3 В m-файле:
 - организовать ввод исходных данных для расчета в режиме диалога с пользователем;
 - построить график функции, определить количество корней и локализовать их расположение;
 - вызвать созданную функцию и вывести результат.
- 4 Выполнить проверку полученного решения с помощью стандартных функций roots, fzero, fsolve и solve.
- 5 Создать программу для решения системы нелинейных уравнений, используя функцию fsolve, начальное приближение определить графическим методом.

Контрольные вопросы

- 1 Дайте определение нелинейного уравнения.
- 2 Какие методы нелинейных уравнений Вам известны?
- 3 Каковы алгоритмы решения методами половинного деления, касательных?
- 4 Каковы основные этапы решения?
- 5 Каковы основные функции для решения нелинейных уравнений в MATLAB. Каковы их синтаксис и работа?



17 Интерполяция и аппроксимация экспериментальных данных методом наименьших квадратов

Цель работы: изучить возможности среды MATLAB по обработке экспериментальных данных, получить практические навыки аппроксимации и интерполяции табличных функций, построения регрессионных моделей и оценки качества аппроксимации с использованием стандартных функции среды MATLAB и средств программирования.

17.1 Общие сведения

В практической деятельности при изучении зависимости между различными параметрами исследуемых объектов получают зависимость между исследуемыми параметрами в виде таблицы – табличную функцию.

Для расчетов нужно заменить табличную функцию аналитической – найти эмпирическую формулу, связывающую между собой соответствующие значения. Получение такой аналитической функции называют регрессией.

Для замены табличной функции аналитической используются методы аппроксимации и интерполяции.

В MATLAB интерполяция реализована функциями:

- **interp1** (x,y, xi), x и y – массивы из табличных данных (координаты точек), xi – массив промежуточных точек, координаты которых находят;
- **interp1** (x,y,xi, metod), metod – определяет метод построения.

Интерполяция сплайнами в MATLAB также осуществляется с помощью команды spline:

- **spline**(x, y, xi), параметры аналогичны команде interp1.

В MATLAB для приближения данных методом наименьших квадратов используется функция **polyfit**(x, y, n), x – независимая переменная; y – зависимая переменная; n – степень полинома.

Для вычисления значений полинома можно использовать функцию **polyval**(p, X), p – вектор коэффициентов регрессионной модели, X – вектор, показывающий интервал табличных значений (может выходить за его пределы и даст экстраполяцию данных).

17.2 Порядок выполнения работы

1 Используя стандартные средства MATLAB, выполнить интерполяцию табличной функции по вариантам (линейную, полиномами и сплайнами), построить на одном листе графики заданной таблично функции в первом подокне, а затем в трех дополнительных подокнах отобразить каждый из результатов интерполяции.

2 Используя программный способ реализации метода наименьших квадратов, аппроксимировать табличную функцию по вариантам линейно:

- реализовать МНК в виде функции;



- в основном m-файле организовать ввод исходных данных, вызов созданной функции, построение кривой регрессионной модели аппроксимирующей функции;

- построить на одном листе графики заданной таблично функции и регрессионной модели аппроксимирующей функции;

- сравнить результаты.

Используя стандартные средства MATLAB:

- выполнить аппроксимацию табличной функции полиномами разной степени (1...n-1);

- сравнить результаты;

- получить и записать уравнение аппроксимирующих полиномов;

- для каждого полинома определить, какая была допущена ошибка при приближении данных;

- построить на одном листе графики заданной таблично функции и регрессионной модели аппроксимирующей функции.

Контрольные вопросы

1 Что такое табличная функция?

2 С какой целью строят регрессионную модель табличной функции?

3 В чем отличие аппроксимации и интерполяции?

4 Каковы основные методы аппроксимации и интерполяции?

5 Каковы основные функции MATLAB для построения регрессионных моделей? Каковы их синтаксис и работа?

18 Вычисление определенного интеграла методом трапеции и методом Симпсона

Цель работы: изучить возможности среды MATLAB по численному интегрированию функций, получить практические навыки составления программ, реализующих численный метод трапеции и метод Симпсона, и использования стандартных функции среды MATLAB для интегрирования.

18.1 Общие сведения

В инженерных задачах получить значение интеграла в аналитическом виде удается редко, и для вычисления используют его геометрический смысл: вычисление интеграла равносильно вычислению площади криволинейной трапеции. Производят замену исходной подынтегральной функции $f(x)$ на функцию, интеграл от которой легко вычисляется аналитически.

При замене подынтегральной функции на полином нулевой, первой и второй степени получают соответственно методы прямоугольников, трапеций, парабол (Симпсона).



Метод трапеций заключается в замене на каждом элементарном отрезке подынтегральной функции на линейную функцию, график функции представляется в виде ломаной (рисунок 18.1, а).

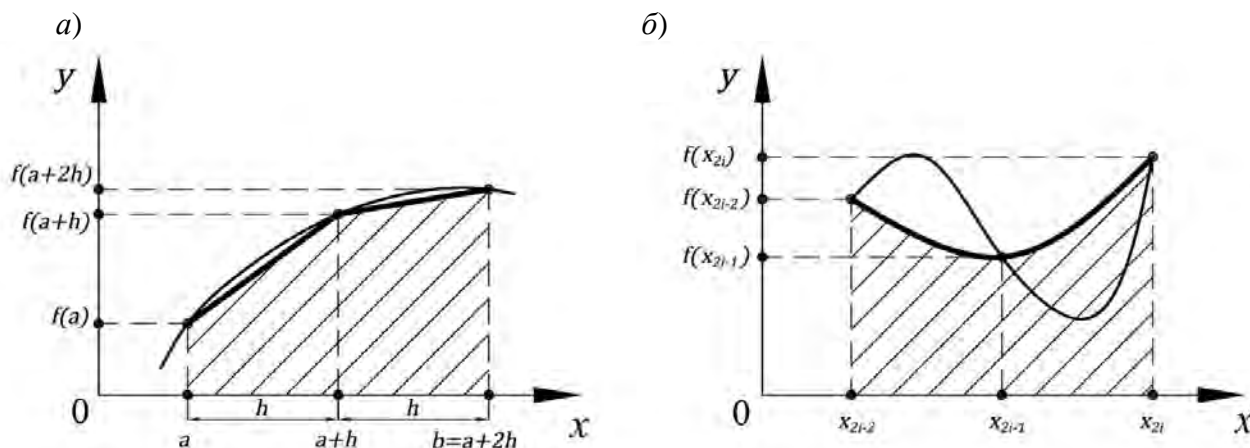


Рисунок 18.1 – Способы замены подынтегральной функции

Площадь под графиком функции аппроксимируется прямоугольными трапециями и описывается зависимостью

$$\int_a^b f(x)dx = h \cdot \sum_{i=1}^n \frac{f(a + (i-1) \cdot h) + f(a + i \cdot h)}{2}. \quad (18.1)$$

Метод парабол (Симпсона) дает более точное приближение, чем метод трапеций, на каждом элементарном отрезке подынтегральная функция приближается квадратичной параболой (рисунок 18.1, б) и описывается зависимостью

$$\int_a^b f(x)dx = \frac{h}{3} (f(x_{2i-2}) + \sum_{i=1}^n 4f(x_{2i-1}) + 2 \sum_{i=1}^{n-1} f(x_{2i}) + f(x_{2n})). \quad (18.2)$$

Порядок вычисления интегралов с заданной точностью следующий:

1) в первом приближении отрезок интегрирования разбивают на n частей (можно начать с 2-х), значение интеграла, вычисленное при n -разбиении обозначим через S_1 ;

2) для оценки точности, с которой вычислено значение интеграла, необходимо найти второе приближение. Для этого увеличим n в 2 раза: $n = 2n$. Значение интеграла, вычисленное при $2n$ разбиений обозначим через S_2 .

3) для достижения заданной точности ϵ следует проверить условие $|S_1 - S_2| < \epsilon$;

4) если условие $|S_1 - S_2| < \epsilon$ выполняется, то S_2 принимается за искомое значение интеграла;

5) если условие $|S1 - S2| < \epsilon$ не выполняется, то последнее полученное значение $S2$ считается предыдущим, т. е. $S1 = S2$, и число точек деления отрезка $[a; b]$ удваивается, $n = 2n$ для выполнения следующей итерации;

б) вычисляется новое значение $S2$;

7) процесс удвоения n , вычисления $S2$ и замены его с $S1$ следует продолжать до тех пор, пока условие $|S1 - S2| < \epsilon$ не станет истинным.

В MATLAB метод трапеций реализован функциями **cumtrapz(x,y)**, **cumtrapz(y)**, **trapz(x, y)**, **trapz(y)**.

В приведенных функциях y – массив значений функции (вектор или матрица), x – вектор значений аргумента.

Функция **trapz** вычисляет интеграл, с постоянным шагом интегрирования, равным единице; если шаг отличен от единицы, но постоянен, достаточно вычисленный интеграл умножить на h .

В MATLAB интегрирование с помощью метода Симпсона производит функция **quad**. При обращении к ней шаг интегрирования не задается, используется требуемая точность вычисления интеграла:

– **quad(f,a,b)**;

– **[I,cnt]=quad(f,a,b)**.

В приведенных функциях f – указатель на подынтегральную функцию в символьном виде, a и b – определяют пределы интегрирования, I – значение интеграла, cnt – количество итераций.

18.2 Порядок выполнения работы

1 Используя численные методы приближенного вычисления интегралов:

– составить программу для вычисления интеграла с заданной точностью $\epsilon = 10^{-5}$ по вариантам;

– составить программу для вычисления интеграла по вариантам с указанием количества элементарных отрезков интегрирования методом Симпсона.

2 Используя стандартные функции MATLAB:

– найти определенный интеграл для табличной функции по заданию преподавателя;

– найти определенный интеграл для подынтегральной функции, по вариантам используя разный шаг и точность интегрирования. Сравнить результаты.

Контрольные вопросы

1 Каковы основные методы численного вычисления интегралов?

2 В каких случаях используют численное интегрирование?

3 Каковы основные функции MATLAB для вычисления интегралов?



19 Решение ОДУ и их систем. Визуализация результатов вычислений

Цель работы: изучить возможности среды MATLAB и получить практические навыки решения ОДУ и их систем средствами среды MATLAB.

19.1 Общие сведения

Дифференциальным называется уравнение, связывающее независимые переменные, их функции и производные этой функции, если дифференциальное уравнение имеет одну независимую переменную, то оно называется обыкновенным дифференциальным уравнением.

Наивысший порядок производных, входящих в уравнение, называется порядком дифференциального уравнения.

Решением дифференциального уравнения называется такая дифференцируемая функция, которая при подстановке в исходное уравнение вместо неизвестной функции обращает уравнение в тождество.

Решение обыкновенных дифференциальных уравнений в MATLAB можно осуществить:

- реализовав алгоритм численного метода Рунге – Кутта с помощью программирования;
- с помощью встроенных функций.

Формулы метода Рунге–Кутта первого и второго порядков позволяют начиная от начального условия (x_0, y_0) найти последовательно величины y_1, y_2, \dots, y_n с шагом h и, таким образом, решить задачу:

$$y_{i+1} = y_i + h \cdot f(x_i, y_i), \quad i = 0 \dots n - 1. \quad (19.1)$$

$$y_{i+1} = y_i + \frac{h}{2} \cdot f(x_i, y_i) + \frac{h}{2} f(x_{i+1}, y_{i+1}), \quad i = 0 \dots n - 1. \quad (19.2)$$

Для решения дифференциальных уравнений и систем в MATLAB предусмотрены следующие стандартные функции:

- **ode45** (f, interval, X0, [options]);
- **ode23** (f, interval, X0, [options]).

В приведенных функциях f – вектор-функция для вычисления правой части уравнений, $interval$ – массив из двух чисел, определяющий интервал интегрирования дифференциального уравнения или системы, X_0 – вектор начальных условий системы дифференциальных уравнений, $options$ – параметры управления ходом решения.

Аналитическое решение обыкновенных дифференциальных уравнений произвольного порядка и их систем осуществляет функция **dsolve**(eqn, cond, value), eqn – ОДУ в символьном виде; cond – начальные (граничные условия), value – независимая переменная (по умолчанию t).



19.2 Порядок выполнения работы

Решить ОДУ по вариантам с использованием стандартных функций MATLAB.

Контрольные вопросы

- 1 Что называют дифференциальным уравнением?
- 2 Каковы основные функции MATLAB для решения ОДУ?

20 Численное дифференцирование функции одной переменной. Моделирование случайных величин с заданными свойствами. Визуализация результатов

Цель работы: изучить возможности среды MATLAB по дифференцированию функций, получить практические навыки вычисления производных функций.

20.1 Общие сведения

Процесс дифференцирования подразумевает нахождение производной. Производная характеризует скорость изменения функции в данной точке и определяется как предел отношения приращения функции к приращению ее аргумента. Геометрический смысл производной – это тангенс угла наклона касательной прямой к графику функции.

Численное дифференцирование применяется:

- когда функцию невозможно продифференцировать аналитически;
- когда функция задана таблицей;
- при решении дифференциальных уравнений, поиске точек экстремума функций.

Наибольшее распространение получили численные методы односторонней (рисунок 20.1, *a*) и двухсторонней разностей (рисунок 20.1, *б*), второй дает более точное решение.

Формула метода односторонней разности

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h}. \quad (20.1)$$

Формула метода двухсторонней разности

$$f'(x_0) = \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1}))}{(x_{i-1} - x_i)^2}. \quad (20.2)$$



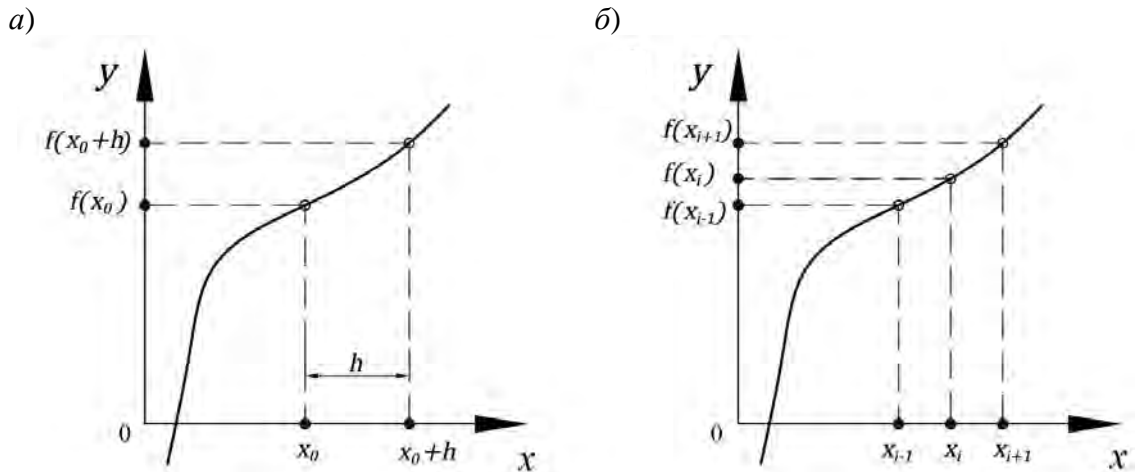


Рисунок 20.1 – Расчетные схемы методов

К стандартным средствам MATLAB для нахождения производной относится функция для вычисления конечной разности **diff(x)**, x – одномерный массив вида $x = [x(1) \ x(2) \ \dots \ x(n)]$.

При наличии пакета символьных вычислений MATLAB возможно дифференцирование функцией **diff(S)** символьного пакета, которая дифференцирует функцию S по свободной переменной в обратном порядке, если функция S содержит переменные a, b , то дифференцирование будет выполнено по переменной b . Переменная x имеет абсолютный приоритет.

Для моделирования случайных величин по равномерному закону используют функцию **rand**, или **rand(n)**, или **rand(m,n)**.

В приведенных функциях n и m – переменные, определяющие размерность вектора или матрицы.

Функция **randn()** генерирует массив с элементами, распределенными по нормальному закону; параметры аналогичны функции **rand**.

Для того чтобы получить экспоненциальное распределение с математическим ожиданием MU , используется функция **exprnd(MU)**, или **exprnd(MU,m)**, или **exprnd(MU,m,n)**.

Для того чтобы получить гамма распределение, используется функция **gamrnd(A,B)**, или **gamrnd(A,B,m)**, или **gamrnd(A,B,m,n)**.

A и B матрицы параметров, n и m – переменные, определяющие размерность вектора или матрицы.

20.2 Порядок выполнения работы

1 Выполнить численное дифференцирование функции по вариантам с использованием метода конечных разностей, шаг и интервал дифференцирования выбрать самостоятельно.

2 Выполнить численное дифференцирование стандартной функцией.

3 Сравнить результаты выполнения, построить графики.

4 Смоделировать случайные величины различными стандартными функциями, визуализировать результат моделирования в графической форме.

Контрольные вопросы

- 1 Когда используется численное дифференцирование?
- 2 Каковы основные функции для нахождения производных?
- 3 Для чего используют моделирование случайных величин?
- 4 Каковы основные функции для моделирования случайных величин?

21 Создание презентации по изученному материалу

Цель работы: получить практические навыки создания презентации.

21.1 Общие сведения

Основные сведения приведены в лабораторной работе № 12.

21.2 Порядок выполнения работы

- 1 Создать слайд, выбрать дизайн с параметрами, указанными в лабораторной работе № 12.
- 2 Создать слайды с соответствующими заголовками, содержащие:
 - маркированный список выполненных в семестре работ;
 - скриншоты окон редактора с созданными программами и графическими окнами если выполнялась визуализация расчетов;
 - таблицы, содержащие наименование работы и оценку, полученную при защите работы;
- 3 Далее полностью следовать порядку, указанному в лабораторной работе № 12.

Контрольные вопросы

- 1 Каковы основные этапы создания презентаций?
- 2 Каковы основные возможности PowerPoint по созданию и редактированию презентаций?



Список литературы

- 1 Информатика. Базовый курс: учебное пособие / Под ред. С. В. Симоновича. – 3-е изд. – Санкт-Петербург: Питер, 2012. – 640 с.: ил.
- 2 **Акулов, О. А.** Информатика. Базовый курс: учебник для вузов / О. А. Акулов, Н. В. Медведев. – 5-е изд., испр. и доп. – Москва: Омега-Л, 2008. – 574 с.
- 3 Информатика: учебное пособие / Г. Н. Хубаев [и др.]; под ред. Г. Н. Хубаева. – 3-е изд., перераб. и доп. – Ростов-на-Дону: МарТ, 2010. – 288 с.
- 4 **Потемкин, В. Г.** MATLAB 6: среда проектирования инженерных приложений / В. Г. Потемкин. – Москва: ДИАЛОГ; МИФИ, 2003. – 448 с.
- 5 **Дьяконов, В. П.** MATLAB 6.5 SP 1/ 7/ 7 SP1/ 7 SP2+Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики. – Москва: СОЛОН-ПРЕСС, 2006. – 456 с.
- 6 **Гулятьев, А.** Визуальное моделирование в среде MATLAB: учебный курс / А. Гулятьев. – Санкт-Петербург: Питер, 2002. – 432 с.
- 7 **Кетков, Ю. Л.** MATLAB 7. Программирование, численные методы / Ю. Л. Кетков. – Санкт-Петербург: БХВ-Петербург, 2005. – 734 с.

