Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 08.09.2026 03:46:57
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«25» 09 2023 г.

СТАНДАРТНЫЕ ПРОГРАММНЫЕ СРЕДСТВА В ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ БИОТЕХНИЧЕСКИХ СИСТЕМ

Методические указания по выполнению лабораторных работ
для студентов направления подготовки
12.03.04 «Биотехнические системы и технологии»

Курск 2023

УДК 004.93:61

Составители: О.В. Шаталова.

Рецензент
Кандидат технических наук, доцент *Т.Н. Конаныхина*

Стандартные программные средства в имитационном моделировании биотехнических систем: методические указания по выполнению лабораторных работ для студентов направления подготовки 12.03.04 «Биотехнические системы и технологии» / Юго-Зап. гос. ун-т; сост.: О.В. Шаталова. - Курск, 2023. - 367 с.

Предназначено для студентов направления подготовки 12.03.04 «Биотехнические системы и технологии» по дисциплине «Стандартные программные средства в имитационном моделировании биотехнических систем». Может быть использована аспирантами, обучающимися по направлениям подготовки 2.3.1. Системный анализ, управление и обработка информации, статистика и 2.2.12. Приборы, системы и изделия медицинского назначения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16.
Усл. печ. л. 21,33. Уч.-изд. л. 19,32. Тираж 100 экз. Заказ 941 . Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

6 семестр

Лабораторная работа № 1
Ознакомление с пакетом Excel. Выполнение операций с данными. Построение диаграмм

1.1 Базовые элементы Microsoft Excel. Выполнение простейших операции с данными в Microsoft Excel

1.1.1 Цель работы

Познакомиться с базовыми элементами Microsoft Excel; изучить основные понятия Microsoft Excel; выработать навыки выполнения простейших операции с данными в Microsoft Excel.

1.1.2 Теоретическое введение и упражнения

Элементы окна и основные понятия Microsoft Excel

Запустите Microsoft Excel (Пуск - Программы - Microsoft Office). На экране компьютера открылось рабочее окно Excel изображенное на рисунке 1.1.1.

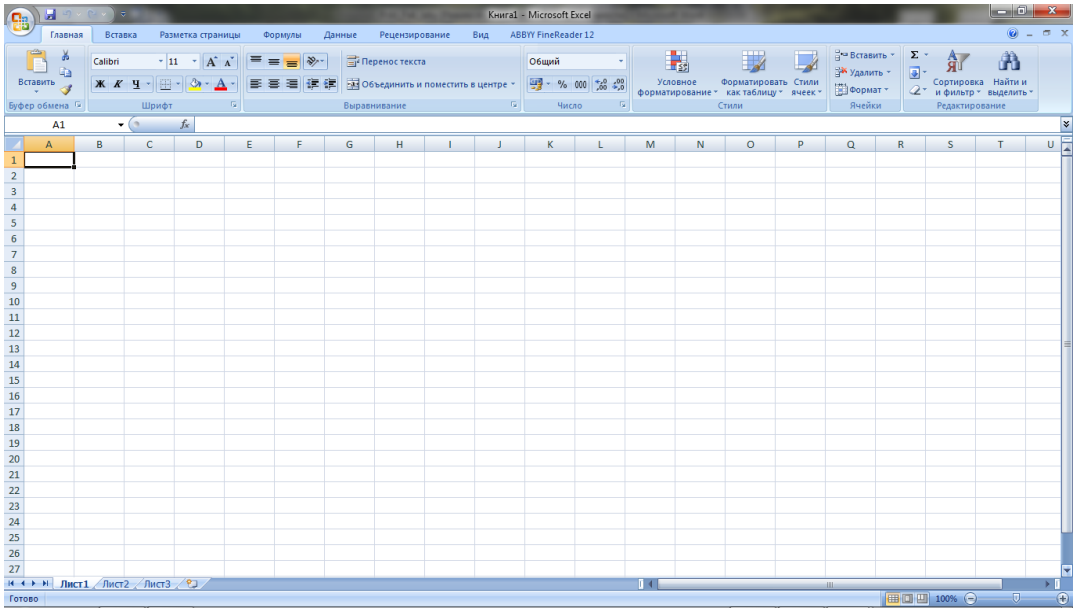


Рисунок 1.1.1 - Рабочее окно Microsoft Excel

Главный элемент пользовательского интерфейса Microsoft Excel представляет собой ленту, рисунок 1.1, которая идет вдоль верхней части окна каждого приложения, вместо традиционных меню и панелей инструментов.

С помощью ленты можно быстро находить необходимые команды (элементы управления: кнопки, раскрывающиеся списки, счетчики, флажки и т.п.). Команды упорядочены в логические группы, собранные на вкладках.

Заменить ленту панелями инструментов или меню предыдущих версий приложения Microsoft Excel нельзя. Удалить ленту также нельзя. Однако чтобы увеличить рабочую область, ленту можно скрыть (свернуть).

Упражнение 1.1.1. Настройка ленты.

Нажмите кнопку *Настройка панели быстрого доступа*, рисунок 1.1.2

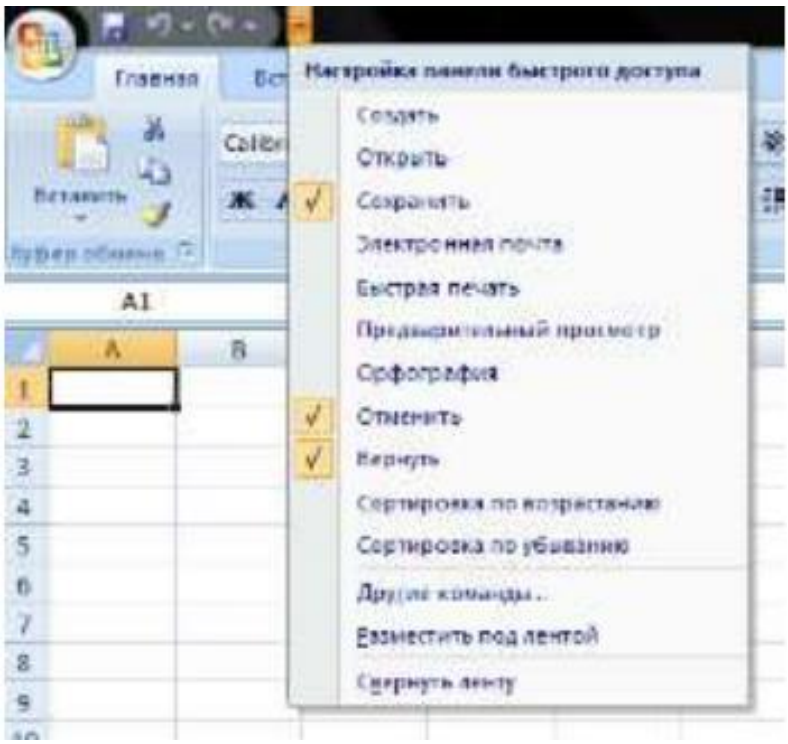


Рисунок 1.1.2 – Панель быстрого доступа

В меню выберите команду *Свернуть ленту*. Лента будет скрыта, названия вкладок останутся.

Для использования ленты в свернутом состоянии щелкните по названию нужной вкладки, а затем выберите параметр или команду, которую следует использовать. Например, при свернутой вкладке можно выделить текст в документе Excel, щелкнуть вкладку *Главная* и в группе *Шрифт* выбрать нужный размер шрифта. После выбора размера шрифта лента снова свернется.

Чтобы быстро свернуть ленту, дважды щелкните имя активной вкладки. Для восстановления ленты дважды щелкните вкладку.

Чтобы свернуть или восстановить ленту, можно также нажать комбинацию клавиш Ctrl+F1.

Содержание ленты для каждой вкладки постоянно и неизменно. Нельзя ни добавить какой-либо элемент на вкладку, ни удалить его оттуда. Несмотря на скрытие отдельных элементов ленты, все они остаются доступны. Достаточно щелкнуть по значку или стрелке кнопки названия группы, после чего будут отображены все элементы группы.

Кнопка "Office"

Кнопка "*Office*" расположена в левом верхнем углу окна. При нажатии кнопки отображается меню основных команд для работы с файлами, список последних документов, а также команда для настройки параметров приложения (например, *Параметры Excel*, рисунок 1.1.3.

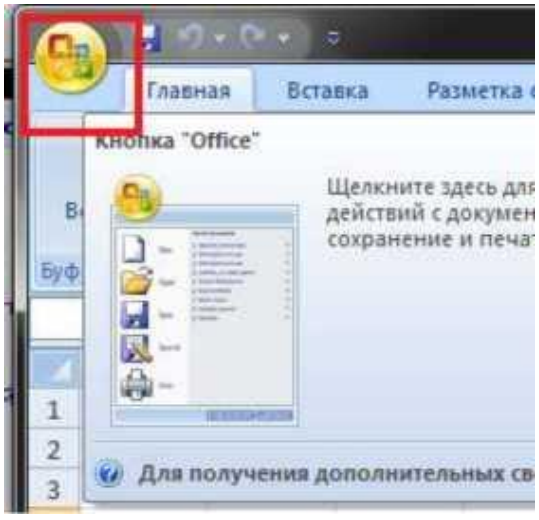


Рисунок 1.1.3 - Кнопка "Office"

Панель быстрого доступа

Панель быстрого доступа по умолчанию расположена в верхней части окна Excel и предназначена для быстрого доступа к наиболее часто используемым функциям. По умолчанию панель содержит всего три кнопки: *Сохранить*, *Отменить*, *Вернуть* (*Повторить*). Панель быстрого доступа можно настраивать, добавляя в нее новые элементы или удаляя существующие.

Упражнение 1.1.2. Настройка панели быстрого доступа.

Нажмите кнопку *Настройка* панели быстрого доступа.

В меню выберите наименование необходимого элемента, рисунок 1.2. Элементы, отмеченные галочкой, уже присутствуют на панели.

Для добавления элемента, отсутствующего в списке, выберите команду *Другие команды*.

В разделе *Настройка* окна *Параметры Excel* в раскрывающемся списке *Выбрать команды из* выберите вкладку, в которой расположен добавляемый элемент, затем выделите элемент в списке и нажмите кнопку *Добавить.*

Для добавления на панель любого элемента из любой вкладки можно также щелкнуть по этому элементу правой кнопкой мыши и в контекстном меню выбрать команду *Добавить на панель быстрого доступа.*

Для удаления элемента из панели достаточно щелкнуть по нему правой кнопкой мыши и в контекстном меню выбрать команду *Удалить с панели быстрого доступа*.

Вкладки

По умолчанию в окне отображается семь постоянных вкладок: *Главная, Вставка, Разметка страницы, Ссылки, Рассылки, Рецензирование, Вид.*

Для перехода к нужной вкладке достаточно щелкнуть по ее названию (имени).

Каждая вкладка связана с видом выполняемого действия. Например, вкладка *Главная*, которая открывается по умолчанию

после запуска, содержит элементы, которые могут понадобиться на начальном этапе работы, когда необходимо набрать, отредактировать и отформатировать текст. Вкладка *Разметка* страницы предназначена для установки параметров страниц документов. Вкладка *Вставка* предназначена для вставки в документы различных объектов. И так далее.

Кроме того, можно отобразить еще одну вкладку: *Разработчик.*

Упражнение 1.1.3. Отобразите вкладку Разработчик.
Щелкните по кнопке *Microsoft Office.*
Выберите кнопку команды *Параметры Excel,* рисунок 1.1.4.



Рисунок 1.1.4 - Окно кнопки Microsoft Office

Во вкладке *Основные* диалогового окна *Параметры Excel* установите соответствующий флажок, рисунок 1.1.5. На вкладке *Разработчик* собраны средства создания макросов и форм, а также функции для работы с XML. Файлы и шаблоны предыдущих версий Excel могли содержать пользовательские панели инструментов. В этом случае при открытии таких файлов в Excel появляется еще одна постоянная вкладка - *Надстройки.*

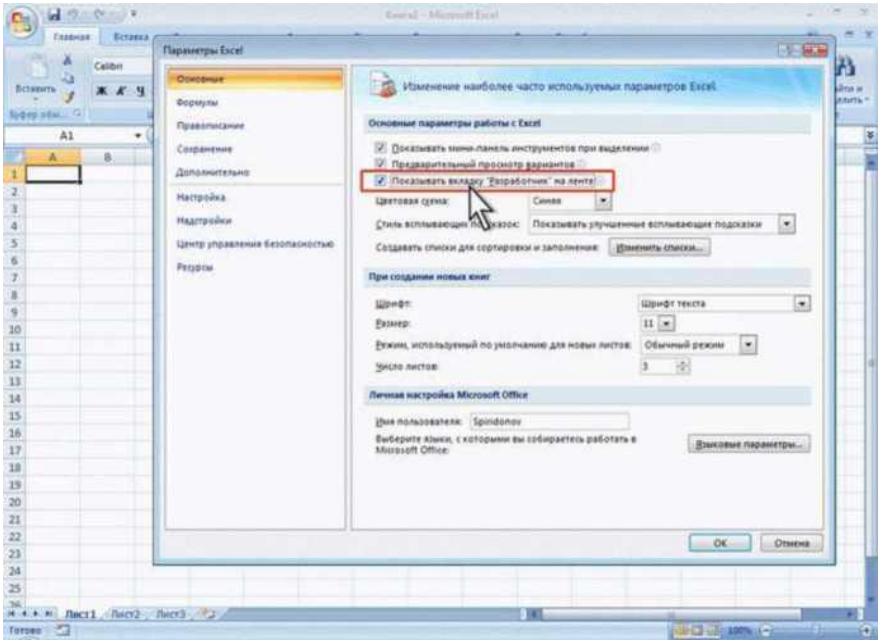


Рисунок 1.1.5 - Отображение вкладки Разработчик

Элементы управления

Элементы управления на лентах вкладках объединены в группы, связанные с видом выполняемого действия. Например, на вкладке *Главная* имеются группы для работы с буфером обмена, установки параметров шрифта, установки параметров абзацев, работы со стилями и редактирования.

Элементами управления являются обычные кнопки, раскрывающиеся кнопки, списки, раскрывающиеся списки, счетчики, кнопки с меню, флажки, значки (кнопки) группы. Кнопки используются для выполнения какого-либо действия. Например, кнопка *Полужирный* группы *Шрифт* вкладки *Главная* устанавливает полужирное начертание шрифта.

Нажатие некоторых кнопок не приводит к выполнению какого-либо действия, а только отображает меню, в котором следует выбрать нужное действие. Например, щелкнув по кнопке *Ориентация* в группе *Выравнивание* вкладки *Главная* в появившемся меню можно выбрать направление текста в ячейке, рисунок 1.1.6.

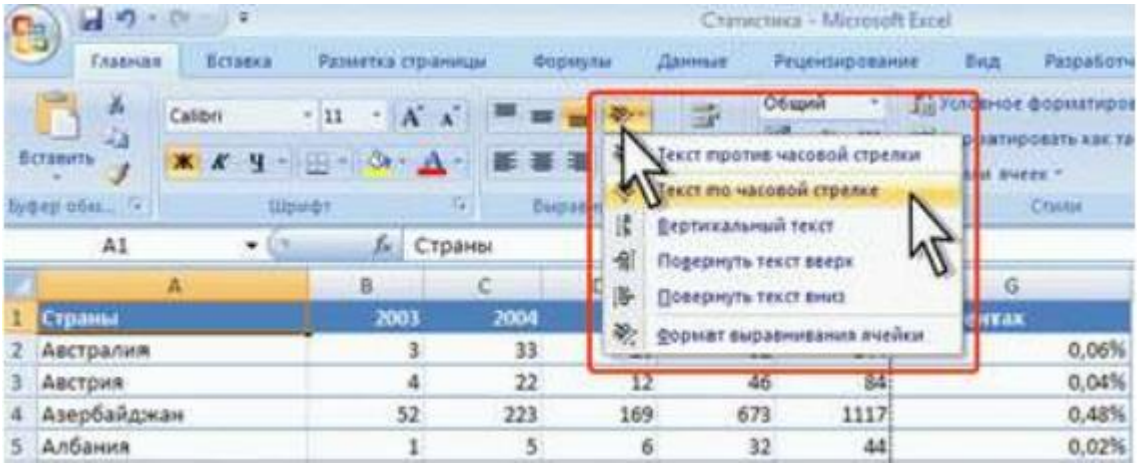


Рисунок 1.1.6 - Использование кнопки с меню

Основные понятия Microsoft Excel

Познакомьтесь с основными понятиями электронных таблиц. Для этого внимательно прочитайте нижеследующий текст и выполните некоторые действия.

Рабочее поле электронной таблицы состоит из *строк* и *столбцов*. Максимальное количество строк равно *1 048 576*, столбцов — *16 384*. Каждое пересечение строки и столбца образует *ячейку*, в которую можно вводить данные (текст, число или формулы).

Номер строки — определяет ряд в электронной таблице. Он обозначен на левой границе рабочего поля.

Буква столбца — определяет колонку в электронной таблице. Буквы находятся на верхней границе рабочего поля. Колонки нумеруются в следующем порядке: A—Z, затем AA—AZ, затем BA—BZ и т. д. Заканчивается нумерация столбцов XFD/

Ячейка — первичный элемент таблицы, содержащий данные. Каждая ячейка имеет уникальный *адрес*, состоящий из буквы столбца и номера строки. Например, адрес B3 определяет ячейку на пересечении столбца B и строки номер 3.

Указатель ячейки — светящийся прямоугольник, определяющий *текущую ячейку*. Указатель можно перемещать, по таблице как при помощи клавиатуры, так и мышью.

Текущая ячейка помечена указателем. По умолчанию ввод данных и некоторые другие действия относятся к текущей ячейке.

> Сделайте текущей ячейку *D4* при помощи мыши.

> Вернитесь в ячейку *A1* при помощи клавиш перемещения курсора.

Блок представляет собой прямоугольную область смежных ячеек. Блок может состоять из одной или нескольких строк, столбцов.

Адрес блока состоит из координат противоположных углов, разделенных двоеточием. Например: *B13:C19*, *A12:D27* или *D:F*.

Блок можно задать при выполнении различных команд или вводе формул посредством указания координат или выделения на экране.

Электронная таблица в Excel имеет трехмерную структуру. Она состоит из *листов*, как *книга*. На экране виден только один лист — верхний. Нижняя часть листа содержит ярлычки других листов. Щелкая кнопкой мыши на ярлычках листов, можно перейти к другому листу.

Сделайте текущим лист *3*.

Вернитесь к листу *1*.

Ячейка имеет сложную структуру, состоящую из пяти информационных уровней, рисунок 1.1.7.

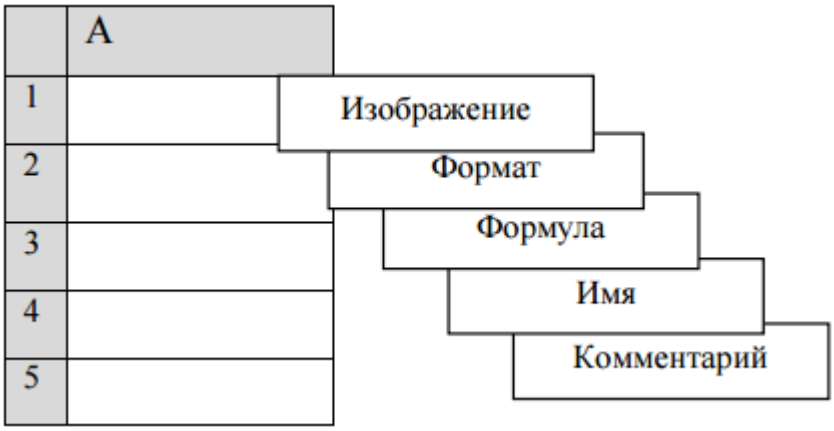


Рисунок 1.1.7 - Информационные уровни ячеек

Верхний уровень - это изображение на экране. Там отображаются отформатированные значения, такие как текст или результат вычисления формул.

Скрытое форматирование сохраняется на втором уровне ячейки. Оно включает в себя формат чисел, шрифт, вид рамки, образец, защиту ячейки.

Третий уровень содержит формулу, которая может состоять из текста, числа или математических выражений, например =A1, =A1+B2, ="Excel".

Четвёртый уровень содержит имя ячейки. Назначив имя ячейке или диапазону ячеек, можно вставлять содержимое этих ячеек и их имя в любом месте рабочей книги.

Пятый уровень - комментарий. Используется для создания текстового примечания к ячейке. После создания примечания оно высвечивается при каждом наведении курсора на данную ячейку.

Выделение ячеек

Для *выделения* с помощью *мыши*:

столбца — щелкнуть кнопкой мыши на букве — имени столбца;

нескольких столбцов — не отпуская кнопку после щелчка, протянуть мышь;

строки — щелкнуть кнопкой мыши на числе — номере строки;

нескольких строк — не отпуская кнопку после щелчка, протянуть мышь;

блока — щелкнуть кнопкой мыши на начальной ячейке блока и, не отпуская кнопку, протянуть мышь на последнюю ячейку;

рабочего листа — щелкнуть кнопкой мыши на пересечении имен столбцов и номеров строк (левый верхний угол таблицы).

Для выделения блока с помощью клавиатуры необходимо, удерживая нажатой клавишу *Shift*, нажимать на соответствующие клавиши перемещения курсора, или, нажав клавишу *F8*, войти в режим выделения и выделить блок при помощи клавиш перемещения курсора. *Esc — выход* из режима выделения.

Для *выделения нескольких несмежных блоков* необходимо: 1) выделить *первую ячейку* или блок смежных ячеек:

2) нажать и удерживать нажатой клавишу *Ctrl*;

3) выделить *следующую ячейку* или блок и т. д.;

4) отпустить клавишу *Ctrl*.

Для отмены выделения достаточно щелкнуть кнопкой мыши по любому невыделенному участку рабочего листа. Новое выделение снимает предыдущее.

Упражнение 1.1.4. Выполните выделения ячеек с помощью мыши и клавиатуры.

Выделите строку *3*.

Выделите столбец *O*.

Выделите блок *A2:E13* при помощи мыши.

Выделите столбцы *A, B, C, D*.

Выделите блок *C4:F10* при помощи клавиатуры.

Выделите рабочий лист.

Выделите одновременно следующие блоки: *F5:G10, H15:K15, C18:F20*.

Ввод данных

В Excel существуют три *типа данных*, вводимых в ячейки таблицы: *текст, число* и *формула*.

Для ввода данных необходимо переместиться в нужную ячейку и набрать данные (до *32767* символов), а затем нажать *Enter* или клавишу перемещения курсора.

Excel определяет, являются вводимые данные текстом, числом или формулой, по первому символу. Если первый символ — *буква* или знак ’ , то Excel считает, что вводится *текст*. Если первый символ цифра или знак = , то Excel считает, что вводится *число* или *формула*.

Вводимые данные *отображаются* в ячейке и строке формул и помещаются в ячейку только при нажатии *Enter* или клавиши перемещения курсора.

В отдельных случаях необходимо вводить числа, которые, по сути, являются текстом и не должны распознаваться как числа. Это могут быть, например, номера банковских счетов, номера телефонов, коды и артикулы, почтовые индексы, номера домов и квартир и т. п. Для того чтобы Microsoft Excel автоматически назначал таким числам текстовый формат, перед числом необходимо ввести знак апострофа ('). Этот знак отображается только в строке формул, но не отображается в ячейке. Такая ячейка будет помечена в левом верхнем углу зеленым треугольником. При выделении такой ячейки слева может появляться кнопка *Источник ошибки*. Если ввод апострофа перед числом произведен

специально, то следует щелкнуть по кнопке и выбрать команду *Пропустить ошибку*. Если апостроф перед числом введен ошибочно, то следует щелкнуть по кнопке и выбрать команду *Преобразовать в число*.

Если ширина текста больше ширины ячейки и ячейка справа пуста, то текст на экране займет и ее место. При вводе данных в соседнюю ячейку предыдущий текст на экране будет обрезан (но при этом в памяти он будет сохранен полностью).

Обратите внимание, что текст прижат к левому краю.

ПРИМЕЧАНИЕ**:** Почти все действия могут быть отменены.

Если в процессе работы произошло что-то, что вы не планировали, воспользуйтесь отменой: используя команду *Панели быстрого доступа/Отменить...* При каждом щелчке мыши по кнопке *Отмена* последовательно отменяется одна операция.

Числа в ячейку можно вводить со знаками =, +, - или без них. Если ширина введенного числа больше, чем ширина ячейки на экране, то Excel отображает его в экспоненциальной форме или вместо числа ставит символы # # # # (при этом число в памяти будет сохранено полностью).

Экспоненциальная форма используется для представления очень маленьких и очень больших чисел. Число 501000000 будет записано как 5,01E+08, что означает $5{,}0110^{8}$. Число 0,000000005 будет представлено как 5E-9, что означает $5\ 10^{-9}$.

Для ввода *дробных чисел* используется *десятичная запятая*.

В виде *формулы* может быть записано *арифметическое выражение*. Оно представляет собой последовательность чисел или ссылок на ячейки, объединенных знаками арифметических операций или функций.

Формула должна начинаться со знака = . Она может включать до 8192 символ и не должна содержать пробелов.

Для ввода в ячейку формулы C1+F5 ее надо записать как =C1+F5. Это означает, что к содержимому ячейки *C1* будет прибавлено содержимое ячейки *F5*. Результат будет получен в той ячейке, в которую занесена формула.

Упражнение 1.1.5. Определите возраст г. Курска.
В ячейку *A1* занесите текст *Возраст г. Курска*.
В ячейку *B1* занесите число *1032* (год основания Курска).
В ячейку *C1* занесите текущий год, например *2017*.
В ячейку *D1* занесите формулу *=C1-B1*.
Что за число получилось в ячейке? Это возраст Курска.

Изменение *ширины столбцов и высоты строк* можно выполнить с помощью мыши, элемента управления *Формат*, группа *Ячейки*, вкладка *Главная* или через контекстное меню. При использовании *мыши* ее указатель необходимо поместить на *разделительную линию* между именами столбцов или номерами строк. Указатель примет вид двойной черной стрелки. Затем необходимо нажать *левую* кнопку мыши и растянуть (сжать) столбец или строку.

При использовании *контекстного меню* для изменения *ширины столбцов и высоты строк* необходимо выделить строки или столбцы и вызвать контекстное мню (правой кнопкой мыши) полнить команды *Формат/Строка/Высота* или *Формат/Столбец/Ширина.*

Упражнение 1.1.6. Настройте ширину столбца и строки.

При помощи мыши измените ширину столбца *A* так, чтобы текст был виден полностью, а ширину столбцов *B, C, D* сделайте минимальной.

При помощи контекстного меню измените высоту строки номер *1* и сделайте ее равной *30*.

Сделайте высоту строки номер *1* первоначальной *12, 75*.

Редактирование данных

Редактирование данных может осуществляться как в процессе ввода в ячейку, так и после ввода.

Если *во время ввода данных* в ячейку допущена ошибка, то она может быть исправлена удалением неверных символов при помощи клавиши удаления последнего набранного символа и

набором символов заново. Клавишей *Esc* можно отменить ввод данных в ячейку и записать их заново.

Чтобы *отредактировать* данные *после завершения ввода* (после нажатия клавиши Enter), необходимо переместить указатель к нужной ячейке и нажать клавишу *F2* для перехода в режим редактирования или щелкнуть кнопкой мыши на данных в строке формул. Далее необходимо отредактировать данные и для завершения редактирования нажать *Enter*.

Упражнение 1.1.7. Определите возраст Курска в *2002* году.

Замените текущий год в ячейке *C1* на *2002*. В ячейке *D1* появился новый возраст города (в 2002 году).

ПРИМЕЧАНИЕ: При вводе новых данных *пересчет в таблице произошел автоматически.* Это важнейшее свойство электронной таблицы.

Отредактируйте текст в ячейке *A1*. Введите новый текст *Курск – соловьиный край*.

Операции со строками, столбцами, блоками

Операции со строками, столбцами, блоками могут быть выполнены с помощью вкладки *Главная*, группы *Буфер обмена* (вырезать, скопировать-вставить), а также с помощью контекстного меню (правая кнопка мыши).

Перемещая данные, необходимо указать, ЧТО перемещается и КУДА. Для выполнения перемещения требуется выделить ячейку или блок (ЧТО перемещается) . Затем поместить указатель мыши на рамку блока или ячейки (он должен принять форму белой стрелки). Далее следует нажав и удерживая левую кнопку мыши перенести блок или ячейку и отпустить кнопку мыши (в место, КУДА нужно переместить данные).

При копировании оригинал (ЧТО) остается на прежнем месте, а в другом месте (КУДА) появляется копия.

Копирование выполняется аналогично перемещению, но при нажатой клавише *Ctrl*.

При заполнении исходная ячейка или блок (ЧТО) повторяется несколько раз за одно действие. Заполнение возможно вправо или вниз.

Заполнение выполняется так же, как и перемещение, но при этом курсор должен находиться на *нижнем правом углу* ячейки или блока (принимает форму черного плюса и называется *маркером заполнения*).

Если надо *очистить только данные* (числа, текст, формулы), то достаточно нажать клавишу на клавиатуре *Delete*, предварительно выделив ячейку или блок.

Для *очистки ячейки или блока* от введенных данных можно установить указатель на ячейку или выделить блок, а затем выбрать элемент управления *Очистить*, группа *Редактирование*, вкладка *Главная*. В появившемся списке необходимо указать, что конкретно требуется очистить: данные, оформление, примечание или всё вместе.

Для *удаления столбцов, строк, блоков* нужно выделить необходимый элемент, а затем воспользоваться элементом управления *Удалить*, группа *Ячейки*, вкладка *Главная*. При удалений место строк, столбцов и блоков «охлопывается».

Упражнение 1.1.8. Выполните перемещение, копирование и очистку ячеек.

Выделите блок *A1:D1* и переместите его на строку ниже.

Верните блок на прежнее место.

Скопируйте блок *A1:D1* в строки *3, 5, 7*.

Выделите строку номер *7* и заполните выделенными данными строки *по 15-ю* включительно.

Скопируйте столбец *C* в столбцы *E, F, G*.

Выделите блок *A10:G15* и очистите его.

Очистите содержимое ячейки *O9*, используя элемент управления *Очистить*.

Удалите столбец *E*. Обратите внимание на смещение столбцов.

В Excel существует интересная *функция автозаполнения*, которая позволяет быстро вводить различные типовые последовательности (дни недели, месяцы, годы и т.д.).

Упражнение 1.1.9. Используя операцию автозаполнение сформируйте списки названий дней недели, месяцев и времен года.

В ячейку *G10* занесите текст «*январь*» и нажмите клавишу Enter.

Установите курсор в ячейку *G10*. Укажите мышью на маркер заполнения (маленький крестик в правом нижнем углу) ячейки *G10* (экранный курсор превращается в маркер заполнения).

Нажмите левую кнопку мыши и, не отпуская ее, двигайте мышь вправо, пока рамка не охватит ячейки *G10:M10*.

ПРИМЕЧАНИЕ: учитывая, что в ячейке вы напечатали «*январь*», Excel продолжил названия последующих месяцев во все выделенные ячейки, т.к. это список занесен в стандартные списки кнопка *Office/ Параметры Excel/Основные/кнопка Изменить списки*, при необходимости можно сформировать список, который Excel будет продолжать, если добавить его в существующие списки.

Добавьте список времен года и занесите времена года в ячейки *A1:A4*.

Введите в ячейки *G11:M11* дни недели, начиная *с понедельника*.

Введите в ячейки *G12:M12* года, начиная *с 1990-го*. Наберите 1990 и 1991 в ячейки *G12:K12* и заполните ячейки.

Excel позволяет вводить некоторые нетиповые последовательности, если в них удается выделить некоторую закономерность.

Упражнение 1.1.10. Создайте таблицу с данными о населении г. Курска.

Используя рисунок 1.1.8, внесите следующие данные в таблицу:

В ячейку *G15* **- заголовок «***Численность г. Курска и Курской обл***».**

В ячейки *G16:M16* - года.

В ячейки *G17: M17* - данные о населении Новотроицка по годам.

Численность г. Курска и Курской обл.							
2006	2009	2010	2011	2012	2013	2014	2015
1177627	1143329	1134990	1125648	1121563	1119262	1118915	1117378

Рисунок 1.1.8 - Пример со списками

Упражнение 1.1.11. Известны обобщенные данные о городском и сельском населении за несколько лет, они приведены на рисунке 1.1.9. Вычислите для каждого года разность между городским и сельским населением и подведите итоги по каждому столбцу.

	A	B	C
1	год	городское	сельское
2	2006	736676	440951
3	2009	734173	409156
4	2010	734753	400237
5	2011	735680	289968
6	2012	739415	382148
7	2013	743795	375467
8	2014	744446	374469
9	2015	747310	370062
10			

Рисунок 1.1.9 - Данные о городском и сельском населении

Откройте *Лист 2*. Переименуйте его, дав название «*Население*» (щелкните по названию листа правой кнопкой мыши и выполните команду «Переименовать»).

В ячейки *A1:C5* внесите исходные данные.

В ячейку *D1* введите название столбца «*Разность*».

В ячейку *D2* введите формулу для вычисления разности *=B2-C2*. По окончании ввода нажмите клавишу Enter. В ячейке D2 появился результат.

Измените данные о населении в 2006 г., например *747657*. Что произошло со значением разности? Введите начальное значение населения.

Заполните остальные ячейки столбца «Разность» соотвествующими формулами. Используйте операцию автозаполнение.

Выполните подведение итогов по каждому столбцу. Для этого в ячейку *D6* введите *«Итого»*.

Поместите в ячейку *B6* формулу для суммирования диапазона *B2:B5*. Для этого выделите ячейки *B2:B5* и нажмите на элемент управления *Сумма*).

Аналогично заполните ячейки *C6* и *D6*, рисунок 1.1.10.

	A	B	C	D	E
1	год	городское	сельское	разность	итого
2	2006	736676	440951	295725	1177627
3	2009	734173	409156	325017	1143329
4	2010	734753	400237	334516	1134990
5	2011	735680	289968	445712	1025648
6	2012	739415	382148	357267	1121563
7	2013	743795	375467	368328	1119262
8	2014	744446	374469	369977	1118915
9	2015	747310	370062	377248	1117372
10					

Рисунок 1.1.10 - Результат решения задачи

Упражнение 1.1.12. Известны обобщенные данные о больных различными психическими заболеваниями за несколько лет, они приведены на рисунке 1.1.11. Вычислите для каждого года разность между общим числом больных и имеющих конкретные заболевания (по каждому виду заболеваний) и подведите итоги по каждому столбцу.

	A	B	C	D	E	F	
1	год	всего	психозы и состояния слабс	шизофрения	псих расстр-ва непсихотического хар-ра	умственная отсталость	р
2	2005	743	290	139	227	226	
3	2008	554	219	75	189	146	
4	2009	527	208	80	161	158	
5	2010	561	239	72	161	161	
6	2011	650	335	59	169	146	
7	2012	572	294	59	143	135	
8	2013	459	249	56	106	104	
9	2014	567	257	54	164	146	
10							

Рисунок 1.1.11 – Данные о различных психических заболеваниях

Сохранение таблиц

В Microsoft Office 2007 введен новый формат файлов, основанный на языке XML. По умолчанию документы, создаваемые в Excel 2007, сохраняются с новым расширением имени файла, которое получается путем добавления суффикса "x" или "m" к привычному расширению. Суффикс "x" указывает на то, что XML-файл не содержит макросов, а суффикс "m" - что XML-файл макросы содержит. Таким образом, имена обычных файлов Excel 2007 имеют расширение xlsx, а не xls.

XML-форматы предоставляют целый ряд преимуществ - не только для разработчиков и создаваемых ими продуктов, но также для отдельных пользователей и организаций. Файлы автоматически сжимаются, и в некоторых случаях их размер может сокращаться на 75 процентов по сравнению с предыдущими версиями Excel. Формат обеспечивает улучшенное восстановление поврежденных файлов. Файлы построены по модульному принципу, при котором разные компоненты данных хранятся в файле отдельно друг от друга. Это позволяет открывать файлы даже при разрушении или повреждении какого-либо компонента файла (например, диаграммы или таблицы).

Обеспечивается более легкое обнаружение документов, содержащих макросы. Файлы, которые сохраняются с расширением xlsx, не могут содержать макросы, написанные на языке VBA (Visual Basic для приложений), следовательно, в них отсутствует угроза безопасности, связанная с такого рода внедренными кодами. Только файлы, с расширением xlsm могут содержать макросы на языке VBA, хранящиеся в обособленных частях файла.

Различные расширения имен файлов позволяют легко отличать файлы, содержащие макрос, от файлов, которые его не содержат, и облегчают обнаружение антивирусными программами файлов, содержащих потенциально опасный код.

Файлы новых форматов недоступны для работы в предыдущих версиях Excel. Для того чтобы можно было открывать и изменять файлы Excel 2007 в более ранних версиях, в них следует загрузить необходимые конвертеры файлов.

Сохранение изменений в существующем файле. Для сохранения изменений в существующем файле достаточно нажать кнопку Сохранить панели быстрого доступа.

Сохранение изменений в виде нового файла. Для сохранения нового документа в виде файла или существующего документа в виде нового файла (с другим именем и/или в другой папке) выполните следующие действия:

Сохранение файлов в других форматах. По умолчанию все файлы сохраняются в том же формате, в котором были открыты. Новые документы и файлы Excel сохраняются в формате *Документ Excel*, файлы предыдущих версий сохраняются в формате *Книга Excel 97-2003*. При сохранении можно изменить формат файла. Для этого в окне *Сохранение документа* щелкните по кнопке с указанием типа файла и в появившемся меню выберите необходимый формат.

Особенности сохранения документов Excel 2007 в формате предыдущих версий Excel. Многие возможности Excel 2007 не поддерживаются в более ранних версиях. При сохранении документа, созданного в Excel 2007, в формате *Книга Excel 97-2003* в нем будут произведены изменения. Для проверки совместимости текущего документа с предыдущими версиями Excel нажмите кнопку *Office*, выберите команду *Подготовить* и в появившемся подчиненном меню - команду *Проверка совместимости*. В окне результатов проверки совместимости будет отображен список несовместимых элементов. Необходимо внести изменения в книгу или для сохранения с потерями функциональности нажать кнопку *Продолжить*.

Упражнение 1.1.12. Выполните сохранение файла в формате *Книга Excel 97-2003*.

Нажмите кнопку *Office* **и выберите команду** *Сохранить как/ Книга Excel 97-2003***.**

В окне *Сохранение документа* перейдите к нужной папке. В поле *Имя файла* введите имя файла *tab1* (расширение имени вводить не следует) и нажмите кнопку *Сохранить*.

1.1.3 Задания для самостоятельной работы

Задание 1. Используя данные из таблицы 1.1.1, рассчитайте общий балл по ЕГЭ для каждого абитуриента.

Таблица 1.1.1 - Данные о ЕГЭ

ФИО	Результаты ЕГЭ			Сумма баллов
	Математика	Русский язык	Обществознание	
Алексеев А.Н.	54	60	56	
Иванов К.С.	67	71	68	
Кореев В.А.	80	54	76	
Малкин С.Н.	44	45	56	
Петров А.К.	81	75	80	

Задание 2. Используя данные из таблицы 1.1.2, рассчитайте средний балл по каждому студенту.

Таблица 1.1.2 - Результаты экзаменов группы 16

ФИО	Дисциплины			Средний балл
	Математический анализ	Информатика	Экономика	
Алексеев А.Н.	5	5	4	
Иванов К.С.	4	4	5	
Кореев В.А.	4	3	3	
Малкин С.Н.	3	4	4	
Петров А.К.	3	4	3	
Петухов А.В	4	3	4	
Сидоров В.В.	3	3	3	

1.1.4 Контрольные вопросы

1. Перечислите основные элементы окна Microsoft Excel.
2. Как определяется адрес ячейки и адрес блока?
3. Перечислите информационные уровни ячеек Excel и прокомментируйте каждый уровень.

4. Перечислите способы выделения ячеек, блоков ячеек.
5. Как выполняется операция “автозаполнение”?
6. Как сохранить таблицы в формате Книга Excel 97-2003?

1.2 Создание, редактирование и форматирование диаграмм в Microsoft Excel

1.2.1 Цель работы

Научиться строить диаграммы на основе таблиц; освоить приемы форматирования и редактирования диаграмм.

1.2.2 Теоретическое введение и упражнения

Одним из самых впечатляющих достоинств Excel является способность превращать абстрактные ряды и столбцы чисел в привлекательные информативные графики и диаграммы.

Диаграммы - это удобное средство графического представления данных. Диаграммы являются средством наглядного представления данных и облегчают выполнение сравнений, выявление закономерностей и тенденций данных.

Диаграммы создают на основе данных, расположенных на рабочих листах. Как правило, используются данные одного листа. Это могут быть данные диапазонов как смежных, так и не смежных ячеек. Несмежные ячейки должны образовывать прямоугольник. При необходимости, в процессе или после создания диаграммы, в нее можно добавить данные, расположенные на других листах. Диаграмма может располагаться как графический объект на листе с данными (не обязательно на том же, где находятся данные, взятые для построения диаграммы). На одном листе с данными может находиться несколько диаграмм. Диаграмма может располагаться на отдельном специальном листе.

Диаграмму можно напечатать. Диаграмма, расположенная на отдельном листе, печатается как отдельная страница. Диаграмма, расположенная на листе с данными, может быть напечатана вместе с данными листа или на отдельной странице.

Диаграмма постоянно связана с данными, на основе которых она создана, и обновляется автоматически при изменении исходных данных. Более того, изменение положения или размера элементов данных на диаграмме может привести к изменению данных на листе.

В Excel можно создавать различные диаграммы. Всего существует 11 типов встроенных диаграмм, каждый из которых имеют еще множество разновидностей (видов). Выбор типа диаграммы определяется задачами, решаемыми при ее создании.

Один из типов диаграмм является стандартным, то есть он используется по умолчанию при создании диаграмм. Обычно стандартной диаграммой является плоская гистограмма.

Построение диаграмм

Перед созданием диаграммы следует убедиться, что данные на листе расположены в соответствии с типом диаграммы, который планируется использовать.

Данные должны быть упорядочены по столбцам или строкам. Не обязательно столбцы (строки) данных должны быть смежными, но несмежные ячейки должны образовывать прямоугольник.

При создании *гистограммы***,** *линейчатой диаграммы***,** *графика***,** *диаграммы с областями*, *лепестковой диаграммы*, *круговой диаграммы* **можно использовать от** *одного* **до** *нескольких* **столбцов (строк) данных.**

При создании диаграммы типа "*Поверхность*" должно быть *два* столбца (строки) данных, не считая столбца (строки) подписей категорий.

При создании *круговой диаграммы* *нельзя* использовать *более одного* столбца (строки) данных, не считая столбца (строки) подписей категорий.

Представим графически данные, которые использовались в *первой части работы*.

Упражнение 1.2.1. Постройте диаграмму на основе данных о населении г. Курска представленных на рисунке 1.1.8 (лабораторная работа 1).

Вызовите файл *tab1.xlsx.*

Выделите фрагмент таблицы, для которого создается диаграмма, блок *G16:M17.*

На вкладке *Вставка* в группе *Диаграммы* щелкните по кнопке с нужным типом диаграмм и в галерее выберите конкретный вид диаграммы, *Гистограмма/ Г истограмма с группировкой*, рисунок 1.2.1.

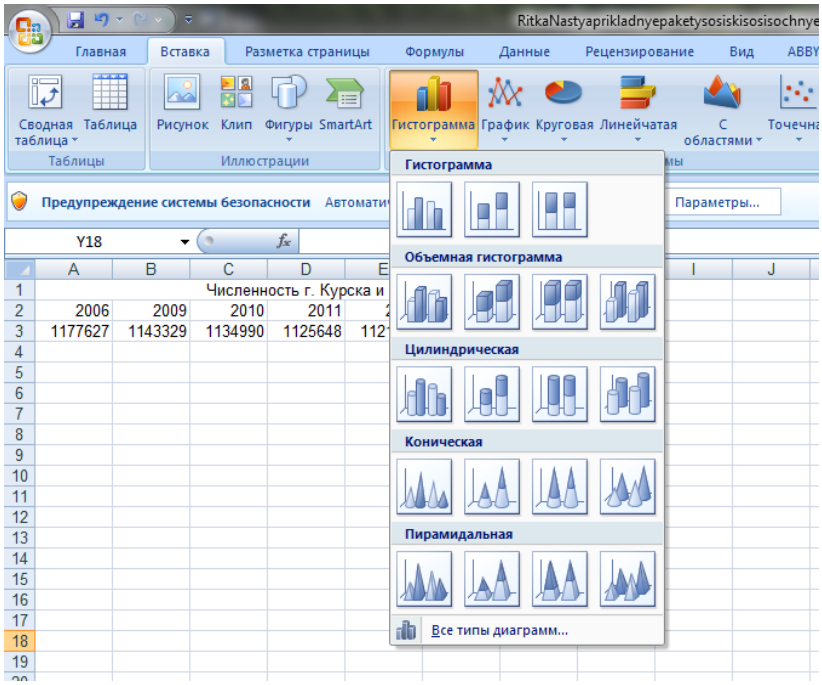


Рисунок 1.2.1 - Выбор вида диаграммы

На листе будет создана диаграмма выбранного вида, рисунок 1.2.2.

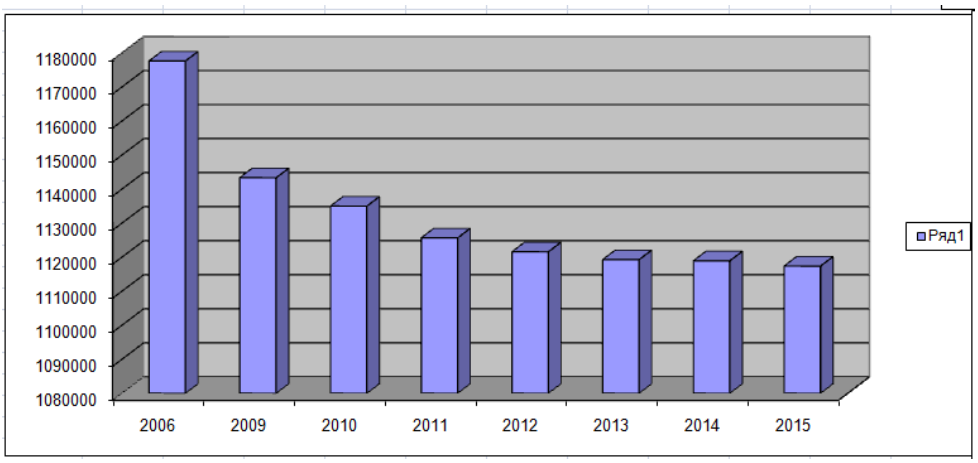


Рисунок 1.2.2 - Созданная диаграмма

Если в группе *Диаграммы* не отображается подходящий тип и вид диаграммы, щелкните по кнопке *Другие диаграммы*.

Для *создания* диаграммы *стандартного* типа достаточно выделить фрагмент листа и нажать клавишу *F11*.

Для *удаления* диаграммы достаточно выделить ее и нажать клавишу *Delete*.

Настройка и редактирование диаграмм

Все действия выполняются с выделенной диаграммой или с ее выделенными элементами.

Упражнение 1.2.2. Оформите диаграмму в соответствии с рисунком 1.2.3 (добавьте заголовок, подписи осей, уберите легенду).

Для *выделения* диаграммы щелкните мышью в любом месте области диаграммы. Признаком выделения являются рамка диаграммы. На рамке имеются маркеры, расположенные по углам и сторонам рамки.

С помощью вкладки *Конструктор* группы *Макеты диаграмм* установите вид диаграммы - *Макет 1*.

Задайте название диаграммы на *Население Новотроицка*, выделив предварительно заголовок диаграммы двойным щелчком мыши.

Уберите легенду диаграммы. Вкладка *Макет*, группа *Подписи*, элемент управления *Легенда*. В раскрывающемся списке выберите *Не добавлять легенду*.

Установите подпись оси с числовыми значениями. Вкладка *Макет*, группа *Подписи*, элемент управления *Названия осей*. В раскрывающемся списке выберите *Название основной вертикальной оси*. Замените появившийся текст на *тыс.чел.*, рисунок 1.2.3.



Рисунок 1.2.3 - Результат редактирования диаграммы

Изменение диаграмм

Если что-то не устраивает в построенной диаграмме, то ее можно изменить.

Изменение размеров диаграммы

Часто бывает весьма затруднительно определить наилучшие размеры внедренной диаграммы до того, как вы увидите представленные на ней данные. Поэтому часто приходится изменять размеры и пропорции внедренной диаграммы для того, чтобы придать ей хороший вид или облегчить ее редактирование.

Упражнение 1.2.3. Увеличьте диаграммы.

Выделите диаграмму, сделав одиночный щелчок по диаграмме. На рамке диаграммы появятся маркеры выделения - маленькие черные квадратики в углах и на серединах сторон рамки.

Захватив курсором мыши рамку, переместите диаграмму.

Установите указатель мыши на маркер справа (указатель мыши при этом изменяет свою форму на двунаправленую стрелку) и увеличьте размеры области построения диаграммы.

«Протаскивание» маркера, расположенного на середине стороны, позволяет изменять вертикальные или горизонтальные размеры диаграммы. «Протаскивание» углового маркера позволяет

изменять вертикальные и горизонтальные размеры диаграммы одновременно.

Изменение типа диаграммы

После того как построена диаграмма Excel предоставляет большие возможности для изменения ее содержимого и вида.

Готовая диаграмма состоит из ряда элементов, которые можно выбирать, щелкнув по ним мышью, и изменять. Выбранный элемент помечается маркерами.

Выделив диаграмму с помощью вкладки *Конструктор*, группы *Тип*, кнопки *Изменить тип диаграммы* можно изменить тип диаграммы.

Упражнение 1.2.4. Измените тип диаграммы.

Выделите диаграмму. Выполнив *Конструктор/Тип/Изменить тип диаграммы*. Установите объемный вид гистограммы, рисунок 1.2.4.

Попробуйте различные типы диаграмм и подберите наиболее наглядный из них.

Сохраните таблицу с диаграммой под именем *tab3. xlsx*.



Рисунок 1.2.4 - Объемная гистограмма

Не все типы диаграмм подходят для наглядного представления данных, а некоторые невозможно построить.

Например, по данным, представленным выше диаграммы типа Точечная или график построить нельзя. Некоторые типы объемных диаграмм тоже могут плохо отразить данные и привести к неразберихе.

Внимание! Если в результате экспериментов диаграмма испорчена, то удалите ее и начните построение сначала. Для удаления следует выделить диаграмму, а затем нажать клавишу *Delete*.

Круговые диаграммы - наиболее часто используемый тип. Они подходят для наглядного представления долей и процентов от целого.

Упражнение 1.2.5. Построить объемную круговую диаграмму, а на основе данных представленных на рисунке 1.2.5, в результате получить диаграмму, представленную на рисунке 1.2.6.

	A	B	C	D	E	F
1	год	всего	психозы и состояния слабс	шизофрения	псих расстр-ва непсихотического хар-ра	умственная отсталость
2	2005	743	290	139	227	226
3	2008	554	219	75	189	146
4	2009	527	208	80	161	158
5	2010	561	239	72	161	161
6	2011	650	335	59	169	146
7	2012	572	294	59	143	135
8	2013	459	249	56	106	104
9	2014	567	257	54	164	146
10						

Рисунок 1.2.5 - Исходные данные

Перейдите на *Лист2* текущей рабочей книги. *Лист2* назовите *Круговая диаграмма.*

Введите заголовок таблицы и данные в соответствии с рисунком 1.2.5.

Постройте диаграмму по шагам:

На 1-м шаге выделите исходные данные *A1:B10*.

На 2-м шаге выполните *Вставка/ Диаграмма/ Круговая/ Объемная резаная круговая***.**

На 3-м шаге укажите месторасположение легенды и добавьте подписи данных - *Макет/ Подписи/ Подписи данных/ У вершины снаружи*.

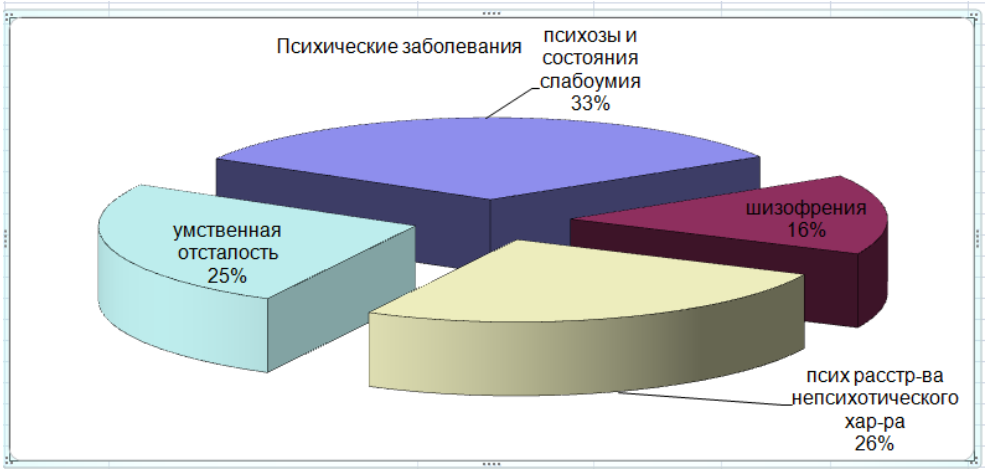


Рисунок 1.2.6 - Результат построения диаграммы

Сохраните диаграмму.

Построение графиков

Для построения обыкновенных графиков функций $y = f(x)$ используется тип диаграммы *точечная*. Этот тип диаграммы требует два ряда значений: X-значения должны быть расположены в левом столбце (или в верхней строке), а Y-значения — в правом (или в нижней строке). На одной диаграмме можно построить несколько графиков функций. Эта возможность используется для проведения сравнительного анализа значений Y при одних и тех же значениях X, а также для графического решения систем уравнений с двумя переменными.

На рисунке 1.2.7 представлен пример построения нескольких графиков функции.

Составьте базу данных принятия лекарственных средств, как указано на рисунке 1.2.7. Постройте графики приема лекарственных средств: а) в течение дня, б) в течение месяца, в) назначенное количество.

	A	B	C	D	E
1	№ пациент	фамилия	назначеное кол-во(шт	кол-во в день(мг	кол-во в месяц(мг)
2	1	Шурыгина	30	5	150
3	2	Сычева	26	2	60
4	3	Соболев	10	5	150
5	4	Хованский	35	3	90
6	5	Ларин	5	1	30
7	6	Миайло	300	2	60
8	7	Серебровский	500	4	120
9	8	Пятницкий	15	2	60
10	9	Букин	30	3	90
11	10	Леонтьев	14	6	180

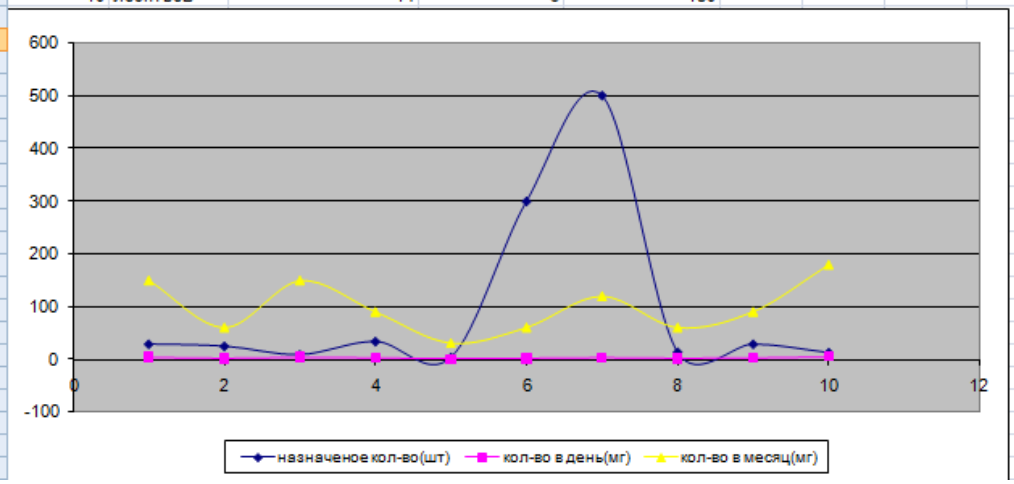


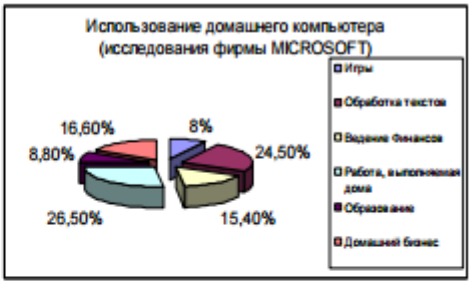
Рисунок 1.2.7 – Пример построения нескольких графиков функций

1.2.3 Задания для самостоятельной работы

Задание 1.2.1. На основе таблицы «Использование домашнего компьютера», рисунок 1.2.8, постройте две диаграммы, представленные на рисунке 1.2.8.

Использование домашнего компьютера
(исследования фирмы MICROSOFT)

Вид работы	%
Игры	8,2
Обработка текстов	24,5
Ведение Финансов	15,4
Работа, выполняемая дома	26,5
Образование	8,8
Домашний бизнес	16,6



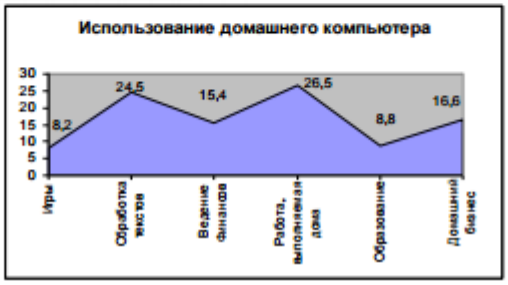


Рисунок 1.2.8 – Задание 1 для самостоятельного выполнения

Примечания:

1. Построение таблицы выполняйте на *Листе4* текущей рабочей книги, дав ему имя *Самостоятельная работа1*;
2. При построении и редактировании обратите внимание на все элементы оформления таблицы и диаграмм. Повторите их. Расположите таблицу и диаграммы на разных листах.

Задание 1.2.2. Выполнить табулирование функций и на основе полученных данных построить совмещенные графики. Для выбора функции используйте таблицу 1.2.1. Номер варианта указывает преподаватель.

Таблица 1.2.1 - Варианты заданий

Вариант	Функция 1	Функция 2	Предел изменения X
1	$Y = -2 / X$	$Y = 4 / X$	$[0,5;5]$ шаг 0,5
2	$Y = X^{-2}$	$Y = X^{-3}$	$[-3;-0]$ шаг 0,5
3	$Y = X^{2}$	$Y = X^{3}$	$[-3;3]$ шаг 0,5
4	$Y = 2^{x}$	$Y = (1/2)^{x}$	$[-2;2]$ шаг 0,5
5	$Y = \ln X$	$Y = \log_{10} X$	$[0,5;1]$ шаг 0,5
6	$Y = X^{-1/2}$	$Y = X^{1/2}$	$[0,5;9]$ шаг 0,5

Примечания:

1. При построении и редактировании обратите внимание на элементы оформления графика.
2. Расположите таблицу с XY-значениями и график на одном *Листе5*.

1.2.4 Контрольные вопросы

1 Перечислите этапы построения диаграммы?

2 Из каких элементов состоит диаграмма и как их добавить (удалить)?

3 Как выполняется изменение типа диаграммы?

4 Как построить совмещенные графики?

5 Приведите примеры задач для построения круговой диаграммы, гистограммы и линейного графика.

Лабораторная работа № 2
Базы данных в пакете Excel

1. Цель работы

Научиться создавать базу данных; выполнять сортировку данных, выборку данных по различным критериям; производить автоматическое подведение итогов в базе данных.

2. Теоретическое введение и упражнения

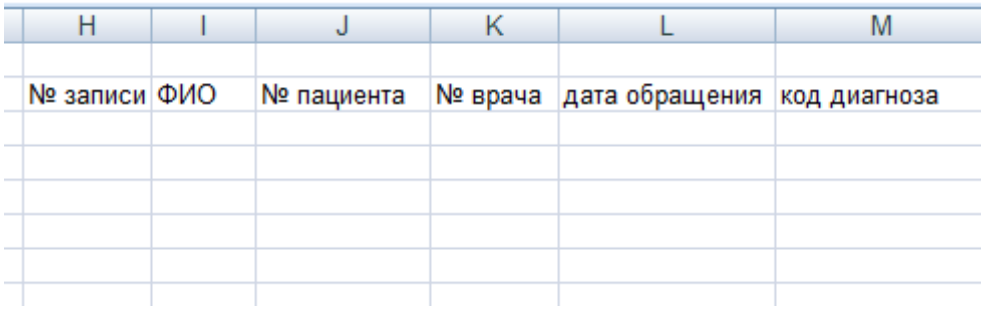
База данных это совокупность определенным образом организованной информации на какую-то тему (в рамках некоторой предметной области). Microsoft Excel содержит широкий набор средств для обеспечения эффективного управления базами данных: создание структуры базы данных; сортировка данных; фильтрация данных; подведение итогов.

Создание структуры базы данных

В Excel базы данных размещаются в таблицах. Каждая таблица состоит из *строк* и *столбцов*, которые в базах данных называются *записями* и *полями*, соответственно. Информация в базах данных имеет постоянную структуру. Каждую строку можно рассматривать как единичную *запись*. Информация в пределах каждой записи содержится в *полях*.

При работе с базами данных в Excel необходимо создать структуру базы данных для этого ввести имена полей. Затем объявить тип и длину каждого поля. И только после этого вводить информацию в базу данных.

Упражнение 2.1**. Создайте базу данных** *Регистратура*.
Назовите *Лист1 БД Регистратуры*, рисунок 2.1.



H	I	J	K	L	M
№ записи	ФИО	№ пациента	№ врача	дата обращения	код диагноза

Рисунок 2.1 - Формирование имен полей базы данных

Введите во второй строке названия полей базы данных: *№ записи, ФИО, № пациента, № врача, дата обращения, код диагноза* (рисунок 2.1).

Отформатируйте поля базы данных в соответствии с рисунком 2.1. Установите ширину полей. В поле *Дата обращения* установите формат ячеек *Дата.* Для этого: выделите ячейки, выполните команды *Главная/ Ячейки/ Формат/ Формат ячеек/ Число/ Дата*, выберите привычный вид даты и нажмите *ОК.*

Заполните базу данных. Данные возьмите из таблицы на рисунке 2.2.

	A	B	C	D	E	F	G
1	№ записи	ФИО	№ пациен	№ врача	дата обращения	код диагноза	
2	526	Шурыгина Н.И.	1283	5	06.03.2017	3657	
3	527	Сычева А.К.	2655	2	14.03.2017	7455	
4	528	Соболев А.Д.	5656	1	06.03.2017	4854	
5	529	Хованский У.М.	4541	4	08.03.2017	87441	
6	530	Ларин И.Н.	1212	3	13.03.2017	45120	
7	531	Миайло Р.Д.Н.	5120	2	14.03.2017	7454	
8	532	Серебровский А,Н.	1582	5	06.03.2017	15452	
9	533	Пятницкий Е.М.	8974	4	08.03.2017	45452	
10	534	Букин Д.В.	1462	1	06.03.2017	1454	
11	535	Леонтьев В.А.	8952	3	13.03.2017	44456	
12							

Рисунок 2.2 - Пример базы данных

Сортировка базы данных

После ввода данных вам может потребоваться упорядочить данные. Процесс упорядочения записей в базе данных называется *сортировкой*. При сортировке изменяется порядок следования записей в базе данных или таблице.

Команда *Данные/ Сортировка и фильтр/ Сортировка* устанавливает порядок записей в базе данных в соответствии с содержимым конкретных столбцов. При выполнении этой команды

открывается окно диалога *Сортировка*, которое позволяет указать поля для сортировки и определить критерии сортировки, рисунок 2.3.



Рисунок 2.3 - Окно сортировки

С помощью раскрывающегося списка *Сортировать по* выбирается столбец для сортировки.

Порядок сортировки устанавливается раскрывающимися списками *Сортировка* (предлагается сортировка по формату) и *Порядок* (определяет как сортировать, в зависимости от типа данных предлагается порядок сортировки).

С помощью кнопки *Параметры...* при сортировке можно учитывать регистр, а также указать, что сортировать строки или столбцы диапазона, рисунок 2.4.

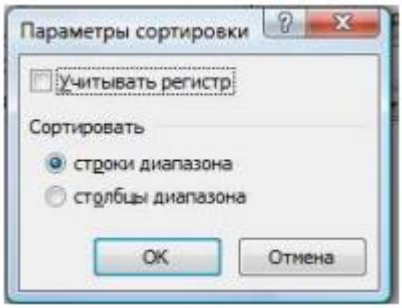


Рисунок 2.4 - Дополнительные параметры сортировки

Упражнение 2.2. Расположите в БД Регистратуре фамилии абонентов в алфавитном порядке.

Установите курсор на ячейку ФИО книги. Выполните *Данные/ Сортировка и фильтр/ Сортировка*.

В раскрывающемся списке *Сортировать по* выберите поле Фамилия, в списке *Порядок* установите *От А до Я*. Нажмите *ОК*. На экране появится база данных после сортировки.

Примечание. Простейшую сортировку по данным одного столбца можно выполнить следующим образом:

1. Выделите одну любую ячейку в столбце, по данным которого сортируется таблица.

2. Нажмите кнопку *Сортировка и фильтр* группы *Редактирование* вкладки *Главная* и выберите направление сортировки.

Предположим необходимо расположить фамилии сбольных по алфавиту, но при наличии одинаковых фамилий необходимо, чтобы имена и отчества также располагались в алфавитном порядке. После выполнения команды *Данные/ Сортировка и фильтр/ Сортировка* в раскрывающемся списке *Сортировать по* выберите поле Фамилия. Добавьте дважды уровень сортировки с помощью кнопки *Добавить уровень* в раскрывающемся списке *Затем по* - Имя, и в последнем уровне в установите поле Отчество. Нажмите *ОК*. На экране появится упорядоченная база данных. Сохраните файл как *tab4.*

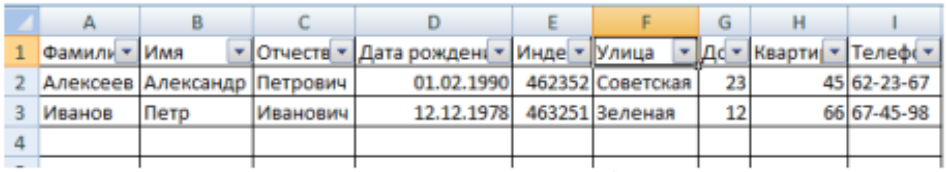
Фильтрация данных

При фильтрации данных находится и отбирается для обработки часть записей в базе данных. В отфильтрованном списке выводятся на экран только те строки, которые содержат определенное значение или отвечают определенным критериям. При этом остальные строки оказываются скрытыми.

Для фильтрации данных используются команда *Фильтр,* группы *Сортировка и фильтр*, вкладки *Данные*, или *Главная/Редактирование/Сортировка и фильтр/ Фильтр,* для организации фильтрации данных на месте их расположения - Автофильтр. А также Расширенный фильтр - элемент управления *Дополнительно*, группы *Сортировка и фильтр*, вкладки *Данные*. В случае простых критериев используется Автофильтр. При использовании сложных критериев применяют Расширенный фильтр.

Автофильтр

Команда *Фильтр*, **группы** *Сортировка и фильтр*, **вкладки** *Данные*, **или** *Главная/ Редактирование/ Сортировка и фильтр/ Фильтр,* помещает кнопки раскрывающихся списков (кнопки со стрелкой) непосредственно в строку с метками столбцов, рисунок 2.5.



	A	B	C	D	E	F	G	H	I
1	Фамили	Имя	Отчеств	Дата рожден	Инде	Улица	Дс	Кварти	Телеф
2	Алексеев	Александр	Петрович	01.02.1990	462352	Советская	23	45	62-23-67
3	Иванов	Петр	Иванович	12.12.1978	463251	Зеленая	12	66	67-45-98
4									

Рисунок 2.5 - Автофильтр

С помощью этих кнопок можно выбрать строки таблицы, которые необходимо вывести на экран. Нажав кнопку со стрелкой, появится раскрывающийся список этого поля, в котором выводятся все значения встречающиеся в столбце. Для выборки данных с использованием фильтра следует выбрать значение или параметр выборки.

На значках раскрывающихся списков в названиях столбцов, выбранных из таблицы с помощью фильтра по которым была произведена выборка, появляется особая отметка. В строке состояния окна Excel в течение некоторого времени отображается текст с указанием количества найденных записей и общего количества записей в таблице, рисунок 2.6.

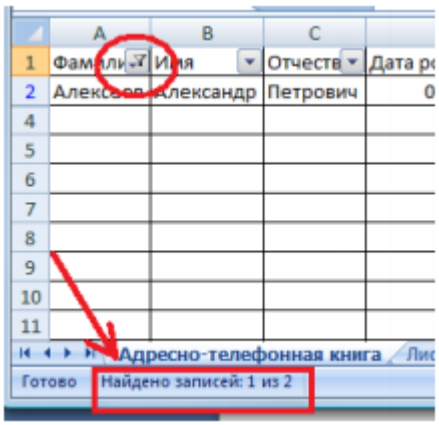


Рисунок 2.6 - Результат работы автофильтра

Можно выбрать строки по *цвету ячейки, цвету текста* или *значку* одного или нескольких столбцов, для этого надо: щелкнуть по значку раскрывающегося списка столбца, по которому производится отбор и выбрать команду *Формат по цвету*, а затем в подчиненном меню выбрать цвет ячейки, цвет шрифта или значок, рисунок 2.7.

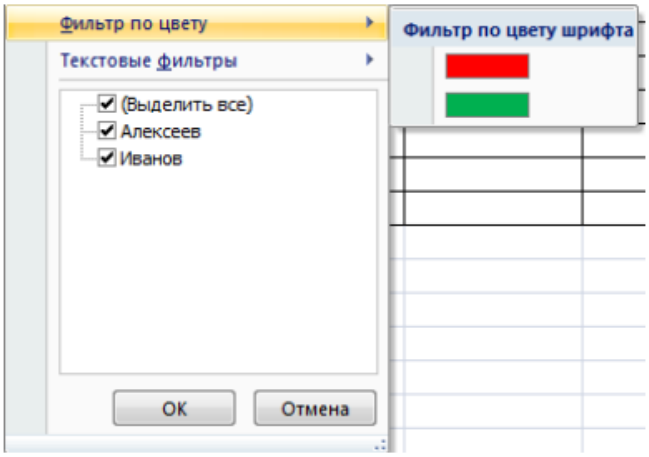


Рисунок 2.7 - Отбор по формату

Можно производить выборку не только по конкретному значению, но и по *условию*. Условие можно применять для *числовых значений, текстовых значений, дат*. Для отбора по условию надо:

1. щелкнуть по значку раскрывающегося списка столбца, по которому производится отбор;
2. выбрать команду *Числовые фильтры* **(**или *Текстовые фильтры* или *Фильтры по дате* - в зависимости от типа фильтруемых данных**)**, а затем в подчиненном меню выбрать применяемое условие, рисунок 2.8.

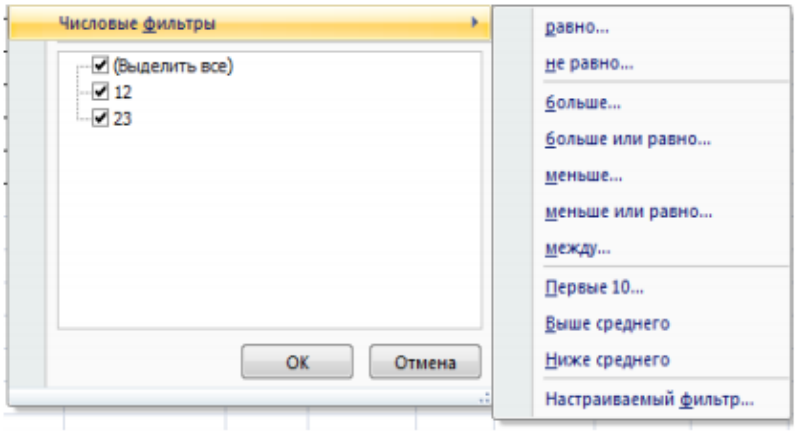


Рисунок 2.8 - Отбор по условию

3. При использовании условий: равно, не равно, больше, меньше, больше или равно, меньше или равно установите значение условия в окне *Пользовательский автофильтр*, рисунок 2.9. Значение условия можно выбрать из списка или ввести с клавиатуры.

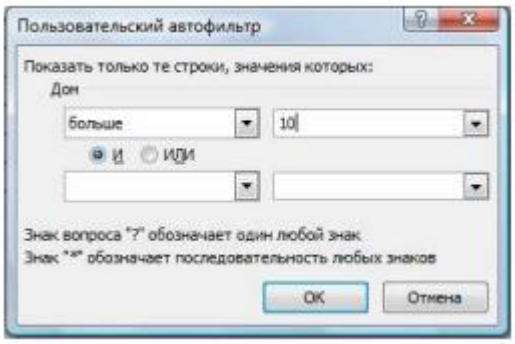


Рисунок 2.9 – Пользовательский фильтр

Для возвращения исходного списка:

1. щелкнуть по значку раскрывающегося списка столбца, по которому производится отбор;

2. выбрать *Снять фильтр с …*

Упражнение 2.3. Создайте базу данных *Телефонный справочник* найдите всех сотрудников с фамилиями Иванов (рисунок 2.10).

Фамилия	Имя	Отчество	Дата рождения	Индекс	Улица	Дом	Квартира	Телефон
Иванов	Петр	Михайлович	12.02.1967	123765	Сове¬ская	123	34	5-67-23
Мазалов	Петр	Александрович	01.09.1984	378653	Зеленая	12	68	3-13-42
Пастухова	Ирина	Юрьевна	05.09.1986	367453	Уральская	4	26	3-19-25
Токарева	Ольга	Александровна	12.03.1986	456359	Сове¬ская	1	8	3-56-98
Черненко	Ирина	Юрьевна	12.01.1988	359456	Зинина	2	10	2-44-02
Мартынюк	Ксения	Андреевна	05.08.1967	359456	Зеленая	4	5	4-67-54
Черненко	Ирина	Юрьевна	08.09.1945	359456	Уральская	34	56	6-70-78
Пастухова	Ирина	Владимировна	06.09.1989	359456	Сове¬ская	156	34	7-67-45
Осипова	Людмила	Григорьевна	04.12.1967	359456	Зинина	4	12	4-60-60
Иванов	Павел	Александрович	20.09.1966	359456	Зеленая	76	5	2-57-17
Ланферова	Елена	Сергеевна	30.11.1934	359456	Уральская	12	7	1-45-67
Далматова	Татьяна	Петровна	05.05.1976	359456	Советская	102	78	7-45-73
Романова	Наталья	Николаевна	08.09.1934	359456	Зинина	2	3	4-67-09
Семенова	Светлана	Даниловна	06.08.1945	359456	Зеленая	45	56	2-56-90
Данилов	Максим	Леонидович	07.08.1988	359456	Уральская	12	89	4-67-92

Рисунок 2.10 – База данных «Телефонный справочник»

На *Листе 2* файла *tab 4.xls* на основе базы данных *Адресно-телефонная книга*, создайте базу данных *Телефонный справочник* дополнив ее полем *Льготы*. Лист2 переименуйте *Телефонный*

справочник. Скопируйте данные с листа *Адресно-телефонная книга.* В поле *Льготы* внесите значения 0 или 1 в зависимости от наличия льгот.

Выделите область с данными. Выполните команду *Фильтр*, группы *Сортировка и фильтр*, вкладки *Данные.* В полях меток столбцов появятся кнопки раскрывающихся списков.

Снимите знак V со всех фамилий за исключением фамилии *Иванов*. Нажмите *ОК*. Нажмите кнопку раскрывающегося списка в поле *Фамилия*. Появится раскрывающийся список, в котором выводятся все значения этого столбца.

Раскрывающийся список исчезнет, а на экране появятся соответствующие строки базы данных, рисунок 2.11.

	A	B	C	D	E	F	G	H	I	J
1										
2	Фамилия	Имя	Отчество	Дата рождения	Индекс	Улица	Дс	Квартира	Телефон	Льготы
3	Иванов	Петр	Михайлович	12.02.1967	123765	Советская	123	34	5-67-23	1
12	Иванов	Павел	Александрович	20.09.1966	359456	Зеленая	76	5	2-57-17	0
18										
19										
20										
21										

Рисунок 2.11 - Результат фильтрации

Верните исходный список базы данных. Для этого в раскрывающемся списке по полю Фамилия выберите команду *Снять фильтр с «Фамилия».*

Упражнение 2.4. Из БД (упражнение 2.3) выведите данные по абонентам, номера телефонов которых начинаются с 64.

Выполните команду *Фильтр*, группы *Сортировка и фильтр*, вкладки *Данные.* В полях меток столбцов появятся кнопки раскрывающихся списков.

Нажмите кнопку для раскрытия списка в поле *Телефон* и выберите *Текстовые фильтры / начинается с ...* Появится окно диалога *Пользовательский автофильтр*, рисунок 2.12.

Нажмите кнопку *ОК*. На экране появится список абонентов, номера телефонов которых начинаются с 64.

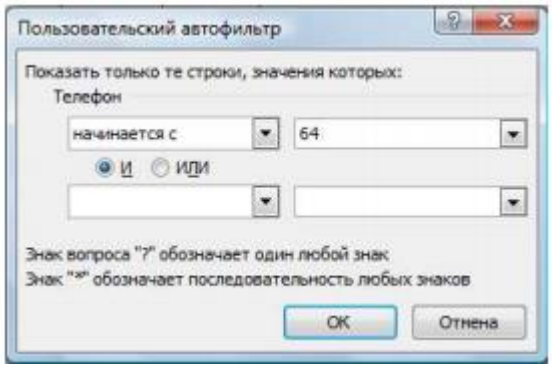


Рисунок 2.12 - Пользовательский фильтр

Расширенный фильтр

Команда *Данные/ Сортировка и фильтр/ Дополнительно* позволяет отыскивать строки с помощью более сложных критериев, чем в пользовательских автофильтрах (2-3 и более условия). При этом Расширенный фильтр использует для фильтрации данных интервал критериев по нескольким полям. На экран выводятся только те строки, которые удовлетворяют всем критериям, записанным в интервале критериев. После выполнения команды на экране появляется окно диалога *Расширенный фильтр*, в котором определяются параметры фильтра, рисунок 2.13.

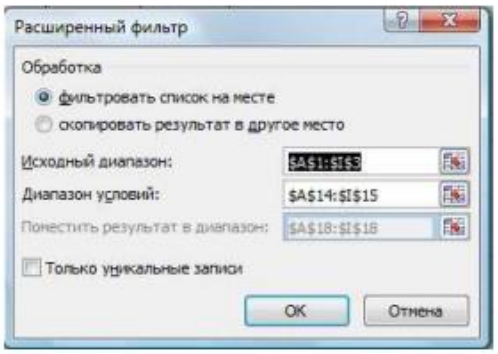


Рисунок 2.13 - Окно расширенного фильтра

Область *Обработка* содержит переключатель, который можно установить в положение *Фильтровать список на месте* или *Скопировать результат на другое место*. При установке переключателя в положение *Фильтровать список на месте* на экране отображаются только те строки, которые удовлетворяют указанному критерию. При выборе *Скопировать результат на*

другое место отфильтрованные данные копируются на другое место на том же рабочем листе.

В поле ввода *Исходный диапазон* указывается интервал, содержащий список, подлежащий фильтрации. В поле ввода *Диапазон критериев* указывается интервал ячеек на рабочем листе, который содержит критерии фильтрации. В поле ввода *Поместить результат в диапазон* указывается интервал ячеек, в который копируются строки, удовлетворяющие критериям. Это поле доступно только в том случае, когда выбран переключатель *Скопировать результат на другое место.* Флажок *Только уникальные записи* предназначен для вывода только неповторяющихся строк.

Упражнение 2.5. Выведите данные по абонентам, имеющим льготы.

Скопируйте ячейки с заголовками базы данных в другое место таблицы (например, ниже базы данных в ячейки *A20: J20*).

Введите в ячейку J20 критерий поиска, в нашем случае наличие льгот - 1, рисунок 2.14.

	A	B	C	D	E	F	G	H	I	J
1										
2	Фамилия	Имя	Отчество	Дата рождения	Индекс	Улица	Дом	Квартира	Телефон	Льготы
3	Иванов	Петр	Михайлович	12.02.1967	123765	Советская	123	34	5-67-23	1
4	Мазалов	Петр	Александрович	01.09.1984	378653	Зеленая	12	68	3-13-42	1
5	Пастухова	Ирина	Юрьевна	05.09.1986	367453	Уральская	4	26	5-19-26	1
6	Токарева	Ольга	Александровна	12.03.1986	496359	Советская	1	8	3-56-98	0
7	Черненко	Ирина	Юрьевна	12.01.1988	359456	Зимняя	2	10	2-44-02	0
8	Мартынюк	Ксения	Андреевна	05.09.1967	359456	Зеленая	4	5	4-67-64	1
9	Черненко	Ирина	Юрьевна	08.09.1945	359456	Уральская	34	56	6-70-78	0
10	Пастухова	Ирина	Владимировна	06.09.1989	359456	Советская	156	34	7-67-45	1
11	Осипова	Людмила	Григорьевна	04.12.1967	359456	Зимняя	4	12	4-60-60	0
12	Иванов	Павел	Александрович	20.09.1966	359456	Зеленая	76	5	2-57-17	0
13	Памфирова	Елена	Сергеевна	30.11.1934	359456	Уральская	12	7	1-45-67	0
14	Далматова	Татьяна	Петровна	06.06.1976	359456	Советская	100	78	7-45-73	1
15	Романова	Наталья	Николаевна	08.09.1934	359456	Зимняя	2	3	4-67-09	1
16	Семенова	Светлана	Денисовна	06.08.1945	359456	Зеленая	45	56	2-56-90	0
17	Данилов	Максим	Леонидович	07.08.1988	359456	Уральская	12	89	4-67-92	1
18										
19										
20	Фамилия	Имя	Отчество	Дата рождения	Индекс	Улица	Дом	Квартира	Телефон	Льготы
21										1
22										

Рисунок 2.14 - Подготовка к расширенному фильтру

Установите курсор в произвольную ячейку базы данных и выполните команду *Данные/Сортировка и фильтр/ Дополнительно*. Появится окно диалога *Расширенный фильтр*.

Установите переключатель *Скопировать результат на другое место***. Становится доступным поле ввода** *Поместить результат в диапазон***.**

Нажмите мышью на поле ввода *Исходный диапазон* и выделите список, ячейки *A2:J17*. При этом в поле ввода появятся адреса выделенных ячеек, рисунок 2.15.

Перейдите в поле ввода *Диапазон критериев* и выделите ячейки, где указаны критерии поиска (*не забудьте выделить названия столбцов в диапазоне критериев*), *A20:J21*. При этом в поле ввода появятся адреса выделенных ячеек.

Перейдите в поле ввода *Поместить результат в диапазон* и укажите ячейку, начиная с которой будет выводиться отфильтрованный список, *A23*. При этом в поле ввода появятся адреса выделенных ячеек. Нажмите *ОК*. На экране появятся данные по абонентам, имеющим льготы.

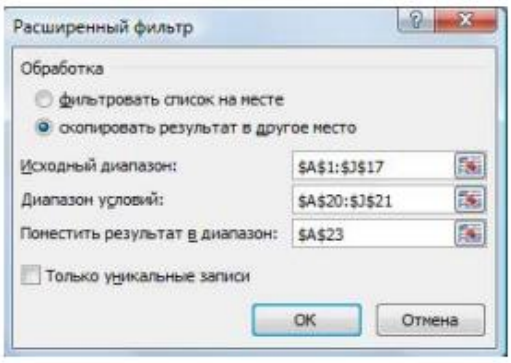


Рисунок 2.15 - Значения для фильтрации

Упражнение 2.6. Выведите данные по абонентам, не имеющим льгот, номера телефонов которых начинаются с 4.

Скопируйте ячейки с заголовками базы данных в другое место таблицы K2:S2

Введите в ячейку *S3* критерий поиска, в данном случае символы *4**, а в ячейку *T3 - 0*.

Примечание: символ ***** означает, что за цифрой 4 могут стоять любые символы.

Выполните команду *Данные/Сортировка и фильтр/ Дополнительно***. Появится окно диалога** *Расширенный фильтр***.**

Установите переключатель *Скопировать результат на другое место***.**

Нажмите мышью на поле ввода *Исходный диапазон*. Выделите список, ячейки *A2- J19*.

Нажмите мышью на поле ввода *Диапазон критериев.* Выделите ячейки, где указаны критерии поиска (не забудьте выделить названия столбцов), *К2:Т3*.

Нажмите мышью на поле ввода *Поместить результат в диапазон* и укажите ячейку, начиная с которой будет выводиться отфильтрованный список, *К7*.

Подведение итогов в базе данных

Один из способов обработки данных состоит в *подведении промежуточных* и *конечных итогов*. С помощью команды *Данные/ Структура/ Промежуточные итоги* можно вставить строки итогов в список, произведя суммирование данных. При вставке строк Excel автоматически помещает в конец списка данных строку общих итогов.

После выполнения команды *Данные/Структура/ Промежуточные итоги* появится окно диалога *Промежуточные итоги*, рисунок 2.16.

В поле ввода *При каждом изменении в* определяется столбец, который содержит группы, для которых нужно вычислять промежуточные итоги. Надо выбрать в раскрывающемся списке название нужного столбца для сортировки списка.

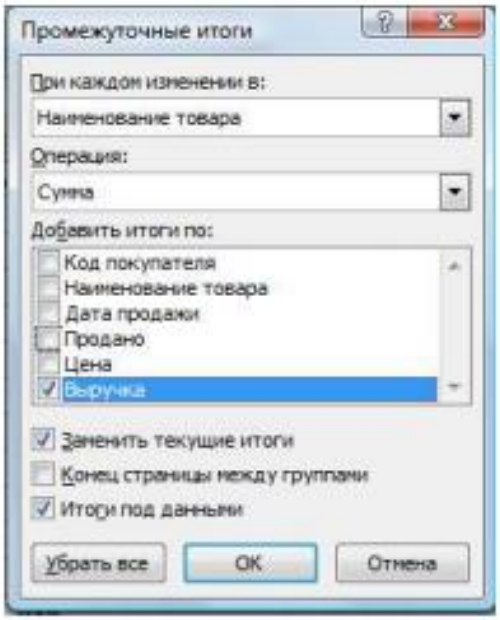


Рисунок 2.16 - Окно Промежуточные итоги

В раскрывающемся списке *Операция* выбирать функцию, которая вычисляет итоги. По умолчанию Excel использует функцию *Сумма* для числовых значений или функцию *Счет* для текстовых значений. Кроме того можно использовать следующие функции: *Среднее, Максимум, Минимум, Произведение, Кол-во чисел, Несмещенное отклонение, Смещенное отклонение, Несмещенная Смещенная дисперсия.*

Вычисленные итоги располагаются под столбцом, который выделяется в поле ввода *Добавить итоги по*. Можно выделить более одного столбца, поставив напротив названия V.

Если установлен флажок *Заменить текущие итоги*, то все итоги в списке заменяются новыми итогами выбранными вами.

Флажок *Конец страницы между группами* автоматически вставляет конец страницы перед каждой группой данных, для которой вычисляются итоги.

Если установлен флажок *Итоги по данным*, то строки итогов и общих итогов помещаются под соответствующими данными. Снимите этот флажок, если строки итогов надо вставить над данными.

При нажатии на кнопку *Убрать все* удаляются все итоги из текущего списка и список приобретает исходный вид.

Упражнение 2.7. Используя базу данных *Товар*, рисунок 2.17, вычислите сумму денег, потраченную каждым покупателем при приобретении товаров.

Перейдите на *Лист3*, назовите его *Товар* и создайте базу данных, рисунок 2.17. Поле *Выручка* формируется как произведение цены единицы товара и количества проданного товара.

Отсортируйте список по полю *Код покупателя*. Выполните для этого команду *Данные/Сортировка и фильтр/ Сортировка*. В раскрывающемся списке поля ввода *Сортировать по* выберите поле *Код покупателя* и нажмите кнопку *ОК*.

Выполните команду *Данные/ Структура/ Промежуточные итоги***. Появится окно диалога** *Промежуточные итоги***.**

В поле *При каждом изменении в* нажмите мышью на кнопке со стрелкой. Раскроется список полей базы данных. Выберите поле *Код покупателя*.

В поле *Операция* нажмите мышью на кнопке со стрелкой. В раскрывшемся списке функций выберите функцию *Сумма*.

	A	B	C	D	E	F
1	Код покупателя	Наименование товара	Дата продажи	Продано	Цена	Выручка
2	1	Молоко	12.03.2010	3	20	60
3	1	Сельдь	12.03.2010	1	57	57
4	1	Хлеб	12.03.2010	5	12	60
5	3	Молоко	20.03.2010	2	20	40
6	4	Сметана	20.03.2010	1	45	45
7	5	Хлеб	02.03.2010	3	12	36
8	6	Конфеты	02.03.2010	1	100	100
9	6	Сельдь	03.03.2010	2	57	114
10	8	Конфеты	20.03.2010	1	100	100
11	8	Молоко	12.03.2010	4	20	80
12	11	Кефир	04.02.2010	2	16	32
13	11	Молоко	02.03.2010	5	20	100
14	11	Хлеб	20.03.2010	3	12	36
15	12	Батон	05.02.2010	1	18	18
16	12	Кефир	04.02.2010	1	16	16
17	14	Конфеты	05.02.2010	2	100	200
18	14	Хлеб	05.02.2010	3	12	36
19						

Рисунок 2.17 - Исходные данные таблицы Товар

В поле *Добавить итоги по* поставьте галочку на поле *Выручка*.

Установите флажки *Заменить текущие итоги* и *Итоги под данными*. Нажмите кнопку *ОК*. Итоги подведены и результат представлен на рисунке 2.18. Для каждого покупателя подсчитано, сколько он потратил денег на покупки.

	A	B	C	D	E	F
1	Код покупателя	Наименование товара	Дата продажи	Продано	Цена	Выручка
2	1	Молоко	12.03.2010	3	20	60
3	1	Сельдь	12.03.2010	1	57	57
4	1	Хлеб	12.03.2010	5	12	60
5	**1 Итог**					177
6	3	Молоко	20.03.2010	2	20	40
7	**3 Итог**					40
8	4	Сметана	20.03.2010	1	45	45
9	**4 Итог**					45
10	5	Хлеб	02.03.2010	3	12	36
11	**5 Итог**					36
12	6	Конфеты	02.03.2010	1	100	100
13	6	Сельдь	03.03.2010	2	57	114
14	**6 Итог**					214
15	8	Конфеты	20.03.2010	1	100	100
16	8	Молоко	12.03.2010	4	20	80
17	**8 Итог**					180
18	11	Кефир	04.02.2010	2	16	32
19	11	Молоко	02.03.2010	5	20	100
20	11	Хлеб	20.03.2010	3	12	36
21	**11 Итог**					168
22	12	Батон	05.02.2010	1	18	18
23	12	Кефир	04.02.2010	1	16	16
24	**12 Итог**					34
25	14	Конфеты	05.02.2010	2	100	200
26	14	Хлеб	05.02.2010	3	12	36
27	**14 Итог**					236
28	**Общий итог**					1130

Рисунок 2.18 - Результат промежуточных итогов

Обратите внимание на маленькие "+" и , расположенные слева от таблицы и числа, расположенные над ними, указывающие на уровень Итогов. Символ скрытия деталей (кнопка со знаком "-") Указывает итоговую строку или столбец с выведенными детальными строками или столбцами. Нажмите мышью на эту кнопку для скрытия детальных данных.

Символ показа деталей (кнопка со знаком "+") указывает итоговую строку или столбец со скрытыми деталями. Для показа деталей нажмите мышью на эту кнопку.

Упражнение 2.8. Подсчитайте количество продано товара одного наименования и выручку по каждому наименованию товара.

Выполните команду *Данные/ Сортировка и фильтр/ Сортировка***. В раскрывающемся списке** *Сортировать по* **выберите поле** *Код товара***.**

Выполните команду *Данные/ Структура/ Промежуточные итоги*, со следующими параметрами:

Промежуточные итоги - Код товара
Операция - Сумма
Добавить итоги по - Продано, Выручка

Нажмите кнопку *ОК*.

Выполните сохранение файла как *tab4*.

Функция промежуточные итоги

Функция *ПРОМЕЖУТОЧНЫЕ ИТОГИ (номер_функ-ции; ссылка;* [ссылки]), где *номер_функции* определяет, какие именно итоги нужно подводить (1 - СРЗНАЧ, 2 - СЧЕТ, 3 - СЧЕТЗ, 4 - . _ МАКС, 5 - МИН, 9 - СУММ и т.д.), а список *ссылок* указывает, над какими ячейками нужно производить вычисления. Бывает удобно использовать эту функцию при промежуточном анализе данных в таблице, например при использовании автофильтра, так как она будет производить соответствующие вычисления только над отфильтрованными данными.

Например, в ячейку *F22* запишите формулу *=СУММ(E2:E20)*, а в ячейку *F23* запишите формулу *=ПРОМЕЖУТОЧНЫЕ ИТОГИ*

(9;F2:F20) и примените *Автофильтр* к полю *Код покупателя*, выбрав покупателя *1*. Как изменился результат в ячейке *F23*?

Если фильтрация не применяется, значения в этих ячейках совпадают, а при применении автофильтра значение в ячейке F21 существенно меньше, так как в вычислениях участвуют только отфильтрованные записи.

3. Задания для самостоятельной работы

Задание 1. В базе данных *Адресно-телефонная книга* с помощью вышеописанных команд найти:

1) всех абонентов, фамилии которых начинаются с буквы «С».
2) просмотреть данные об абонентах, родившихся после 1970 года, фамилии которых начинаются на букву «С».
3) записи, содержащие информацию об абонентах, номер телефона которых, начинается на 2 и фамилия оканчивается на «ов»;
4) записи, в которых, содержится информация об абонентах, проживающих по улице Советская, с номером дома больше 100.

Задание 2. **В базе данных** *Товар***:**

1) с помощью расширенного фильтра отобразить записи, относящиеся к покупкам в заданном диапазоне дат, произведенным в выходные дни;
2) используя функцию *ПРОМЕЖУТОЧНЫЕ ИТОГИ()* ,

получить максимальную и суммарную стоимость всех покупок. Оценить, как изменится результат при применении к базе данных *Автофильтра*;

3) определить какое количество товара приобрел каждый покупатель;
4) определить какова выручка магазина за каждый день.
5)

4. Контрольные вопросы

1. Как формируется структура базы данных?

2. Как выполнить сортировку данных? Приведите пример.
3. Для чего используются команды Фильтр и Дополнительный фильтр? В чем их отличие?
4. Как подвести промежуточные и конечные итоги в базах данных?
5. Как работает функция ПРОМЕЖУТОЧНЫЕ ИТОГИ(), для чего она предназначена?

Задания для выполнения индивидуальных работ

Задание 1. Вычислить значения функции F(x) для всех x на заданном интервале с заданным шагом. Исходные данные должны располагаться во вспомогательной таблице 1.

Таблица 1 - Шаблон вспомогательной таблицы

a	b	c	x	шаг

Наличие столбцов a, b, c определяется условием задачи.

Решение должно быть получено в виде таблицы 2 содержащей значение аргумента и соответствующие значения табулируемой функции.

Таблица 2 - Шаблон таблицы результатов

X	F(x)

Построить с помощью мастера диаграмм график заданной функции.

Вид функции, коэффициенты, интервал табулирования и значение шага определяются по номеру варианта, таблица 3. Вариант определяется по номеру в списке группы.

Таблица 3 - Варианты заданий

№	Функция	Исходные данные	Диапазон и шаг изменения аргумента
1	2	3	4
1.	$y = at^2 \ln t$	$a = -0,2$	$t \in [0,1;3] \Delta t = 0,15$
2.	$y = ax^3 + 7\sqrt{x}$	$a = 1,5$	$x \in [0,8;2] \Delta x = 0,1$
3.	$w = ax^2 + bx + c$	$a = 2,8$ $b = -0,3$ $c = 4$	$x \in [1;2] \Delta x = 0,05$
4.	$Q = \lg(x + 7\sqrt{\lvert x + a)\rvert}$	$a = 1,65$	$x \in [0,7;2] \Delta x = 0,1$
5.	$y = 1,5\cos^2 x$	-	$x \in [0,2;2,8] \Delta x = 0,2$
6.	$w = x\sqrt[3]{a - x}$	$a = 2,5$	$x \in [1;5] \Delta x = 0,5$
7.	$Q = bx - \lg bx$	$b = 1,5$	$x \in [0,1;1] \Delta x = 0,1$
8.	$y = \sin x \lg x$	-	$x \in [2;5] \Delta x = 0,25$
9.	$f = \sin^2 \sqrt{\lvert ax\rvert}$	$a = 20,3$	$x \in [0,5;2] \Delta x = 0,1$
10.	$z = \sqrt{x + t} + 1/x$	$t = 2,2$	$x \in [0,2;2] \Delta x = 0,2$
11.	$s = \dfrac{a + b}{e^x + \cos x}$	$a = 2,6$ $b = -0,39$	$x \in [0;7] \Delta x = 0,5$
12.	$y = 2a\cos x + 3x^2$	$a = 0,9$	$x \in [0,8;2] \Delta x = 0,1$
13.	$w = a/i + bi^2 + c$	$a = 2,1$ $b = 1,8$ $c = -20,5$	$i \in [0;12] \Delta i = 1$
14.	$z = a\sin\left(\dfrac{i^2 + 1}{n}\right)$	$a = 0,3$ $n = 10$	$i \in [1;10] \Delta i = 1$
15.	$w = \sqrt{at^2 + b\sin t + 1}$	$a = 2,5$ $b = 0,4$	$t \in [-1;1] \Delta t = 0,2$
16.	$Q = \pi x^2 - 7/x^3$	-	$x \in [0,5;2] \Delta x = 0,1$
17.	$y = \cos x \lg x$	-	$x \in [2;6] \Delta x = 0,25$
18.	$y = a\ln x + \sqrt[5]{\lvert x\rvert}$	$a = 1,8$	$x \in [0,5;2] \Delta x = 0,1$

Продолжение таблицы 3

1	2	3	4
19.	$w = ai + bi^2$	$a = 2,5$ $b = 1,7$	$x \in [0,5;2] \Delta x = 0,1$
20.	$s = e^x + \cos x$	-	$x \in [0;8] \Delta x = 0,5$
21.	$z = \sqrt{x + t} + 1$	$t = 2,2$	$x \in [0,2;2] \Delta x = 0,2$
22.	$s = (a + b)/(x + 1)$	$a = 2,6$ $b - 0,39$	$x \in [0;7] \Delta x = 0,5$
23.	$W = \sqrt{\sqrt{at^2} + b\cos t}$	$a = 2,5$ $b = 0,4$	$t \in [-1;1] \Delta t = 0,2$
24.	$Q = ax + 8\sqrt{x}$	$a = 1,55$	$x \in [0,5;2] \Delta x = 0,1$
25.	$w = a / \sqrt{i} + bi^2$	$a = 2,1$ $b = 1,8$	$i \in [1;12] \Delta i = 1$

Задание 2. Составить накладную, рисунок 1, которая:

1) на основе цены одной единицы товара и количества товара считает общую стоимость, представляя отдельно рубли и копейки;

2) найденное значение представляет в виде текста, рубли - числом, а копейки прописью;

3) данные вводимые в поля «От кого» и «Кому» преобразует в формате И.О.Фамилия в полях «Сдал» и «Принял» .

Примечания:

1. При подсчете суммы в рублях необходимо учитывать накопившиеся копейки, а при подсчете суммы в копейках необходимо учитывать, что 100 коп.=1 руб. Поэтому сумму в копейках нужно поделить на 100, остаток от деления с помощью функции *ОСТАТ()* поместить в графу «коп.», а целую чисть, выделяемую с помощью функции *ЦЕЛОЕ(),* прибавить к сумме в рублях.

2. При составлении итоговой формулировки числовое значение необходимо переместив текстовое с помощью функции *ТЕКСТ()***.**

3. Составление строки расшифровки подписи выполняется с помощью функций *СЦЕПИТЬ(), ЛЕВСИМВ(), ПСТР(), ПО-*

ИСК(): функция *ПОИСК()* разыскивает пробелы, ПСТР() выделяет инициалы, *ЛЕВСИМВ()* выделяет фамилию.

4. Чтобы написать прописью количество копеек, нужно, составив два списка числительных (первый - от 1 до 19 (прописью), второй - десятки от 20 до 90 (тоже прописью), выбирать нужные числительные из этих списков с помощью функции *ИНДЕКС()*. Номер строки поиска задается количеством копеек, если их не более 19, или отдельно первой и второй цифрой из этого количества для получения соответственно десятков и единиц.

«___»_____20__г.						
НАКЛАДНАЯ №_____						
От кого	Филиппов Кирилл Александрович					
Кому	Васильев Андрей Анатольевич					
№ п/п	Наименование	Кол-во	Цена		Сумма	
			руб.	коп.	руб.	коп.
1	Карандаш НВ	142	1	7	151	94
2	Карандаш цв.	216	1	37	295	92
3	Ручка шар.	180	2	64	475	20
4	Линейка	96	3	42	328	32
5	Ручка гель	220	5	29	1163	80
Итого:					2415	18
Итого на сумму 2415 руб. восемнадцать коп.						
Сдал				К.А. Филиппов		
Принял				А.А. Васильев		

Рисунок 1 - Образец накладной

Лабораторная работа №3

3.1 Основы работы в пакете MathCad

Пользовательский интерфейс системы создан так, что пользователь, имеющий элементарные навыки работы с Windows-приложениями, может сразу начать работать с MathCad.

Под интерфейсом понимается не только легкое управление системой, как с клавишного пульта, так и с помощью мыши, но и просто набор необходимых символов, формул, текстовых комментариев с последующим запуском документов (Worksheets) в реальном времени.

Запустив систему MathCad из Windows, вы увидите на экране диалоговое окно, первоначально пустое (рисунок 3.1.1).

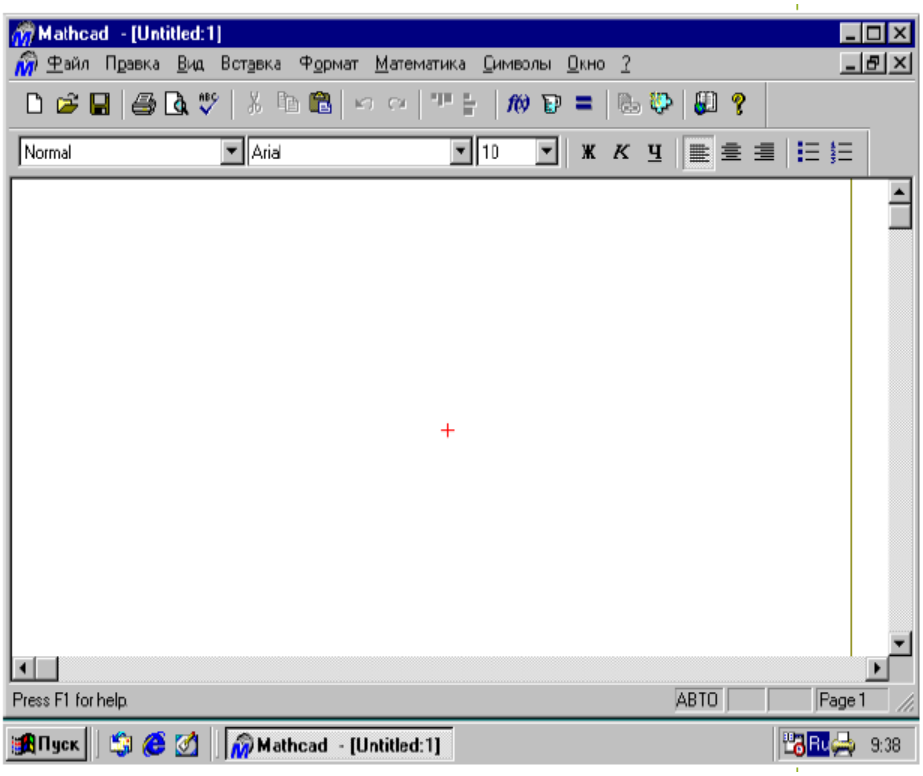


Рисунок 3.1.1 – Диалоговое окно

Над ним видна строка с основными элементами интерфейса. Опции главного меню, содержащиеся в этой строке, легко изучит самостоятельно; некоторые из них очень похожи на стандартные опции, принятые в текстовых редакторах Windows.

MathCad работает с документами. С точки зрения пользователя, документ – это чистый лист бумаги, на котором можно размещать блоки трех основных типов: математические выражения, текстовые фрагменты и графические области.

Расположение нетекстовых блоков в документе имеет принципиальное значение - слева направо и сверху вниз.

Работа с документами MathCad не требуют обязательного использования возможностей главного меню, так как основные из них дублируются кнопками быстрого управления, которые расположены в удобных перемещаемых с помощью мыши наборных панелях – палитрах. Наборные панели появляются в окне редактирования документов при активизации кнопок – пиктограмм. Они служат для вывода заготовок – шаблонов математических знаков (цифр, знаков арифметических операций, матриц, знаков интеграла, производных, приделов и др.). Указатель мыши подводим к “Вид” в главном меню, щелкаем левой кнопкой мыши; указатель подводим к “Панели инструментов” и щелкаем левой кнопкой мыши; Выпадает следующее меню. Указатель мыши подводим к “Математика” и щелкаем левой кнопкой мыши. Выпадают наборные панели (рисунок 3.1.2)

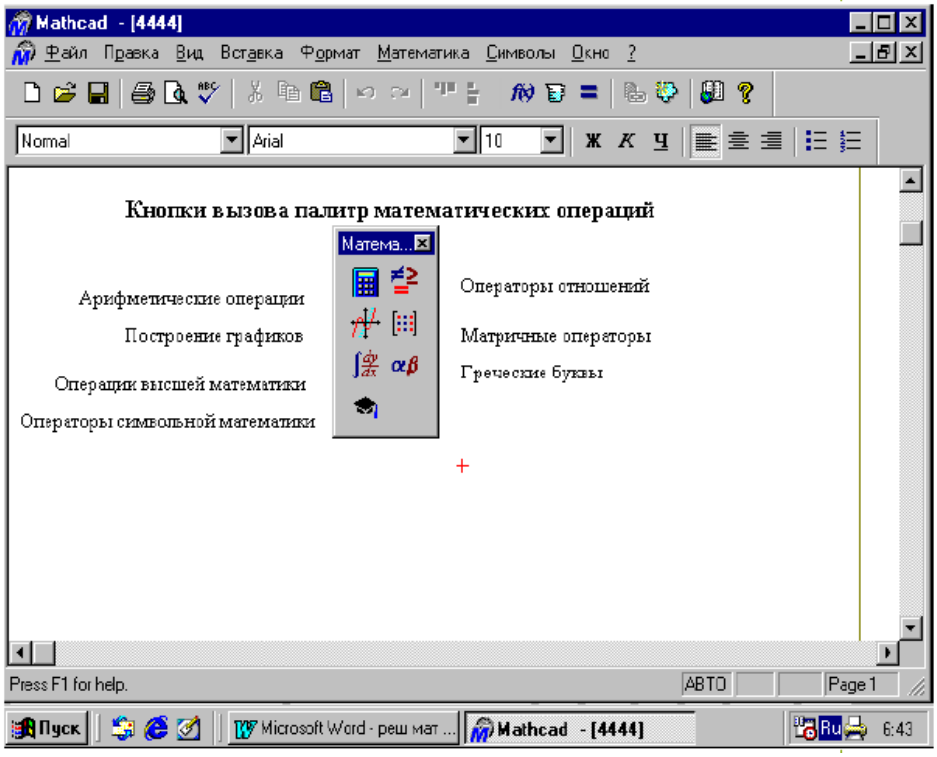


Рисунок 3.1.2 – Наборные панели

Математические выражения

К основным элементам математических выражений MathCad относятся типы данных, операторы, функции и управляющие структуры.

Операторы

Операторы - элементы MathCad, с помощью которых можно создавать математические выражения. К ним, например, относятся символы арифметических операций, знаки вычисления сумм, произведений, производной и интеграла и т.д.

Оператор определяет:

- действие, которое должно выполняться при наличии тех или иных значений операндов;
- сколько, где и какие операнды должны быть введены в оператор.

Операнд - число или выражение, на которое действует оператор. Например, в выражении 5!+3 число 3 и выражение 5! - операнды оператора + (плюс), а число 5 операнд оператора факториал (!). После указания операндов операторы становятся исполняемыми по документу блоками.

Типы данных

К типам данных относятся числовые константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

Константами называют поименованные объекты, хранящие некоторые значения, которые не могут быть изменены.

Переменные являются поименованными объектами, имеющими некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д. Имена констант, переменных и иных объектов называют идентификаторами. Идентификаторы в MathCad представляют собой набор латинских или греческих букв и цифр.

В MathCad содержится небольшая группа особых объектов, которые нельзя отнести ни к классу констант, ни к классу переменных, значения которых определены сразу после запуска программы. Их правильнее считать системными переменными, имеющими предопределенные системой начальные значения. Изменение значений системных переменных производят во вкладке **Встроенные переменные** диалогового окна **Math Options** команды **Математика → Опции.**

Обычные переменные отличаются от системных тем, что они должны быть предварительно определены пользователем, т. е. им необходимо хотя бы однажды присвоить значение. В качестве оператора присваивания используется знак **:=**, тогда как знак = отведен для вывода значения константы или переменной.

Если переменной присваивается начальное значение с помощью оператора **:=**, вызывается нажатием клавиши **:** (двоеточие) на клавиатуре, такое присваивание называется локальным. До этого присваивания переменная не определена и ее нельзя использовать. Однако с помощью знака ≡ (клавиша ~ на клавиатуре) можно обеспечить глобальное присваивание (рисунок 3.1.3).

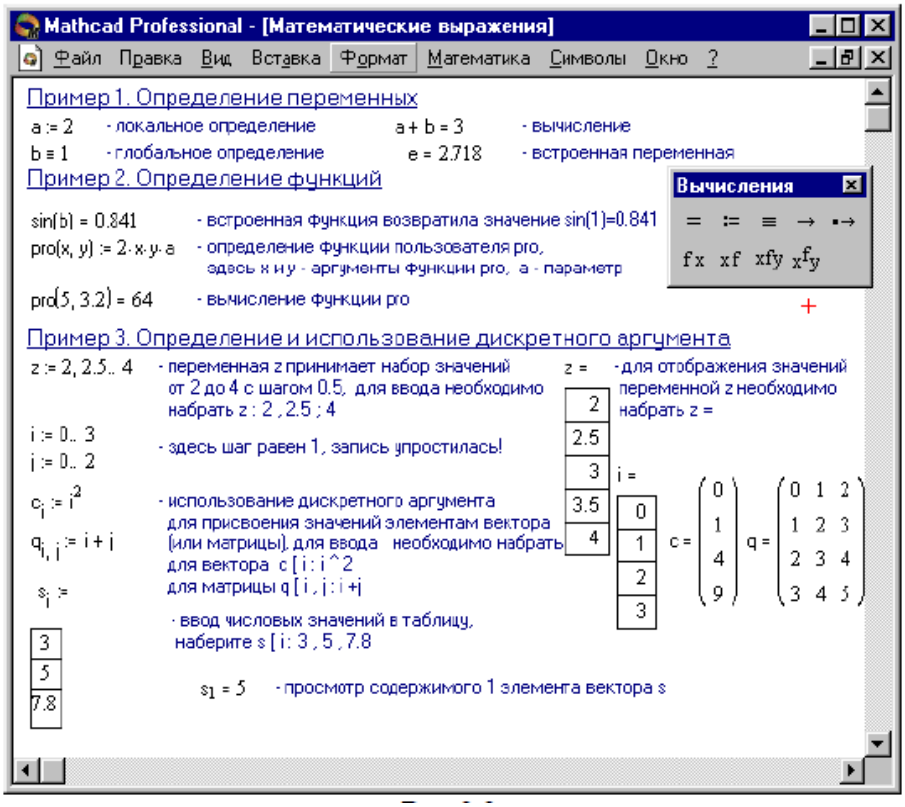


Рисунок 3.1.3 – Обеспечение глобального присваивания

MathCad прочитывает весь документ дважды слева направо и сверху вниз. При первом проходе выполняются все действия, предписанные локальным оператором присваивания (≡), а при втором - производятся действия, предписанные локальным оператором присваивания (**:=**), и отображаются все необходимые результаты вычислений (=).

Существуют также жирный знак равенства ⩵ (комбинация клавиш **Ctrl +** =), который используется, например, как оператор приближенного равенства при решении систем уравнений, и символьный знак равенства ÷ (комбинация клавиш **Ctrl + .**).

Дискретные аргументы

Дискретные аргументы - особый класс переменных, который в пакете MathCad зачастую заменяет управляющие структуры, называемые циклами (однако полноценной такая замена не является). Эти переменные имеют ряд фиксированных значений, либо целочисленных (1 способ), либо в виде чисел с определенным шагом, меняющихся от начального значения до конечного (2 способ).

1. *Name* := *Nbegin* .. *Nend*, где *Name* - имя переменной, *Nbegin* - ее начальное значение, *Nend* - конечное значение, **..** - символ, указывающий на изменение переменной в заданных пределах (вводится клавишей **;**). Если *Nbegin* < *Nend*, то шаг переменной будет равен +1, иначе -1.

2. *Name* := *Nbegin*, (*Nbegin* + *Step*) .. *Nend*

Здесь *Step* - заданный шаг изменения переменной (он должен быть положительным, если *Nbegin* < *Nend*, или отрицательным в обратном случае).

Дискретные аргументы значительно расширяют возможности MathCad, позволяя выполнять многократные вычисления или циклы с повторяющимися вычислениями, формировать векторы и матрицы (рисунок 3.1.3, пример 3).

Массив

Массив - имеющая уникальное имя совокупность конечного числа числовых или символьных элементов, упорядоченных некоторым образом и имеющих определенные адреса. В пакете MathCad используются массивы двух наиболее распространенных типов:

- одномерные (векторы);
- двумерные (матрицы).

Порядковый номер элемента, который является его адресом, называется индексом. Индексы могут иметь только целочисленные значения. Они могут начинаться с нуля или единицы, в соответствии со значением системной переменной **ORIGIN.**

Векторы и матрицы можно задавать различными способами:

- с помощью команды **Вставка → Матрица**, или комбинации клавиш **Ctrl + M**, или щелчком на кнопке  панели **Матрица,** заполнив массив пустых полей для не слишком больших массивов;
- с использованием дискретного аргумента, когда имеется некоторая явная зависимость для вычисления элементов через их индексы (рисунок 3.1.3, пример 3).

Функции

Функция - выражение, согласно которому проводятся некоторые вычисления с аргументами и определяется его числовое значение.

Следует особо отметить разницу между аргументами и параметрами функции. Переменные, указанные в скобках после имени функции, являются ее аргументами и заменяются при вычислении функции значениями из скобок. Переменные в правой части определения функции, не указанные скобках в левой части, являются параметрами и должны задаваться до определения функции.

Главным признаком функции является возврат значения, т.е. функция в ответ на обращение к ней по имени с указанием ее аргументов должна возвратить свое значение. Функции в пакете

MathCad могут быть встроенные, т. е. заблаговременно введенные разработчиками, и определенные пользователем.

Способы вставки встроенной функции:

- Выбрать пункт меню **Вставка → Функция.**
- Нажать комбинацию клавиш **Ctrl + E.**
- Щелкнуть на кнопке  .

Текстовые фрагменты

Текстовые фрагменты представляют собой куски текста, которые пользователь хотел бы видеть в своем документе. Существуют два вида текстовых фрагментов:

- текстовая область предназначена для небольших кусков текста - подписей, комментариев и т. п. Вставляется с помощью команды **Вставка → Текстовая регион** или комбинации клавиш **Shift + "** (двойная кавычка);
- текстовый абзац применяется в том случае, если необходимо работать с абзацами или страницами. Вставляется с помощью комбинации клавиш **Shift + Enter.**

Графические области

Графические области делятся на три основных типа - двумерные графики, трехмерные графики и импортированные графические образы. Двумерные и трехмерные графики строятся самим MathCad на основании обработанных данных.

Для создания декартового графика:

- Установить визир в пустом месте рабочего документа.
- Выбрать команду **Вставка → График - X-Y график,** или нажать комбинацию клавиш **Shift + @,** или щелкнуть кнопку  панели **Графики**. Появится шаблон декартового графика.
- Введите в средней метке под осью X первую независимую переменную, через запятую - вторую и так до 10, например x_1, x_2 …

- Введите в средней метке слева от вертикальной оси Y первую независимую переменную, через запятую - вторую и т. д., например $y_1(x_1)$, $y_2(x_2)$ …, или соответствующие выражения.
- Щелкните за пределами области графика, чтобы начать его построение.

Трехмерные, или 3D-графики, отображают функции двух переменных вида Z(X, Y). При построении трехмерных графиков в ранних версиях MathCad поверхность нужно было определить математически (рисунок 3.1.4, способ 2). Теперь применяют функцию MathCad *CreateMesh.*

CreateMesh(F (или G, или f1, f2, f3), x0, x1, y0, y1, xgrid, ygrid, fmap)

Создает сетку на поверхности, определенной функцией *F*. *x*0, *x*1, *y*0, *y*1 - диапазон изменения переменных, *xgrid*, *ygrid* - размеры сетки переменных, *fmap* - функция отображения. Все параметры, за исключением *F*, - факультативные. Функция *CreateMesh* по умолчанию создает сетку на поверхности с диапазоном изменения переменных от -5 до 5 и с сеткой 20020 точек.

Пример использования функции *CreateMesh* для построения 3D-графиков приведен на рисунке 3.1.4, способ 1.

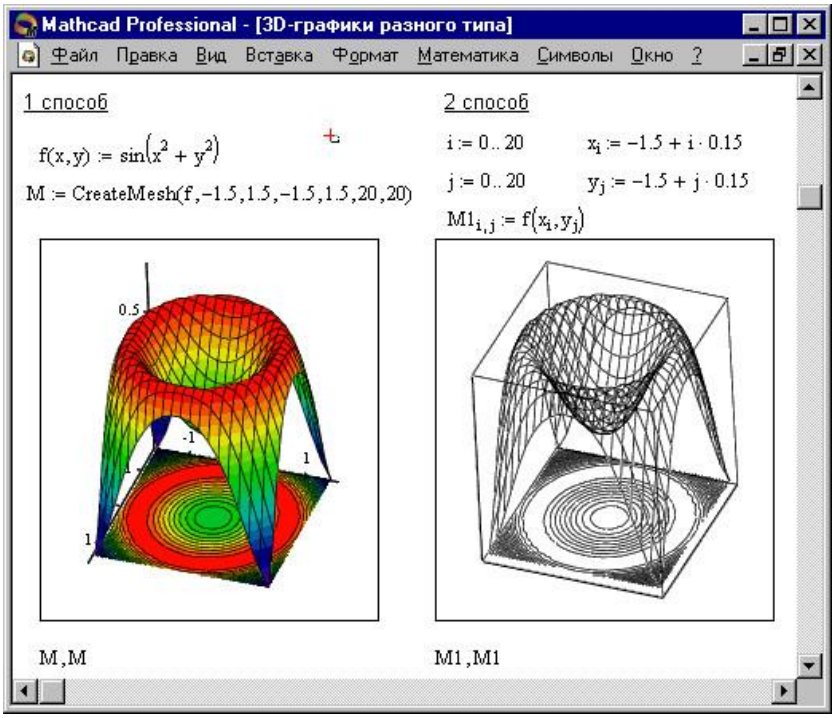


Рисунок 3.1.4 - Различные способы отображения поверхностей

На рисунке 3.1.4 построена одна и та же поверхность разными способами, с разным форматированием, причем изображены поверхности и под ними те же поверхности в виде контурного графика. Такое построение способно придать рисунку большую наглядность.

Нередко поверхности и пространственные кривые представляют в виде точек, кружочков или иных фигур. Такой график создается операцией **Вставка → График → 3D Точечный**, причем поверхность задается параметрически - с помощью трех матриц (X, Y, Z).

Для определения исходных данных для такого вида графиков используется функция *CreateSpace* (рисунок 3.1.5, способ 1).

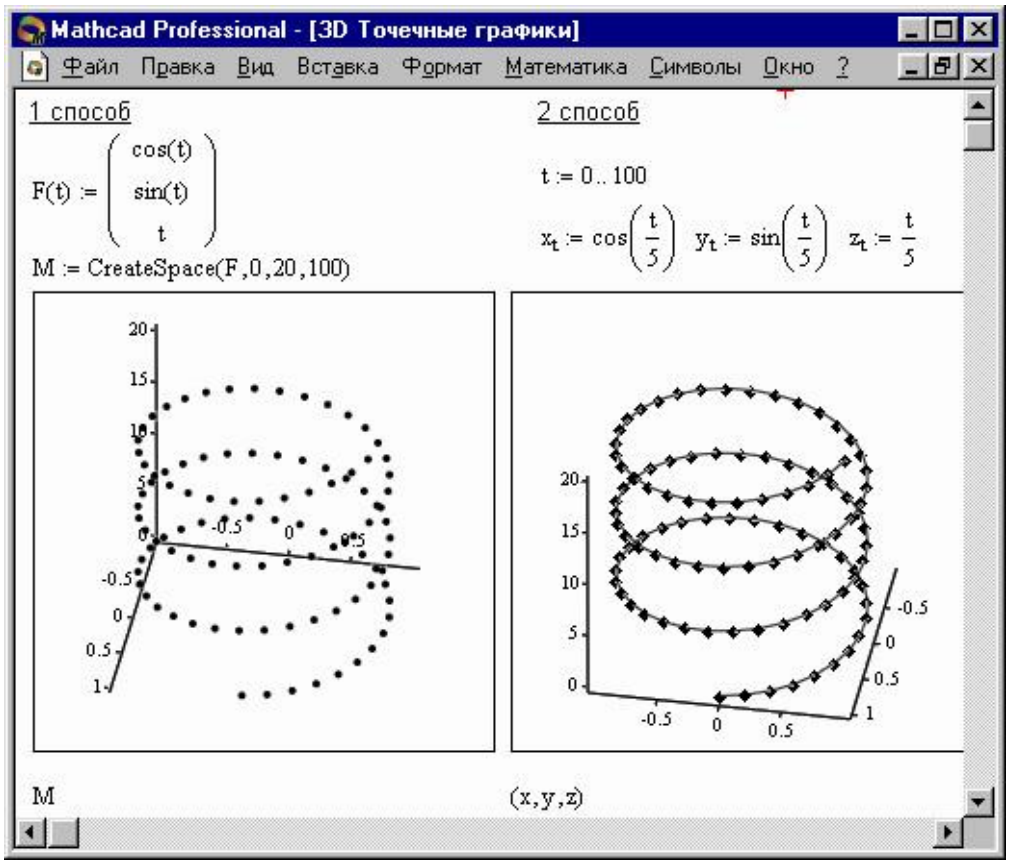


Рисунок 3.1.5 – Определение исходных данных (способ 1)

CreateSpace (F , t0, t1, tgrid, fmap)

Возвращает вложенный массив трех векторов, представляющих *x*-, *y*-, и *z*-координаты пространственной кривой, определенной функцией *F*. *t0* и *t1* - диапазон изменения

переменной, *tgrid* - размер сетки переменной, *fmap* - функция отображения. Все параметры, за исключением *F*, - факультативные.

Построение пересекающихся фигур

Особый интерес представляет собой возможность построения на одном графике ряда разных фигур или поверхностей с автоматическим учетом их взаимного пересечения. Для этого надо раздельно задать матрицы соответствующих поверхностей и после вывода шаблона 3D-графика перечислить эти матрицы под ним с использованием в качестве разделителя запятой (рисунок 3.1.6).

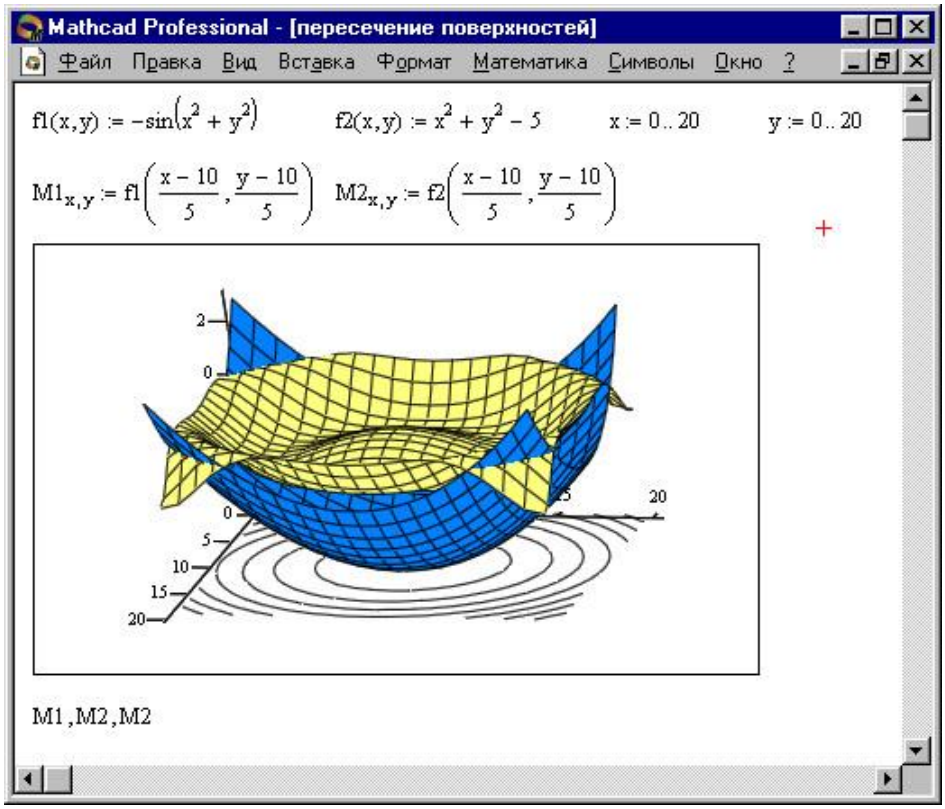


Рисунок 3.1.6 – Перечисление матриц соответствующих поверхностей

Создание анимационного клипа

MathCad имеет встроенную переменную FRAME, чье единственное назначение - управление анимациями:

- Создайте объект, чей вид зависит от FRAME.

- Убедитесь, что установлен режим автоматического расчета (**Математика → Автоматическое → Вычисление**).
- Выберите **Вид → Анимация** для вызова одноименного диалогового окна.
- Заключите в выделяющий пунктирный прямоугольник часть рабочего документа, которую нужно анимировать.
- Установите нижние и верхние границы FRAME (поля **От:** и **До:**).
- В поле **Скорость** введите значение скорости воспроизведения (кадров/сек).
- Выберите **Анимация**. Сейчас анимация только создается.
- Сохраните анимацию как AVI файл (**Сохранить как**).
- Воспроизведите сохраненную анимацию **Вид → Воспроизведение.**

Порядок выполнения лабораторной работы

Упражнение 3.1.1
Знакомство с интерфейсом MathCad

Войдите в MathCad

1. Поиграйте красным крестиком на экране, переместите его курсором по экрану. Введите любой символ. Появится рамка - шаблон. Нажав на клавишу мыши, выделите участок с рамкой, нажмите кнопку с ножницами на панели инструментов. Шаблон исчезнет.

2. Рассмотрите в верхней части экрана меню с двумя группами команд.

Выведите панели инструментов. Для этого нажав мышью кнопку команды **View** (вид), выведите подменю с рядом подкоманд. Если около кнопок с надписями **Toolbar** (инструментальная панель), **Mathpallete** (математическая панель) и **Formatbar (**панель форматирования) не стоит галочка, нажмите на них и этим выведите на экран эти панели.

3. Освойте перемещение панелей по экрану и их преобразование в строки меню.

Упражнение 3.1.2
Элементарные вычисления в MathCad

Все формулы в MathCad набираются только в латинском алфавите, поэтому, прежде чем начинать работу, перейдите на английский язык.

1. Выведите показанную на рисунке 3.1.7 математическую панель (если она не выведена).



Рисунок 3.1.7 – Математическая панель MathCad

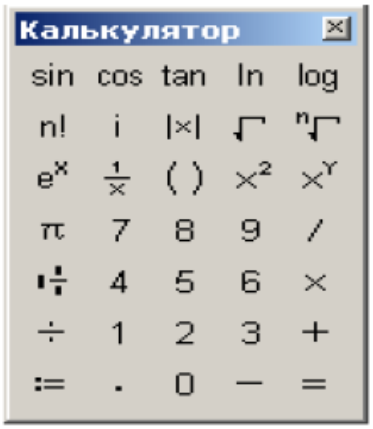


Рисунок 3.1.8 – Панель калькулятора

Нажмите кнопку с изображением калькулятора. Появится показанная на рисунке 3.1.8 панель калькулятора. На ней имеется ряд кнопок, например, таких как

n!(факториал n),
тригонометрических функций (**tan, sin, cos**),
логарифмов (**ln, log**) ,
цифры и знаки =, +, -,
:= (присвоение).

Следует различать кнопки := (присвоение) и = (равно).

Здесь же находятся кнопки x^2, x^y, служащие для возведения в степень, кнопки вычисления корня, нахождения модуля.

Арифметические действия в MathCad можно совершать, вводя знаки операций с клавиатуры или с панели калькулятора. Знак умножения в MathCad - точка, но набирается он на клавиатуре знаком *. Деление набирается клавишей /.

Если в процессе решения задачи происходит нарушение грамматики MathCad, то все содержание шаблона окрашивается в красный цвет и появляется разъяснение ошибки. К сожалению, часто на хорошем английском языке.

А) Вычислить 4! (факториал числа четыре).

Нажмем кнопку «n!». На экране, в том месте, где расположен крестик, появится шаблон: прямоугольная рамка, внутри которой расположен черный прямоугольник со знаком «!». Подведя курсор к этому прямоугольнику, введем мышью или с клавиатуры число 4 и нажмем кнопку «=» на клавиатуре или на панели вычислений. Мгновенно высветится ответ. Зачерненная точка рядом с ответом служит для вставки размерности, пока мы ею не пользуемся.

Б) Вычислить логарифм натуральный от 25.

Аналогично предыдущему, нажмем на панели калькулятора кнопку «ln», внутри появившихся кнопок вставим число 25 и, нажав «=», получим ответ.

Аналогично вычисляются sin, cos, tg любого угла в радианах, десятичный логарифм log, модуль числа.

В) Вычислить e^{25}**.** Нажмем на панели калькулятор кнопку «ex». В появившемся зачерненном прямоугольнике верхнего индекса наберем число 25, нажмем «=»**.**

Большую роль в наборе чисел играет расположение уголка. На экране – он голубого цвета. Например (рисунке 3.1.9), если уголок расположен, как показано в левой части рисунка, то любые знаки операций (сложение, вычитание и т.п.) будут добавляться к показателю степени, если же уголок расположен, как показано в

правой части рисунка, то они будут добавляться ко всему выражению**.**



Рисунок 3.1.9 - Влияние расположения уголка на вычисления в MathCad

Г) Вычислить два выражения:

$$e^{15}+\sqrt{47+56^6+\sin(0.6)}$$

$$e^{15}+\sqrt{47}+56^6+\sin(0.6)$$

Выражения отличаются тем, что в первом случае корень извлекается из трех слагаемых, а во втором случае только из числа 47.

Для набора первого выражения наберем e^{15}, как это было объяснено в предыдущем примере и, добившись того, чтобы уголок обнимал всю степень, нажмем +. После этого нажмем кнопку $\sqrt{\ }$ калькулятора, введем число 47, добьемся, чтобы уголок «обнимал» только число 47 и продолжим набор выражения. Для набора 56^6 наберем сначала 56, затем нажмем на панели калькулятор кнопку x^y и введем показатель степени 6.

При наборе второго выражения уголок после ввода числа 47 должен «обнимать» как число 47, так и корень. Остальной набор не отличается от первого примера.

Д) Вычислить $\dfrac{5\cdot|-6|+4}{8^2}$ **дробь**

При решении задачи знак модуля вводится с панели калькулятора, а дробь – с кнопки клавиатуры / (нижний правый угол клавиатуры)

$$\frac{5\cdot|-6|+4^2}{8^2}=16.469$$

Упражнение 3.1.3

Вычислить:

$\sqrt{100}=$ $|-10|=$ 10!

Это и все остальные задания снабдить комментариями, используя команду **Вставка → Текстовая область**.

Упражнение 3.1.4

Определить переменные: *a*:=3.4, *b*:=6.22, *c*≡0.149 (причем переменную *c* - глобально) и выражения:

$$z:=\frac{2ab+\sqrt[3]{c}}{\sqrt{\left(a^2+b^{a+c}\right)\cdot c}} \qquad N:=e^{\sin c}\cos\frac{a}{b}$$

- Вычислить выражения.
- С помощью команды **Формат → Результат → Формат чисел → Число знаков** изменить точность отображения результатов вычисления глобально.

Упражнение 3.1.5

Вычислить: $10x^2-5y^2$ при x=1,5 и y=-1,6.

Решение. На экране набираем; с клавиатуры набираем знак =, компьютер сам поставит знак :=.

x: =1.5 y: =-1.6

$10x^2-5y^2$ $y^2=$

рядом со знаком равенства читаем ответ: 9.1.7.

Упражнение 3.1.6

Вычислите: $\dfrac{36^{-1/2}}{27^{1/3}-81^{1/4}\cdot 5}$.

Решение. На экране набираем искомый пример. Ставим знак равенства и читаем ответ: - 0. 014.

Вычисление функций

Все вычисления в MathCad можно производить, набирая их на клавиатуре, или с помощью окна встроенных функций. Простые выражения типа вычисления функции набираются непосредственно на экране

Упражнение 3.1.7

Вычислить функцию $y = 4x^2 + 5x + 3$ для x = 1,2,…..10.

Решение. Сначала (рисунок 3.1.10) набирается диапазон значений x как x:=1,2..10. Тогда x := 1 , 2 .. 5

$$y(x) := 4\cdot x^2 + 5\cdot x + 8$$

x =	y(x) =
1	17
2	34
3	59
4	92
5	133

Рисунок 3.1.10 - Вычисление многочлена

Здесь:

1) используется знак присвоения «:=» , а не знак «=» ,

2) набирается первое значение x, затем через запятую второе его значение, чем задается шаг вычислений, и, наконец, последнее значение.

Две точки между 2 и 10 набираются нажатием клавиши с русской буквой Ж клавиатуры или кнопки **m..n** панели матрицы.

Затем, опять-таки через знак присвоения, набирается выражение для y. Причем следует набирать y(x), а не просто y.

Маткад выполняет команды слева направо и сверху вниз. Поэтому выражение для y(x) должно быть расположено справа и несколько ниже выражения «x :=»

После этого следует набрать «x=» (равно, а не присвоить) и появится столбец со всеми значениями x. Так же после нажатия «y(x)=» (y(x) равно) появляется столбец вычисленных значений y(x). Построение диапазона изменения аргумента x называется ранжировкой.

Упражнение 3.1.8

Вычислить функцию: $y = 3x^4 - 7x^3 + 4x^2 - 9x + 2$ для всех значений *x* таких, что $x = 0,\ 0.5,\ 1,\ 1.5,\ ...,\ 3$.

Пояснение. Читатель должен обратить внимание, что автор для повышения его внимания задал излишнее количество данных для ранжировки x. Необходимыми данными, которые и следует вводить, являются: первое значение, второе значение через запятую, последнее значение после двоеточия, вводимого, как указано выше.

Упражнение 3.1.9

Вычислить функцию двух переменных $z := 3x^2 + 4x + 8$ для значений $x = 1,\ 1.5,\ 2.0,\ ...,\ 5$ и для значений $y = 0,\ 0.5,\ 1.0,\ ...,\ 5$.

Пояснение. Задача решается аналогично предыдущим:

Сначала производится ранжировка обеих независимых переменных x и y.

Затем набирается выражение для функции «z(x,y):=»

После этого набирается «x=», «y=» и «z(x,y)=»

Упражнение 3.1.10

Вывести на экран значение системной константы π и установить максимальный формат ее отображения локально.

Упражнение 3.1.11

Выполнить следующие операции с комплексными числами:

$|Z| =$ $\quad$ $\mathrm{Re}(Z) =$ $\quad$ $\mathrm{Im}(Z) =$ $\quad$ $\arg(Z) =$ $\quad$ $Z := -3 + 2i$

$\sqrt{Z} =$ $\quad$ $\sqrt{-5} =$ $\quad$ $2 \cdot Z =$ $\quad$ $Z1 := 1 + 2i$ $\quad$ $Z2 := 3 + 4i$

$Z1 + Z2 =$ $\quad$ $Z1 - Z2 =$ $\quad$ $Z1 \cdot Z2 =$ $\quad$ $Z1 / Z2 =$

Упражнение 3.1.12

Выполнить следующие операции:

$i := 1...10$ $\quad$ $\sum_i i =$ $\quad$ $\prod_i (i+1) =$ $\quad$ $\int_0^{0.4} x^2 \cdot \lg(x+2)dx =$

$x := 2$ $\quad$ $\frac{d}{dx} x^5 =$ $\quad$ $\frac{d}{dx} \sin(x) =$ $\quad$ $\int_{0.8}^{1.2} \frac{\mathrm{ctg}2x}{(\sin 2x)} dx =$

Упражнение 3.1.13

Определить векторы d, S и R через дискретный аргумент i. Отобразить графически таблично заданные функции $S_i(d_i)$ и $R_i(d_i)$, используя команду **Вставка → График → X-Y Зависимость** (рисунок 3.1.11).

i	d_i	S_i	R_i
0	0.5	3.3	2
1	1	5.9	3.9
2	1.5	7	4.5
3	2	6.3	3.7
4	2.5	4.2	1.2

Рисунок 3.1.11 – Пример графического отображения таблично заданные функции

Чтобы оформить график, необходимо выполнить следующие команды:

Щелкнуть левой клавишей мыши на графике, чтобы выделить его. Затем щелкнуть правой клавишей мыши, при этом появится контекстное меню в котором необходимо выбрать команду **Формат** (появится диалоговое окно **“Formatting Currently Selected X-Y Plot”**).

Нанести линии сетки на график (**Оси X-Y** $\rightarrow$ **Вспом. линии**) и отобразить легенду (**След** $\rightarrow$ **Скрыть легенду**)

Отформатировать график так, чтобы в каждой узловой точке графика функции $S_i(d_i)$ стоял знак вида (**След** $\rightarrow$ **Символ** $\rightarrow$ **box**), а график функции $R_i(d_i)$ отобразить в виде гистограммы (**След** $\rightarrow$ **Тип** $\rightarrow$ **bar**).

Упражнение 3.1.14

Построить декартовы (**X-Y Зависимость**) и полярные (**Полярные Координаты**) графики следующих функций: $X(\alpha) := \cos(\alpha)\cdot\sin(\alpha)$, $Y(\alpha) := 1.5\cos(\alpha)^2 - 1$, $P(\alpha) := \cos(\alpha)$.

Для этого необходимо определить α как дискретный аргумент на интервале от 0 до $2\cdot\pi\cdot c$ шагом $\pi/30$.

Определить по графику **X-Y Зависимость** координаты любой из точек пересечения графиков $Y(\alpha)$ и $P(\alpha)$, для этого необходимо:

- Выделить график и выбрать из контекстного меню **Масштаб** (появится диалоговое окно **“X-Y Zoom”**) для увеличения части графика в области точки пересечения.
- На чертеже выделить пунктирным прямоугольником окрестность точки пересечения графиков $Y(\alpha)$ и $P(\alpha)$, которую нужно увеличить.
- Нажать кнопку **Масштаб+**, чтобы перерисовать график.
- Чтобы сделать это изображение постоянным, выбрать ОК.
- Выбрать из контекстного меню Трассировка (появится диалоговое окно **“X-Y Trace”**).

- Внутри чертежа нажать кнопку мыши и переместить указатель мыши на точку, чьи координаты нужно увидеть.
- Выбрать **Copy X** (или **Copy Y**), на свободном поле документа набрать *Xper* := (или *Yper* :=) и выбрать пункт меню **Правка → Вставка.**
- Вычислить значения функций $X(\alpha)$ и $Y(\alpha)$ при $\alpha := \pi / 2$.

Упражнение 3.1.15

Используя команду **Вставка → Матрица** создать матрицу Q размером 6×6, заполнить ее произвольно и отобразить графически с помощью команды **Вставка → График → Поверхности**.

Упражнение 3.1.16

Построить график поверхности (**Поверхности**) и карту линий уровня (**Контурный**) для функции двух $X(t, \alpha) := i \cdot \cos(\alpha) \cdot \sin(\alpha)$, двумя способами:

1. С помощью функции *CreateMesh* (сетка размером 40×40, диапазон изменения t от -5 до 5, α - от 0 до $2 \cdot \pi$).

2. Задав поверхность математически, для этого:

- Определить функцию $X(t, \alpha)$
- Задать на осях переменных t и α по 41 точке *i*:=0..40 *j*:=0..40 для переменной t_i со значениями, изменяющимися от -5 до 5 с шагом 0.25 $t_i := -5 + 0.25 \cdot i$, а для переменной α_j - от 0 до $2 \cdot \pi$ с шагом $\pi / 20$ $\alpha_j := \pi / 20 \cdot j$.
- Определить матрицу $M_j := X(t_j, \alpha_j)$ и отобразить ее графически.

С помощью команды **Формат** контекстного меню вызвать диалоговое окно “**Формат 3-D графика”** и изменить:

- характеристики просмотра (**Общее → Вид → Вращение, Наклон**),
- цвета и линии поверхности (**Внешний Вид → Свойства линии, Свойства заливки**),
- параметры осей (**Оси**),
- вид заголовка графика (**Название**).

Упражнение 3.1.17

Отобразить графически пересечение поверхностей $f1(x,y) := \frac{(x+y)^2}{10}$ и $f2(x,y) := 5\cdot\cos\left(\frac{x-y}{3}\right)$. Матрицы для построения поверхностей задать с помощью функции *CreateMesh*, значения факультативных параметров не указывать. Выполнить однотонную заливку для поверхностей, выбрав из контекстного меню команду **Формат**. Также из контекстного меню выбрать эффекты **Туман**, **Освещение, Перспектива**.

Упражнение 3.1.18

Используя переменную **FRAME** и команду **Вид → Анимация**, создать анимационные клипы с помощью данных приведенных в Таблице 3.1.1.

Таблица 3.1.1 – Данные для анимационных клипов

№ вар.	Переменные и функции	FRAME	Тип графика
1	$x := 0, 0.1, ..., 30$ $f(x) := x + FRAME$	от 0 до 20	График Полярные Координаты
2	$f(x) := 0....FRAME + 1$ $g_i := 0.5\cdot i\cdot\cos(i)$ $h_i := i\cdot\sin(i)$ $k_i := 2\cdot i$	от 0 до 50	3D точечный график границы на осях Min Max X -50 50 Y -50 50 z 0 50 В вместе для ввода матрицы укажите (g, h, k)
3	$i := 0 \ ... \ 20 \quad j := 0 \ ... \ 20$ $f(x,y) := \sin(x^2 + y^2 + FRAME)$ $x_i := -1.5 + 0.15\cdot i$ $y_j := -1.5 + 0.15\cdot j$ $M_{i,j} := f(x_i, y_i)$	от 0 до 50	График Поверхности В метке для ввода матрицы укажите M

Продолжение таблицы 3.1.1

4	r := FRAME R := 6 n := 0 ... 20 m := 0 ... 20 $v_n := \frac{2 \cdot \pi \cdot n}{r+1}$ $w_m := \frac{2 \cdot \pi \cdot m}{r+1}$ $x_{m\,n} := (R + r \cdot \cos(v_n)) \cdot \cos(w_m)$ $y_{m\,n} := (R + r \cdot \cos(v_n)) \cdot \sin(w_m)$ $z_{m\,n} := r \cdot \sin(v_n)$	от 0 до 20	График Поверхности (границы на всех осях установить от -11 до 11) В метке для ввода матрицы укажите (*x*, *y*, *z*)

Контрольные вопросы

1. С помощью какого оператора можно вычислить выражение?
2. Как вставить текстовую область в документ MathCad?
3. Чем отличается глобальное и локальное определение переменных?
4. Как изменить формат чисел для всего документа?
5. Как изменить формат чисел для отдельного выражения?
6. Какие системные переменные Вам известны? Как узнать и изменить их значение?
7. Какие виды функций в MathCad Вам известны?
8. Как вставить встроенную функцию в документ MathCad?
9. Какие операторы позволяют вычислить интеграл, производную, сумму, произведение?
10. Как определить дискретные переменные с произвольным шагом?
11. Как определить индексированную переменную?
12. Какие виды массивов в MathCad Вам известны?
13. Какая переменная определяет нижнюю границу индексации элементов массива?
14. Опишите способы создания массивов в MathCad.
15. Как просмотреть содержимое массива, определенного через дискретный аргумент?
16. Как построить графики: поверхности; полярный; декартовый?

17. Как построить несколько графиков в одной системе координат?

18. Как изменить масштаб графика?

19. Как определить координату точки на графике?

20. Как построить гистограмму?

3.2 Символьные вычисления

Системы компьютерной алгебры снабжаются специальным процессором для выполнения аналитических (символьных) вычислений. Его основой является ядро, хранящее всю совокупность формул и формульных преобразований, с помощью которых производятся аналитические вычисления. Чем больше этих формул в ядре, тем надежней работа символьного процессора и тем вероятнее, что поставленная задача будет решена, если такое решение существует в принципе (что бывает далеко не всегда).

Ядро символьного процессора системы MathCad - несколько упрощенный вариант ядра известной системы символьной математики Maple V фирмы Waterloo Maple Software, у которой фирма MathSoft (разработчик MathCad) приобрела лицензию на его применение, благодаря чему MathCad стала (начиная с версии 3. 0) системой символьной математики. Символьные вычисления выполняются столь же просто (для пользователя), как вычисление квадрата *x*.

Символьные операции можно выполнять двумя способами:

- Непосредственно в командном режиме (используя операции меню **Символы**);
- С помощью операторов символьного преобразования (используя палитру инструментов **Символы** ).

Выделение выражений для символьных вычислений. Чтобы символьные операции выполнялись, процессору необходимо указать, над каким выражением эти операции должны производиться, т. е. надо выделить выражение. Для ряда операций следует не только указать выражение, к которому они относятся, но и наметить переменную, относительно которой выполняется та или иная символьная операция. Само выражение в таком случае не выделяется.

Таким образом, для выполнения операций с символьным процессором нужно выделить объект (целое выражение или его часть) синими сплошными линиями.

Символьные операции разбиты на пять характерных разделов. Первыми идут наиболее часто используемые операции. Они могут выполняться с выражениями, содержащими комплексные числа или имеющими решения в комплексном виде.

Операции с выделенными выражениями

Если в документе есть выделенное выражение, то с ним можно выполнять различные операции, представленные ниже:

Расчеты - преобразовать выражение с выбором вида преобразований из подменю;

Символические [Shift] F9 - выполнить символьное преобразование выделенного выражения;

С плавающей запятой… - вычислить выделенное выражение в вещественных числах;

Комплексные - выполнить вычисления в комплексном виде;

Упростить - упростить выделенное выражение с выполнением таких операций, как сокращение подобных слагаемых, приведение к общему знаменателю, использование основных тригонометрических тождеств и т д.;

Расширить - раскрыть выражение [например, для $(X+Y)(X-Y)$ получаем X^2-Y^2];

Фактор - разложить число или выражение на множители [например, X^2-Y^2 даст $(X+Y)(X-Y)$];

Подобные - собрать слагаемые, подобные выделенному выражению, которое может быть отдельной переменной или функцией со своим аргументом (результатом будет выражение, полиномиальное относительно выбранного выражения);

Коэффициенты Полинома - по заданной переменной найти коэффициенты полинома, аппроксимирующего выражение, в котором эта переменная использована.

Операции с выделенными переменными

Для ряда операций надо знать, относительно какой переменной они выполняются. В этом случае необходимо выделить переменную, установив на ней маркер ввода. После этого становятся доступными следующие операции подменю **Переменные**:

Вычислить - найти значения выделенной переменной, при которых содержащее ее выражение становится равным нулю;

Замена - заменить указанную переменную содержимым буфера обмена;

Дифференциалы - дифференцировать выражение, содержащее выделенную переменную, по этой переменной (остальные переменные рассматриваются как константы);

Интеграция - интегрировать все выражение, содержащее переменную, по этой переменной;

Разложить на составляющие... - найти несколько членов разложения выражения в ряд Тейлора относительно выделенной переменной;

Преобразование в Частичные Доли - разложить на элементарные дроби выражение, которое рассматривается как рациональная дробь относительно выделенной переменной.

Операции с выделенными матрицами

Операции с выделенными матрицами представлены позицией подменю **Матрицы**, которая имеет свое подменю со следующими операциями:

Транспонирование - получить транспонированную матрицу;

Инвертирование - создать обратную матрицу;

Определитель - вычислить детерминант (определитель) матрицы.

Результаты символьных операций с матрицами часто оказываются чрезмерно громоздкими и поэтому плохо обозримы.

Операции преобразования

В позиции **Преобразование** содержится раздел операций преобразования, создающий подменю со следующими возможностями:

Фурье - выполнить прямое преобразование Фурье относительно выделенной переменной;

Фурье Обратное - выполнить обратное преобразование Фурье относительно выделенной переменной;

Лапласа - выполнить прямое преобразование Лапласа относительно выделенной переменной (результат - функция переменной s);

Лапласа Обратное - выполнить обратное преобразование Лапласа относительно выделенной переменной (результат - функция переменной t); Z - выполнить прямое Z- преобразование выражения относительно выделенной переменной (результат - функция переменной z);

Обратное Z - выполнить обратное Z-преобразование относительно выделенной переменной (результат - функция переменной n).

Стиль представления результатов вычислений

На наглядность вычислений влияет стиль представления их результатов. Следующая команда позволяет задать тот или иной стиль:

Стиль Вычислений... - задать вывод результата символьной операции под основным выражением, рядом с ним или вместо него (рисунок 3.2.1).

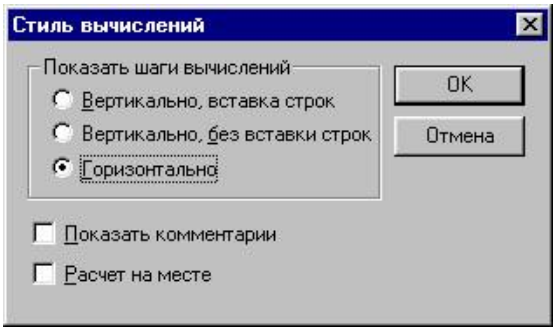


Рисунок 3.2.1 – Окно «стиль Вычислений…»

Примеры символьных операций в командном режиме

Большинство символьных операций легко выполняются, так что ниже мы остановимся лишь на некоторых примерах. Символьная операция **Расчеты** обеспечивает работу с математическими выражениями, содержащими в строенные в систему функции и представленными в различном виде: полиномиальном, дробно-рациональном, в виде сумм и произведений, производных и интегралов и т. д. (рисунок 3.2.2).

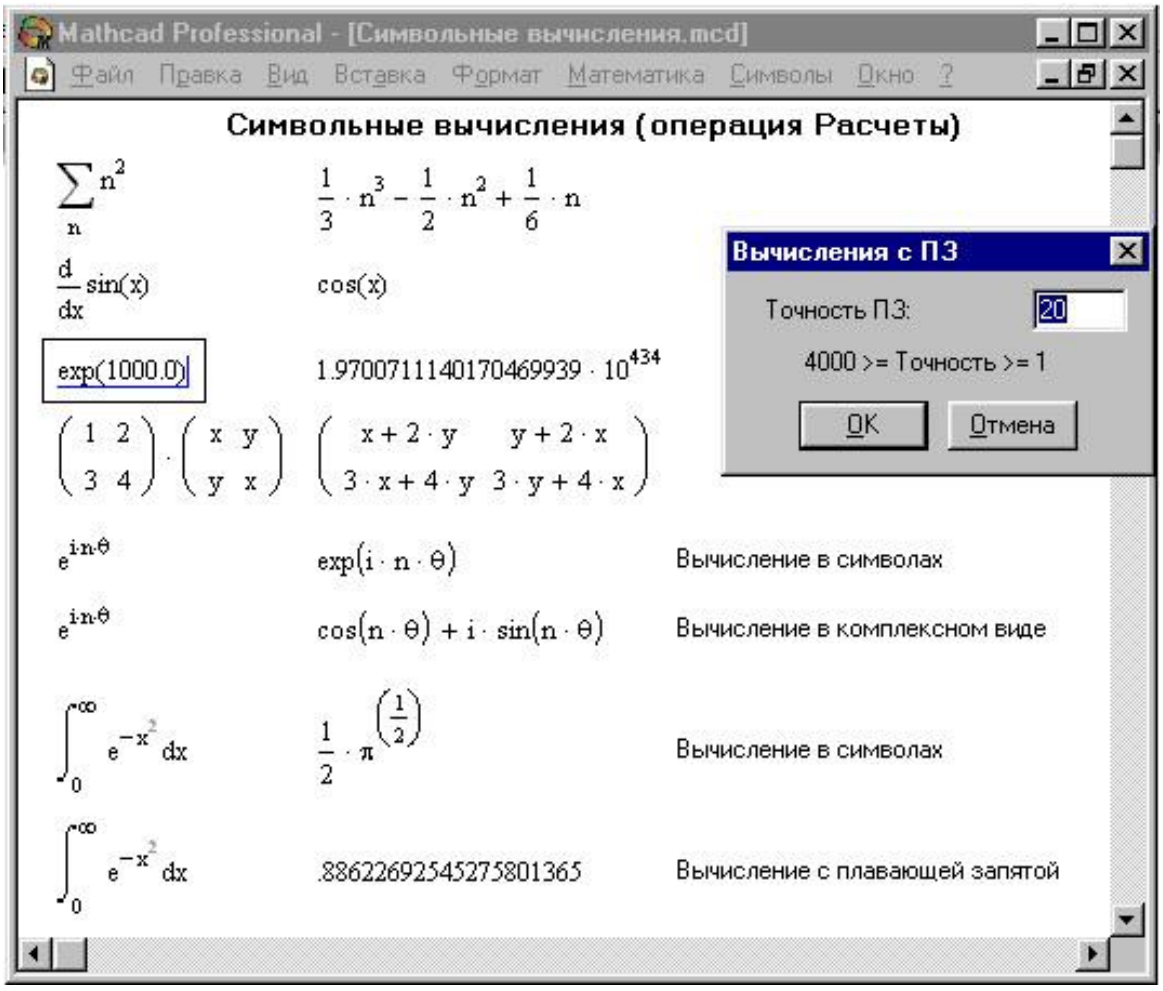


Рисунок 3.2.2 – Пример работы с математическими выражениями

Операция стремится произвести все возможные численные вычисления и представить выражение в наиболее простом виде. Она возможна над матрицами с символьными элементами. Производные и определенные интегралы, символьные значения

которых вычисляются, должны быть представлены в своей естественной форме.

В MatCad предусмотрена возможность выполнения численных вычислений с повышенной точностью - 20 знаков после запятой.

Для перехода в такой режим вычислений нужно числовые константы в вычисляемых объектах задавать с обязательным указанием десятичной точки, например, 10.0 или 3.0, а не 10 или 3. Этот признак является указанием на проведение вычислений такого типа.

На рисунке 3.2.2 показаны типовые примеры действия операции **Расчеты.**

Здесь слева показаны исходные выражения, подвергаемые символьным преобразованиям, а справа - результат этих преобразований.

Операция **Расчеты** одна из самых мощных.

Как видно из рисунка 3.2.2, она позволяет в символьном виде вычислять суммы (и произведения) рядов, производные и неопределенные интегралы, выполнять символьные и численные операции с матрицами.

Эта операция содержит подменю.

Команда **Символические** тут наиболее важная. Назначение других команд очевидно: они нужны, если результат требуется получить в форме комплексного или действительного числа.

К примеру, если вы хотите вместо числа π получить 3.141..., используйте команду **С плавающей запятой….** В режиме символьных вычислений результат может превосходить машинную бесконечность системы - см. пример на вычисление exp(1000.0) на рисунке 3.2.2. При этом число точных значащих цифр результата практически не ограничено (или, точнее говоря, зависит от емкости ОЗУ).

Операция **Разложить на составляющие...** возвращает разложение в ряд Тейлора выражения относительно выделенной переменной с заданным по запросу числом членов ряда n (число определяется по степеням ряда).

По умолчанию задано n=6.

В разложении указывается остаточная погрешность разложения. На рисунке 3.2.3 представлено применение этой операции для разложения функции $\frac{\sin(x)}{x}$.

Минимальная погрешность получается при малых x (см. графическое представление функции и ее ряда).

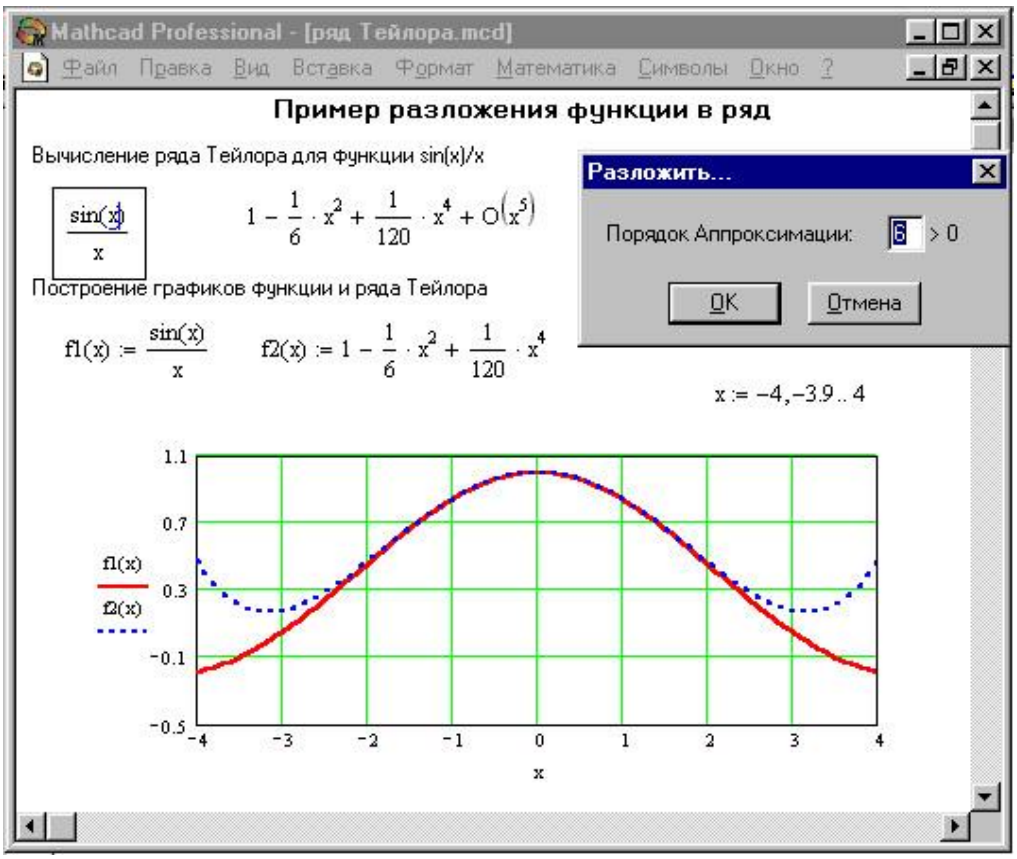


Рисунок 3.2.3 - Режим символьных вычислений

Операторы вычисления пределов функций

Для вычисления пределов функций в систему введен символьный оператор limit.

Помимо ввода с наборной панели **Матанализ,** его в трех формах можно ввести нажатием следующих комбинаций клавиш:

[**Ctrl**] **L** - ввод шаблона оператора вычисления предела функции при x, стремящемся к заданному значению,

[**Ctrl**] **A** - ввод шаблона вычисления предела функции слева от заданной точки,

[**Ctrl**] **B** - ввод шаблона вычисления предела функции справа от заданной точки.

На ринке 3.2.4 показаны примеры вычисления пределов. При вычислении пределов нужно заполнить шаблоны, входящие в главный шаблон для вычисления пределов, а затем ввести функцию, имя переменной, по которой ищется предел, и значение переменной - аргумента функции.

Для получения результата установите после блока вычисления предела стрелку с острием, направленным вправо.

Предел (если он существует) будет вычислен и появится в шаблоне у острия стрелки.

Если функция не имеет предела, вместо результата появится надпись *Undefine*.

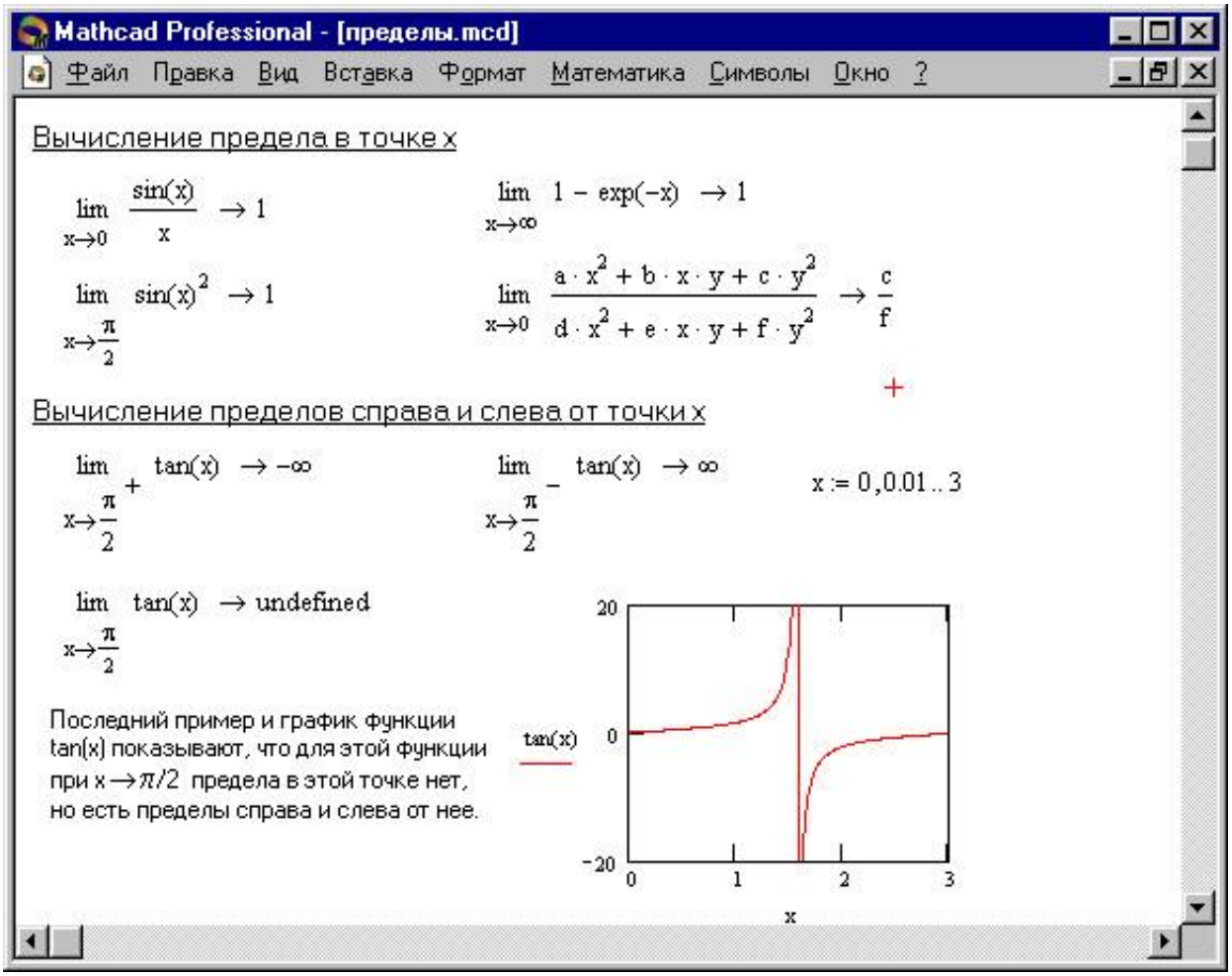


Рисунок 3.2.4 - Примеры вычисления пределов

Задание операторов пользователя

Еще одна экзотическая возможность, присущая новым версиям системы MathCad, - задание новых операторов пользователя. Такой оператор задается практически так же, как функция пользователя, но вместо имени выбирается какой-либо подходящий знак. Например, можно задать оператор деления в виде таблицы 3.2.1:

Таблица 3.2.1 – Варинты задание оператора деления

$+(A,B) := \frac{A}{B}$	$\div(6,2) = 3$	$6 \div 2 = 3$
задание нового оператора	применение функции деления	применение нового оператора деления

При кажущейся простоте такого задания здесь есть проблемы. Встроенные в систему операторы нельзя переопределить. Поэтому набор доступных знаков для обозначения новых операторов ограничен. Нельзя задать новый оператор деления знаком / (он уже использован), но можно взять знак ÷ поскольку этот символ системой не используется.

Вторая проблема связана с вводом символа нового оператора - напрямую его ввести нельзя. Придется воспользоваться типовыми приемами ввода новых символов в документы Windows. Один из этих приемов - использование приложения, выдающего таблицу символов, с возможностью его экспорта из этой таблицы в документ другого приложения.

Можно также воспользоваться подходящим знаком из набора MATH SYMBOL, имеющегося в составе Шпаргалок, доступ к которым дает Ресурс Центр (**Ресурс Центр → Справочный стол и краткое руководство → Дополнительные математические символы**). На рисунке 3.16 показан такой вариант задания нового оператора пользователя. Для перетаскивания знака можно скопировать его в буфер обмена с помощью операции **Копировать**, а затем ввести в документ, используя операцию **Вставка.**

После того как оператор задан, его можно использовать, как функцию и как оператор. Примеры показаны на рисунке 3.2.5.

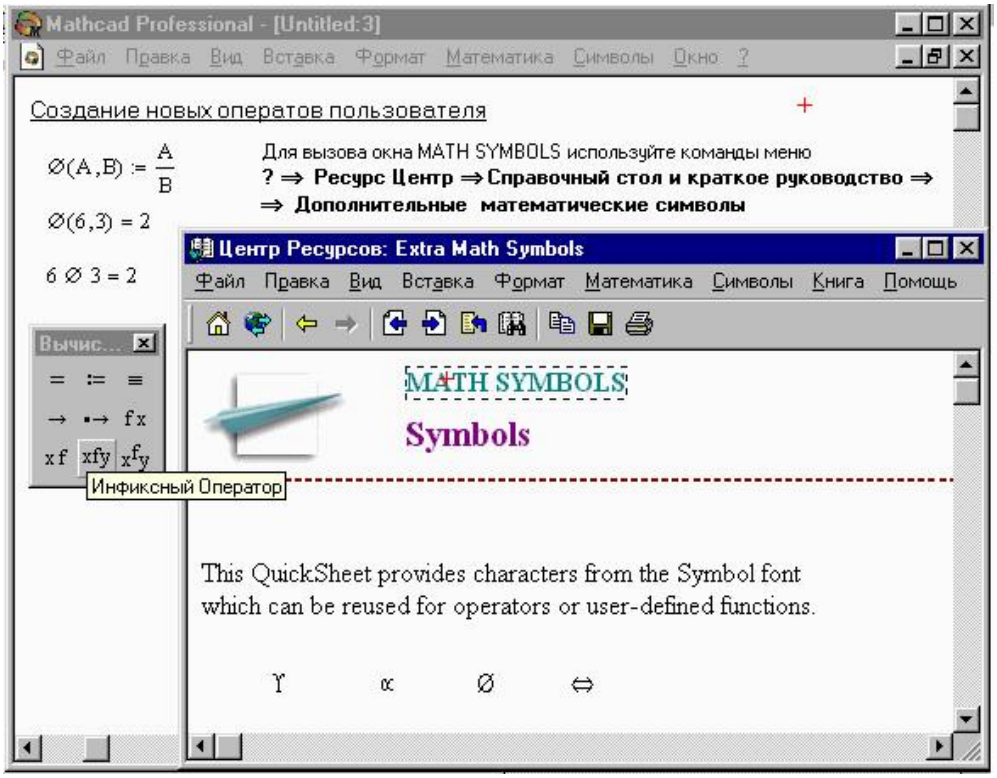


Рисунок 3.2.5 – Примеры задания операторов пользователя

Для применения нового оператора надо вывести его шаблон с помощью панели математических знаков (она также показана рисунке 3.2.5).

В нашем случае следует нажать кнопку 

этой панели - она выводит особый шаблон. Введите операнды, например 6 и 3 в крайние прямоугольники, а символ оператора - в средний. Поставив после этой конструкции знак равенства, увидите результат - число 2.

Можно задать и другие операторы, например, для работы с одним операндом. Так, вы можете задать оператор для пересчета значения температуры по шкале Цельсия, с тем чтобы определить соответствующее ему значение по шкале Фаренгейта, следующим образом

$$°C(x) := \frac{9}{5} \cdot x + 32 \qquad °F := 1.$$

Затем, используя кнопку xf наборной панели символов отношения, можно выполнять операцию пересчета в виде:

$$37°C := 98.6°F$$

Есть области математики и физики, где задание новых операторов необходимо, поскольку является частью специфического языка их описания.

Порядок выполнения работы

Упражнение 3.2.1

Используя операцию **Символы → Расчеты → С плавающей запятой…**, представьте:

1. число 28,375764 в 7 позициях;
2. число 12,345667 в 3 позициях.

Упражнение 3.2.2

Выведите следующие числа в комплексной форме, используя операцию **Расчеты → Комплексные** меню **Символы**:

1. $\sqrt{-7}$;
2. $\mathrm{tg}(a^{\sqrt{-3}})$;
3. $e^{1+\frac{x_i}{4}}$;
4. для выражения 3) последовательно выполните операции **Расчеты → Комплексные** и **Упростить** меню **Символы.**

Упражнение 3.2.3

Для полинома g(x) (таблица 3.2.2) выполнить следующие действия:

1. разложить на множители, используя операцию **Символы → Фактор;**

2. подставьте выражение x=y+z в g(x), используя операцию **Символы → Переменные → Замена** (предварительно скопировав подставляемое выражение в буфер обмена, выделив его и нажав комбинацию клавиш **Ctrl + C**);

3. используя операцию **Символы → Расширить,** разложите по степеням выражение, полученное в 2)**;**

4. используя операцию **Символы → Подобные,** сверните выражение, полученное в 3), по переменной z**.**

Таблица 3.2.2 - Варианты упражнения 3.2.3.

№ вар.	g(x)	№ вар.	g(x)
1	$x^4 - 2x^3 + x^2 - 12x + 20$	9	$x^4 + x^3 - 17x^2 - 45x - 100$
2	$x^4 + 6x^3 + x^2 - 4x + 60$	10	$x^4 - 5x^3 + x^2 - 15x + 50$
3	$x^4 - 14x^2 - 40x - 75$	11	$x^4 - 4x^3 - 2x^2 - 20x + 25$
4	$x^4 - x^3 + x^2 - 11x + 10$	12	$x^4 + 5x^3 + 7x^2 + 7x - 20$
5	$x^4 - x^3 - 29x^2 - 71x - 140$	13	$x^4 - 7x^3 + 7x^2 - 5x + 100$
6	$x^4 + 7x^3 + 9x^2 + 13x - 30$	14	$x^4 + 10x^3 + 36x^2 + 70x + 75$
7	$x^4 + 3x^3 - 23x^2 - 55x - 150$	15	$x^4 + 9x^3 + 31x^2 + 59x + 60$
8	$x^4 - 6x^3 + 4x^2 + 10x + 75$		

Упражнение 3.2.4

Разложите выражения на элементарные дроби используя операцию **Символы → Переменные → Преобразование в частичные доли:**

$$\frac{6x^2 - x + 1}{x^3 - x}; \qquad \frac{3x^2 - 2}{(x^2 + x + 1)(x + 1)}; \quad \frac{5x^2 - 4x + 16}{(x^2 - x + 1)(x - 3)}; \quad \frac{x + 1}{x(x - 1)^3}.$$

Упражнение 3.2.5

Разложите выражения в ряд с заданной точностью, используя операцию **Символы → Переменные → Разложить на составляющие:**

1. $\ln(1 + x)$, $x_0 = 0$, порядок разложения 6;

2. $\sin(x)^2$, $x_0 = 0$, порядок разложения 6.

Упражнение 3.2.6

Найти первообразную аналитически заданной функции f(x) (таблица 3.2.3), используя операцию **Символы → Переменные → Интеграция**.

Упражнение 3.2.7

Определить символьное значение первой и второй производных f(x) (таблица 3.2.3), используя команду **Символы → Переменные → Дифференциалы**.

Таблица 3.2.3 - Варианты упражнений 3.2.6 и 3.2.7

№ вар.	f(x)	№ вар.	f(x)	№ вар.	f(x)
1	$1/(\mathrm{tg}2x+1)$	6	$x^{2\mathrm{arctg}(x/3)}$	11	$(2x+3)\sin x$
2	$\cos x/(2x+5)$	7	$e^{2x}\sin 3x$	12	$\cos 3x/(1-\cos 3x)^2$
3	$1/(x^{\sqrt{x^3+4}})$	8	$\mathrm{ctg}2x/(\sin 2x)^2$	13	$1/(1+x+x^2)$
4	$\sin x/(1+\sin x)$	9	$(x+1)\sin x$	14	$(1+x)/(2+x)$
5	$x^{2\lg(x+2)}$	10	$5x+x\lg x$	15	$\sqrt{1+e^{-x}}$

Упражнение 3.2.8

1. Транспонируйте матрицу M

$$\begin{pmatrix} 1 & a & b \\ x & 2 & c \\ x^2 & 3 & d \end{pmatrix}$$

с помощью операции
Символы → Матрицы → Транспонирование.

2. Инвертируйте матрицу

$$\begin{pmatrix} 1 & y \\ x & 2 \end{pmatrix}$$

с помощью операции

Символы → Матрицы → Инвертирование.

3. Вычислите определитель матрицы M

$$\begin{pmatrix} 1 & a & b \\ x & 2 & c \\ x^2 & 3 & d \end{pmatrix}$$

с помощью операции **Символы → Матрицы → Определитель**.

Упражнение 3.2.9

Вычислите пределы:

$$\lim_{x\to 1}\frac{x^2+2\cdot x+5}{x^2+1} \qquad \lim_{x\to\frac{\pi}{2}}(2\cdot\sin(x)-\cos(x)+\operatorname{ctg}(x)) \qquad \lim_{h\to 0}\frac{(x+h)^3-x^3}{h}$$

$$\lim_{x\to\infty}\frac{\sqrt{x^2-3}}{\sqrt[3]{x^3}+1}\to 1 \qquad \lim_{x\to-\infty} x\cdot(\sqrt{x^2+1}-x)\to-\infty \qquad \lim_{x\to 0^+}(1+x)^{\frac{1}{x}}$$

$$\lim_{x\to+\infty} x\cdot(\sqrt{x^2+1}-x) \qquad \lim_{n\to\infty^-}\left(1+\frac{1}{n}\right)^n$$

Упражнение 3.2.10

Задайте операторы пользователя:

1.Для пересчета единиц электрической энергии (кВт ч в Дж, эВ в Дж) если известно, что

1 кВт ч=$3{,}6\cdot10^6$ Дж;

1 эВ=$1{,}602\cdot10^{-19}$ Дж.

2.Для пересчета единиц магнитной индукции (Вб/см^2 в Т, Гс в Т) если известно, что

1 Вб/см^2=$1\cdot10^4$ Т;

1 Гс =$1 \cdot 10^{-4}$ Т.

3.Для пересчета единиц мощности (эрг/с в Вт, кгс м/с в Вт) если известно, что

1 эрг/с =$1 \cdot 10^{-7}$ Вт;

1 кгс м/с = 9,80665 Вт.

Контрольные вопросы

1. Назовите способы выполнения символьных операций в MathCad.

2. Что необходимо сделать с выражением перед применением символьных преобразований в командном режиме?

3. Перечислите символьные операции с выделенными выражениями.

4. Перечислите символьные операции с выделенными переменными.

5. Перечислите символьные операции с выделенными матрицами.

6. Перечислите символьные операции преобразования.

7. Какие параметры определяет стиль представления результатов вычислений и где он задается?

8. В каких случаях результат символьных преобразований помещается в буфер обмена?

9. Как символьно решить уравнение или систему уравнений в MathCad? Какой знак равенства используется? Какой комбинацией клавиш вставляется в документ?

10. Назовите особенности использования символьного решения уравнений.

11. Каким образом можно вычислить предел в MathCad?

12. Для чего необходимо задание операторов пользователя?

13. Как задать оператор пользователя?

3.3 Решение уравнений средствами MathCad

Известно, что многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Нельзя также построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение степени выше четвертой. Однако такие уравнения могут решаться численными методами с заданной точностью (не более значения заданного системной переменной **TOL**).

3.3.1 Решение нелинейных и трансцендентных уравнений с одной переменной

Рассмотрим уравнение

$$f(x)=0 \qquad (3.3.1)$$

где f(x) определена и непрерывна на некотором конечном или бесконечном интервале $a<x<b$.

Всякое значение x^*, обращающее функцию f(x) в нуль, $f(x^*)\equiv 0$, называется корнем уравнения, а способ нахождения этого значения x^* и есть решение уравнения (3.3.1).

Найти корни уравнения вида (3.1) точно удается лишь в редких случаях. Кроме того, часто уравнение содержит коэффициенты, известные лишь приблизительно и, следовательно, сама задача о точном определении корней уравнения теряет смысл. Разработаны методы численного решения уравнений вида (3.3.1), позволяющие отыскать приближенные значения корней этого уравнения.

При этом приходится решать две задачи:

- отделение корней, т.е. отыскание достаточно малых областей, в каждой из которых заключен только один корень уравнения;
- 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 способом, либо двигаясь вдоль области определения с некоторым шагом, проверять на концах интервалов условие смены знака функции.

Для решения второй задачи существует многочисленные методы:

- метод итераций;
- метод половинного деления;
- метод хорд;
- метод касательных;
- др. методы.

3.2 Уточнение корней методом половинного деления

В качестве начального приближения выбирается значение:

$$c=(a+b)/2,$$

затем исследуются значения функции на концах отрезков: [a, c] и [c, b].

Для продолжения вычислений выбирается тот отрезок, у которого значение функции на концах имеет противоположные знаки.

Процесс продолжается до тех пор, пока не выполнится условие:

$$|b-a|<\varepsilon$$

где ε – заданная точность вычислений, например $\varepsilon=10^{-5}$.

3.3.3 Уточнение корней методом простой итерации

Пусть корни отделены и [a, b] содержит единственный корень. Уравнение (3.1) приводится к итерационному виду:

$$x = \varphi(x) \qquad (3.3.2)$$

где функция $\varphi(x)$ дифференцируема на [a, b] и $\forall x \in [a,b]$ выполняется условие сходимости метода простой итерации как

$$|\varphi'(x)| < 1,\ x \in [a,b] \qquad (3.3.3)$$

Функцию $\varphi(x)$ можно подобрать, например, в следующем виде:

$$\varphi(x) = x + kf(x) \qquad (3.3.4)$$

где k находится из условия

$$|\varphi'(x,k)| = |1 + kf'(x)| < 1,\ \forall x \in [a,b] \qquad (3.3.5)$$

Последнее условие гарантирует сходимость итерационной последовательности:

$$x_1, x_2, ..., x_{n-1}, x_n ... \qquad (3.3.6)$$

к искомому корню

$$x^* = \xi. \qquad (3.3.7)$$

Условием окончания счета будем считать выполнение неравенства

$$|x_n - x_{n-1}| < \frac{\varepsilon(1-q)}{q};\ q = \max|\varphi'(x)|. \qquad (3.3.8)$$

3.3.4 Расчетная формула для метода хорд:

$$x_{n+1} = \frac{x_0 f(x_n) - x_n f(x_0)}{(f(x_n) - f(x_0))} \qquad (3.3.9)$$

3.3.5 Расчетная формула для метода касательных:

$$x_{n+1} = x_{n-} \frac{f(x_n)}{f'(x_n)} \qquad (3.3.10)$$

3.3.6 Выбор начального приближения и условия завершения вычислений

Значение x_0 для метода хорд и начальная точка для метода касательных выбирается из условия выполнения неравенства:

$$f(x_0)f''(x_0) > 0. \qquad (3.3.11)$$

В результате вычислений по этим формулам может быть получена последовательность приближенных значений корня

$$x_1, x_2, ..., x_{n-1}, x_n ... \qquad (3.3.12)$$

Процесс вычислений заканчивается при выполнении условия:

$$|x_n - x_{n-1}| < \varepsilon, \qquad (3.3.13)$$

где ε – заданная точность вычислений, например $\varepsilon = 10^{-5}$.

В каждом случае на печать выводится количество итераций, необходимых для достижения заданной точности.

В качестве иллюстративного примера рассмотрим задачу вычисления корня заданного уравнения, построение таблиц значений и графиков функций, а также отделение корней уравнения:

$$y=x–\sin(x)–0,25. \qquad (3.3.14)$$

В этом случае:

Отделяем корни графически (рисунок 3.3.1):

y:=x–sin(x)–0,25.

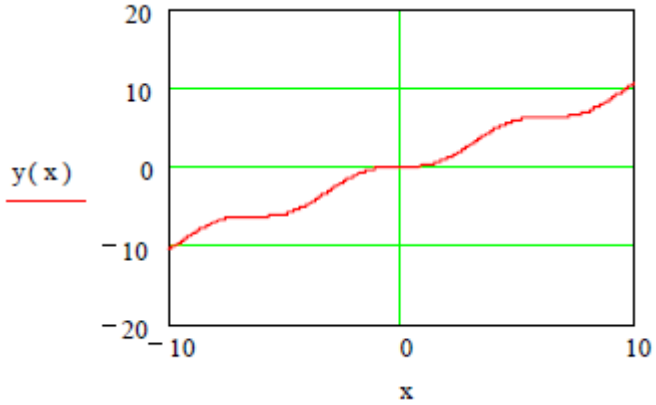


Рисунок 3.3.1 –Графическое отделение корней

Вычисляем значения аргумента и функции (рисунок 3.3.2):

Набираем i, x_i F_i. Ниже, x= и рядом щелкаем мышью, набираем F=, также рядом щелкаем мышью.

$$i := 0..10 \qquad x_i := -5 + i \qquad F_i := y(x_i)$$

x =

	0
0	-5
1	-4
2	-3
3	-2
4	-1
5	0
6	1
7	2
8	3
9	4
10	5

F =

	0
0	-6.209
1	-5.007
2	-3.109
3	-1.341
4	-0.409
5	-0.25
6	-0.091
7	0.841
8	2.609
9	4.507
10	5.709

Рисунок 3.3.2 – Вычисление значения аргумента и функции

Решение уравнения с использованием операторов given, find

Given

$x - \sin(x)0.25 = 0$

$\text{Find}(x) \rightarrow 1.1712296525016659939$

Символьное решение (рисунок 3.3.3):

$x - \sin(x)0.25 \text{ solve}, x \rightarrow 1.1712296525016659939$

$i := 0..10$ $\quad x_0 := 1$ $\quad x_{i+1} := \sin(x_i) + 0.25$

	0
0	1
1	1.091471
2	1.137306
3	1.157505
4	1.165804
5	1.169105
6	1.170401
7	1.170907
8	1.171104
9	1.171181
10	1.171211
11	1.171222

$i := 0..10$ $\quad x_0 := 1$ $\quad x_{i+1} := x_i - \frac{\left[x_i - \left(\sin(x_i) + 0.25\right)\right]}{1 + \cos(x_i)}$

	0
0	1
1	1.059385
2	1.101462
3	1.129285
4	1.146676
5	1.157108
6	1.163197
7	1.16669
8	1.168674
9	1.169794
10	1.170424
11	1.170778

$i := 0..10$ $\quad x_0 := 1$ $\quad x_{i+1} := \frac{\left[x_0 \cdot \left(x_1 - \sin(x_1) - 0.25\right) - x_1 \left(x_0 - \sin(x_0) - 0.25\right)\right]}{\left(x_1 - \sin(x_1) - 0.25\right) - \left(x_0 - \sin(x_0) - 0.25\right)}$

	0
0	1
1	0
2	1.576998
3	1.126117
4	1.177917
5	1.170273
6	1.171367
7	1.17121
8	1.171232
9	1.171229
10	1.17123
11	1.17123

Рисунок 3.3.3 – Символьное решение уравнения различными методами

Слева решение методом итераций, посредине методом касательных, справа методом хорд.

3.3.7 Решение нелинейного уравнения применением функции root

Для простейших уравнений вида f(x)=0 решение в MathCad находится с помощью функции *root* (рисунок 3.3.4).

root (f (x1, x2, …), x1, a, b)

Возвращает значение *x*1, принадлежащее отрезку [a, b], при котором выражение или функция f(x) обращается в 0. Оба аргумента этой функции должны быть скалярами. Функция возвращает скаляр.

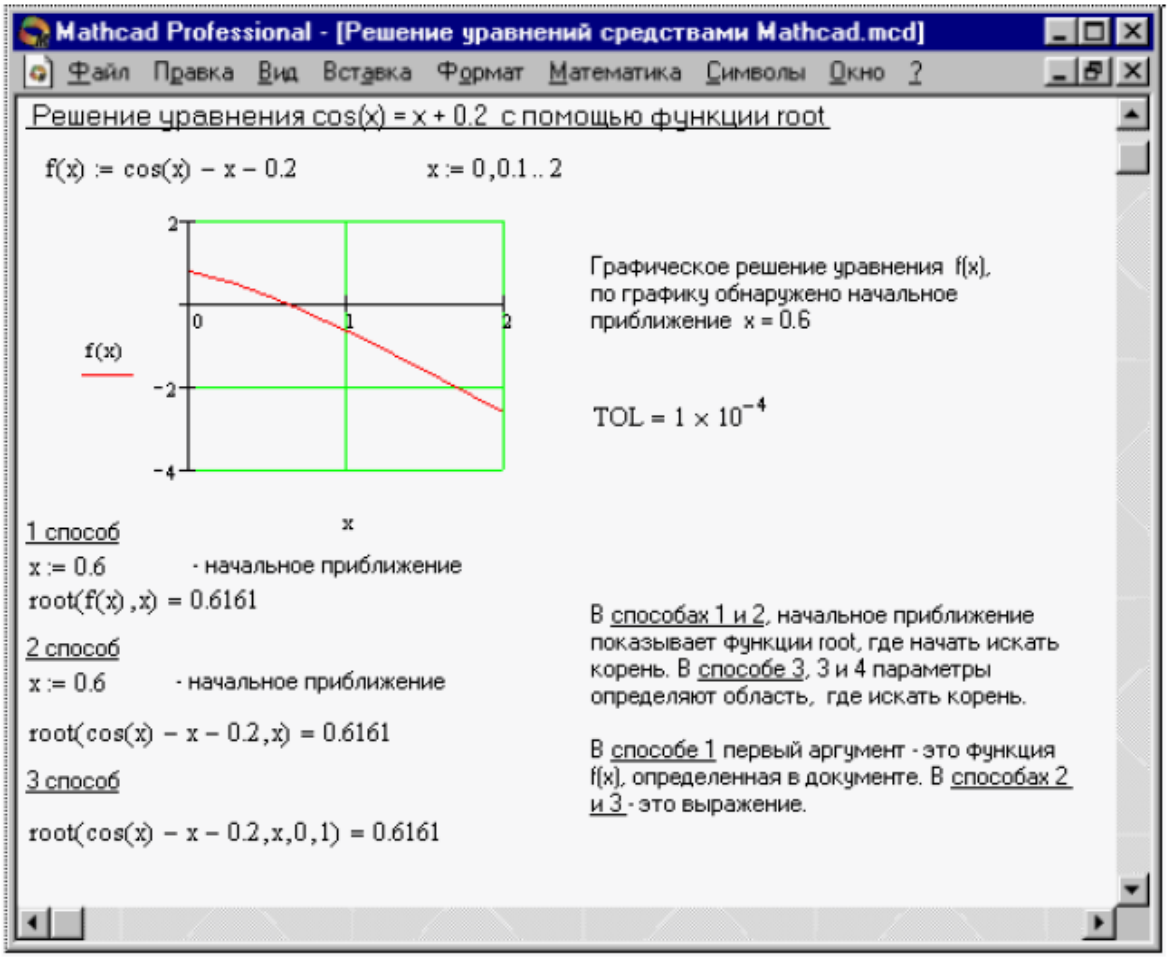


Рисунок 3.3.3 – Решение уравнений с помощью функции root

Аргументы:

***f*(*x*1, *x*2, …)** - функция, определенная где-либо в рабочем документе, или выражение. Выражение должно возвращать скалярные значения.

***x*1** - имя переменной, которая используется в выражении. Этой переменной перед использованием функции *root* необходимо присвоить числовое значение. MathCad использует его как начальное приближение при поиске корня.

a, b - необязательны, если используются, то должны быть вещественными числами, причем ***a* <*b*.**

Приближенные значения корней (начальные приближения) могут быть:

1. Известны из физического смысла задачи.

2. Известны из решения аналогичной задачи при других исходных данных.

3. Найдены графическим способом.

Наиболее распространен графический способ определения начальных приближений. Принимая во внимание, что действительные корни уравнения f(x)=0 - это точки пересечения графика функции f(x) с осью абсцисс, достаточно построить график функции f(x) и отметить точки пересечения f(x) с осью O_x, или отметить на оси O_x отрезки, содержащие по одному корню. Построение графиков часто удается сильно упростить, заменив уравнение f(x)=0 равносильным ему уравнением:

$$f_1(x)=f_2(x),$$

где функции $f_1(x)$ и $f_2(x)$ - более простые, чем функция f(x). Тогда, построив графики функций $y=f_1(x)$ и $y=f_2(x)$, искомые корни получим как абсциссы точек пересечения этих графиков.

Пример. Графически отделить корни уравнения: xlgx=1.

Указанное уравнение удобно переписать в виде равенства:

$$\lg x = \frac{1}{x}.$$

Отсюда ясно, что корни уравнения могут быть найдены как абсциссы точек пересечения логарифмической кривой y=lgx и

гиперболы $y = \frac{1}{x}$ (рисунок 3.3.4). Построив эти кривые, приближенно найдем единственный корень уравнения как $\xi \cong 2{,}5$ или определим его содержащий отрезок [2, 3].

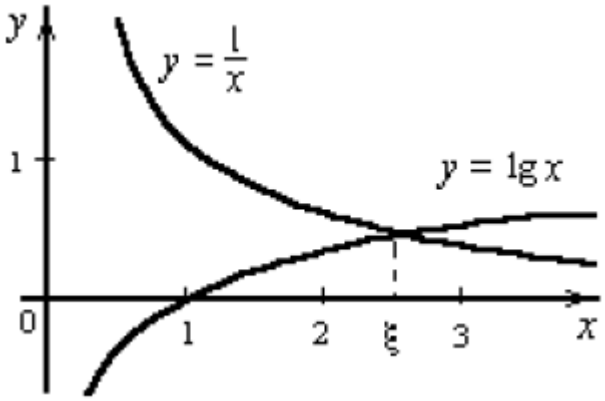


Рисунок 3.3.4 – Нахождение корня уравнения на пересечении кривых

Отсутствие сходимости функции root

Если после многих итераций MathCad не находит подходящего приближения, то появится сообщение Can't converge to a solution. (отсутствует сходимость). Эта ошибка может быть вызвана следующими причинами:

- Уравнение не имеет корней.
- Корни уравнения расположены далеко от начального приближения.
- Выражение имеет локальные max и min между начальным приближением и корнями.
- Выражение имеет разрывы между начальными приближениями и корнями.
- Выражение имеет комплексный корень, но начальное приближение было вещественным.

Чтобы установить причину ошибки, исследуйте график f(x). Он поможет выяснить наличие корней уравнения f(x)=0 и, если они есть, то определить приблизительно их значения. Чем точнее выбрано начальное приближение корня, тем быстрее будет *root* сходиться.

Рекомендации по использованию функции root

- Для изменения точности, с которой функция *root* ищет корень, нужно изменить значение системной переменной **TOL**. Если значение **TOL** увеличивается, функция *root* будет сходиться быстрее, но ответ будет менее точен. Если значение **TOL** уменьшается, то функция *root* будет сходиться медленнее, но ответ будет более точен. Чтобы изменить значение **TOL** в определенной точке рабочего документа, используйте определение вида TOL:=0.01. Чтобы изменить значение **TOL** для всего рабочего документа, выберите команду **Математика → Параметры… → Переменные → Допуск сходимости (TOL)**.
- Если два корня расположены близко друг от друга, следует уменьшить **TOL**, чтобы различить их.
- Если функция f(x) имеет малый наклон около искомого корня, функция *root* (*f*(*x*), *x*) может сходиться к значению r, отстоящему от корня достаточно далеко. В таких случаях для нахождения более точного значения корня необходимо уменьшить значение **TOL**. Другой вариант заключается в замене уравнения f(x)=0 на g(x)=0:

$$g(x) = \frac{f(x)}{\frac{d}{dx} f(x)}.$$

- Для выражения f(x) с известным корнем a нахождение дополнительных корней f(x) эквивалентно поиску корней уравнения h(x)=f(x)/(x-a). Подобный прием полезен для нахождения корней, расположенных близко друг к другу. Проще искать корень выражения h(x), чем пробовать искать другой корень уравнения f(x)=0, выбирая различные начальные приближения.

3.3.8 Вычисление корней полинома применением функции polyroots

Для нахождения корней выражения, имеющего вид

$$v_n x^n + \ldots + v_2 x^2 + v_1 x + v_0$$

лучше использовать функцию *polyroots*, нежели *root.* В отличие от функции *root,* функция *polyroots* не требует начального приближения и возвращает сразу все корни, как вещественные, так и комплексные (рисунок 3.3.5).

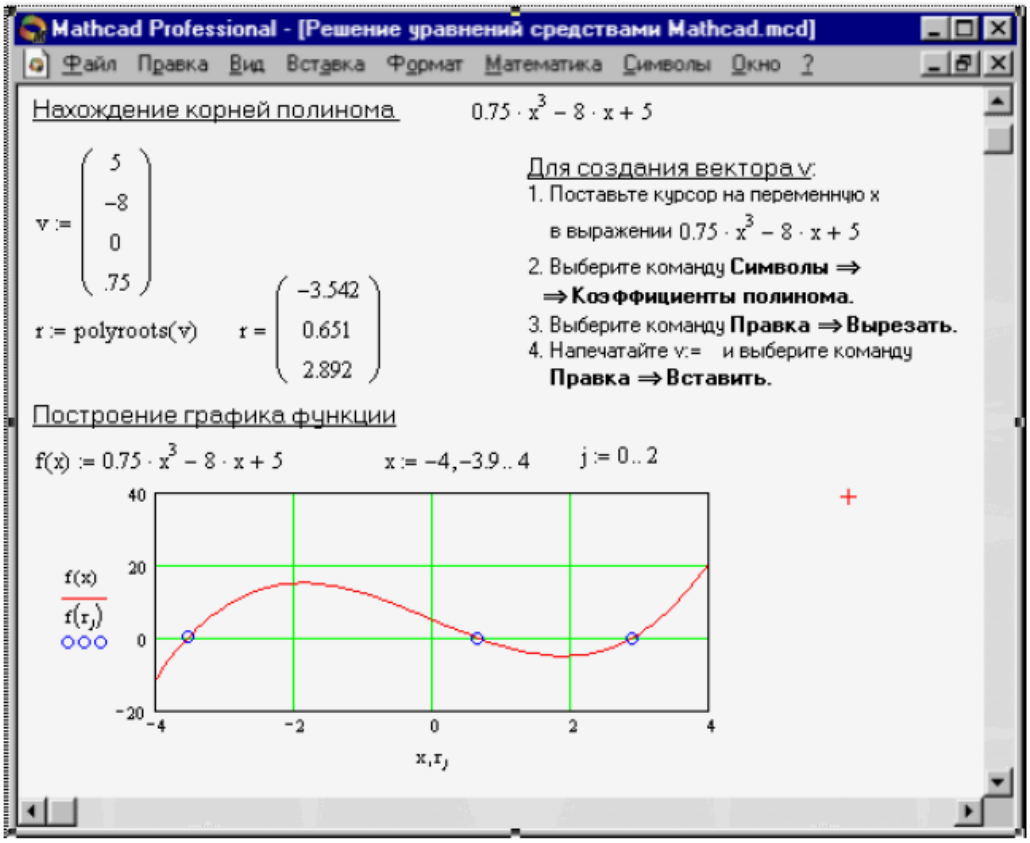


Рисунок 3.3.5 – Нахождение корней с помощью функции *polyroots*

Polyroots(*v*)

Возвращает корни полинома степени n. Коэффициенты полинома находятся в векторе ***v*** длины n+1. Возвращает вектор длины n, состоящий из корней полинома.

Аргументы:

v **-** вектор, содержащий коэффициенты полинома.

Вектор ***v*** удобно создавать используя команду **Символы → Коэффициенты полинома**. Рисунок 3.3.5 иллюстрирует определение корней полинома средствами MathCad.

3.3.9 Решение систем нелинейных уравнений применением функции Find

MathCad позволяет решать системы уравнений (максимальное число уравнений равно 50). Результатом решения системы будет численное значение искомого корня. Для решения системы уравнений необходимо выполнить следующее:

- Задать начальное приближение для всех неизвестных, входящих в систему уравнений. MathCad решает систему с помощью итерационных методов.
- Напечатать ключевое слово *Given*. Оно указывает MathCad, что далее следует система уравнений.
- Введите уравнения и неравенства в любом порядке. Используйте **[Ctrl]=** для печати символа =. Между левыми и правыми частями неравенств может стоять любой из символов <, >, и.

Введите любое выражение, которое включает функцию *Find,* например:

a:=Find(x, y).

Функция Find(z1, z2, ...)

Возвращает точное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Ключевое слово *Given*, уравнения и неравенства, которые следуют за ним, и какое-либо выражение, содержащее функцию *Find*, называют блоком решения уравнений.

Следующие выражения недопустимы внутри блока решения:

- Ограничения со знаком ≠.
- Дискретный аргумент или выражения, содержащие дискретный аргумент в любой форме.
- Неравенства вида a <b<c.

Блоки решения уравнений не могут быть вложены друг в друга, каждый блок может иметь только одно ключевое слово *Given* и имя функции *Find*.

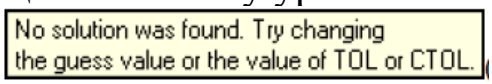
Функция, которая завершает блок решения уравнений, может быть использована аналогично любой другой функции. Можно произвести с ней следующие три действия:

- Можно вывести найденное решение, напечатав выражение вида: *Find*(*var* , *var*2,…)=.
- Определить переменную с помощью функции *Find*:a:= Find(x) - скаляр, var := Find(var1, var2,…) - вектор.

Это удобно сделать, если требуется использовать решение системы уравнений в другом месте рабочего документа.

Определить другую функцию с помощью *Find* как f(a, b, c, …):= Find(x, y, z, …).

Эта конструкция удобна для многократного решения системы уравнений для различных значений некоторых параметров a, b, c,…, непосредственно входящих в систему уравнений.

Сообщение об ошибке 

(Решение не найдено) при решении уравнений появляется, когда:

- Поставленная задача может не иметь решения.
- Для уравнения, которое не имеет вещественных решений, в качестве начального приближения взято вещественное число и наоборот.
- В процессе поиска решения последовательность приближений попала в точку локального минимума невязки. Для поиска искомого решения нужно задать различные начальные приближения.
- Возможно, поставленная задача не может быть решена с заданной точностью. Попробуйте увеличить значение **TOL**.

Рисунок 3.3.6, пример 1 иллюстрирует решение системы уравнений в MathCad.

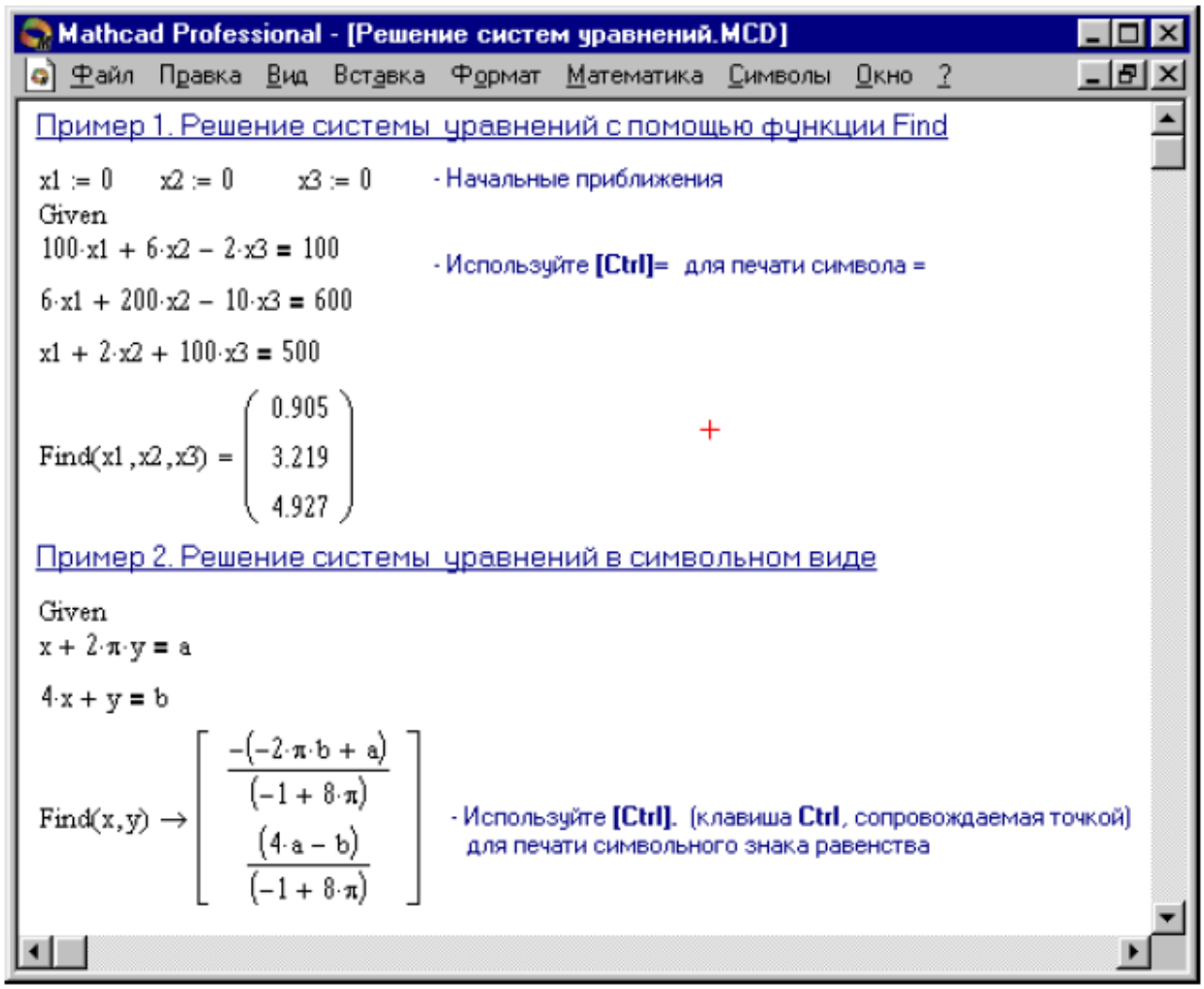


Рисунок 3.3.6 - Решение системы уравнений в MathCad

3.3.10 Приближенные решения уравнений применением функции Minerr

Функция *Minerr* очень похожа на функцию *Find* (использует тот же алгоритм). Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, *Minerr* возвращает это приближение. Функция *Find* в этом случае возвращает сообщение об ошибке. Правила использования функции *Minerr* такие же, как и функции *Find.*

Minerr(*z*1 , *z*2 , ...)

Возвращает приближенное решение системы уравнений. Число аргументов должно быть равно числу неизвестных. Если

Minerr используется в блоке решения уравнений, необходимо всегда включать дополнительную проверку достоверности результатов.

3.3.11 Символьное решение уравнений в MathCad

В MathCad можно быстро и точно найти численное значение корня с помощью функции *root*. Но имеются некоторые задачи, для которых возможности MathCad позволяют находить решения в символьном (аналитическом) виде.

Решение уравнений в символьном виде позволяет найти точные или приближенные корни уравнения:

- Если решаемое уравнение имеет параметр, то решение в символьном виде может выразить искомый корень непосредственно через параметр. Поэтому вместо того, чтобы решать уравнение для каждого нового значения параметра, можно просто заменять его значение в найденном символьном решении.
- Если нужно найти все комплексные корни полинома со степенью меньше или равной 4, символьное решение даст их точные значения в одном векторе или в аналитическом или цифровом виде.

Команда **Символы → Переменные → Вычислить** позволяет решить уравнение относительно некоторой переменной и выразить его корни через остальные параметры уравнения.

Чтобы решить уравнение символьно необходимо:

- Напечатать выражение (для ввода знака равенства используйте комбинацию клавиш **[Ctrl]=**).
- Выделить переменную, относительно которой нужно решить уравнение, щелкнув на ней мышью.
- Выбрать пункт меню **Символы → Переменные → Вычислить.**

Нет необходимости приравнивать выражение нулю. Если MathCad не находит знака равенства, он предполагает, что требуется приравнять выражение нулю.

Чтобы решить систему уравнений в символьном виде, необходимо выполнить следующее:

- Напечатать ключевое слово *Given*.

- Напечатать уравнения в любом порядке ниже слова *Given*. Удостоверьтесь, что для ввода знака = используется **[Ctrl]=**.
- Напечатать функцию *Find*, соответствующую системе уравнений.
- Нажать **[Ctrl].** (клавиша CTRL, сопровождаемая точкой). MathCad отобразит символьный знак равенства ⩳.
- Щелкнуть мышью на функции *Find*.

Рисунок 3.3.6, пример 2 иллюстрирует символьное решение системы уравнений в MathCad.

3.3.12 Решение систем линейных алгебраических уравнений (СЛАУ)

Методы решения систем линейных уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ ..... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_n \end{cases} \qquad (3.3.15)$$

или в векторном виде

$$Ax=b, \qquad (3.3.16)$$

можно разделить на две основные группы:

- прямые методы;
- итерационные методы.

Здесь через A, x, и b соответственно обозначены:

$$A = \begin{vmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{vmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{bmatrix}.$$

Матрица A, столбцами которой являются коэффициенты при соответствующих неизвестных, а строками - коэффициенты при неизвестных в соответствующем уравнении, называется матрицей системы; матрица-столбец b, элементами которой являются правые части уравнений системы, называется матрицей правой части или просто правой частью системы. Матрица-столбец x, элементы которой - искомые неизвестные, называется решением систем*ы*.

Если матрица A - неособенная, то есть $\det A \neq 0$ то система (3.3.15), или эквивалентное ей матричное уравнение (3.3.16), имеет единственное решение.

В самом деле, при условии $\det A \neq 0$ существует обратная матрица A^{-1}. Умножая обе части уравнения (3.3.16) на матрицу A^{-1} получим:

$$A^{-1}Ax = A^{-1}b$$

$$x = A^{-1}b. \qquad (3.3.17)$$

Формула (3.3.17) порождает решение матричного уравнения (3.3.16) и оно единственно.

Прямые методы дают точное решение за конечное число операций. К прямым методам относятся, например:

- метод обратной матрицы;
- метод Крамера;
- метод Гаусса;
- др. методы.

Итерационные методы дают решение системы уравнений как предел последовательных приближений. Для итерационных методов необходимо выполнение условий сходимости и дополнительных преобразований системы в эквивалентную ей.

В качестве иллюстративного примера рассмотрим решение следующей системы линейных алгебраических уравнений:

$$x_1 + 2x_2 + 3x_3 = 7,$$
$$x_1 - 3x_2 + 2x_3 = 5,$$
$$x_1 + x_2 + x_3 = 3.$$

3.3.13 Решение системы линейных алгебраических уравнений методом обратной матрицы

В этом случае алгоритм вычислений состоит в следующем:

- установить режим автоматических вычислений;
- ввести матрицу системы и матрицу-столбец правых частей;
- вычислить решение системы по формуле

$$x = A^{-1}b;$$

- проверить правильность решения умножением матрицы исходной системы уравнений на вектор-столбец решения.

Выполнив указанные действия, можно получить следующее решение (рисунок 3.3.7):

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \\ 1 & 1 & 1 \end{bmatrix} \qquad b := \begin{bmatrix} 7 \\ 5 \\ 3 \end{bmatrix}$$

$$x := A^{-1} \cdot b \qquad x = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \qquad A \cdot x - b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Рисунок 3.3.7 - Решение системы линейных алгебраических уравнений методом обратной матрицы

Таким образом, искомое решение порождается следующим вектор-столбцом:

$$x = [1 \;\; 0 \;\; 2]^T.$$

3.3.14 Решение системы линейных алгебраических уравнений методом Гаусса

Функция ***augment(A,b)*** формирует расширенную матрицу системы добавлением к матрице системы справа столбца правых частей.

Функция ***rref*** приводит расширенную матрицу системы к ступенчатому виду, выполняя прямой и обратный ходы гауссова исключения. Последний столбец содержит решение системы.

$$\mathrm{rref}(\mathrm{augment}(A, b)) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}.$$

Таким образом, искомое решение порождается тем же вектор-столбцом: $x = [1 \;\; 0 \;\; 2]^T$.

3.3.15 Решение системы линейных алгебраических уравнений методом Крамера

Алгоритм решения системы линейных алгебраических уравнений методом Крамера состоит в следующем:

- Вычислить определитель матрицы A и обозначить его через D.
- Определить матрицу A1, заменой первого столбца матрицы A, матрицей b и вычислить определитель матрицы A1 как Dx1.
- Определить матрицу A2, заменой второго столбца матрицы A, матрицей b и вычислить определитель матрицы A2 как Dx2.
- Определить матрицу A3, заменой третьего столбца матрицы A, матрицей b и вычислить определитель матрицы A3 как Dx3.
- Вычислить решение системы линейных уравнений как

x_1=Dx1/D; x_2=Dx2/D ; x_3=Dx3/D.

Реализация указанного алгоритма на MathCad порождает следующее решениеи(рисунок 3.3.8):

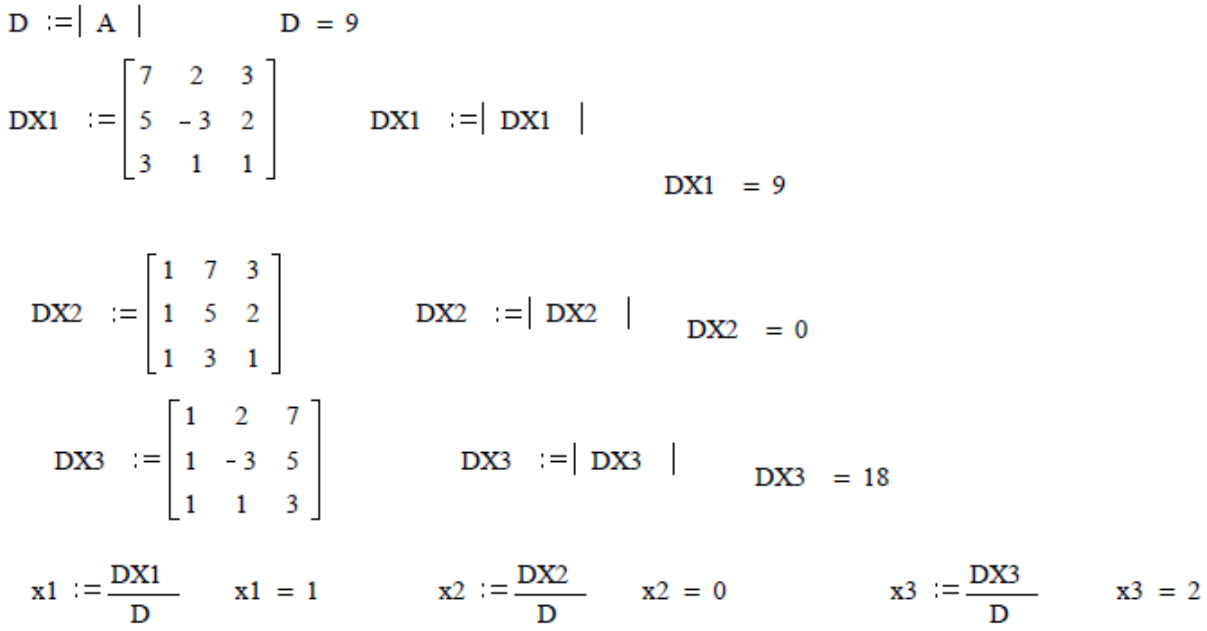


Рисунок 3.3.8 – Реализация алгоритма решения системы линейных алгебраических уравнений методом Крамера в системе MathCad

Как и следовало ожидать, искомое решение порождается тем же вектор-столбцом:

$$x = [1 \;\; 0 \;\; 2]^T$$

3.3.16 Символьное решение систем линейных алгебраических уравнений

Фрагмент рабочего документа с соответствующими вычислениями представлен на рисунке 3.3.9.

Здесь ≡ - логическое равенство.

Легко видеть, что полученное решение в точности совпадает с предыдущим.

Given

$$x1 + 2 \cdot x2 + 3 \cdot x3 = 7$$

$$x1 - 3 \cdot x2 + 2 \cdot x3 = 5$$

$$x1 + x2 + x3 = 3$$

$$\text{Find}\ (x1, x2, x3) \rightarrow \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

Рисунок 3.3.9 – Скриншот фрагмента рабочего документа с символьным решением систем линейных алгебраических уравнений

3.3.17 Решение системы линейных алгебраических уравнений применением функции lsolve

Системы линейных уравнений удобно решать с помощью функции *lsolve*. **lsolve(** ***A*****,** ***b*****)**. В этом случае возвращается вектор решения *x* такой, что

$$Ax=b.$$

На рисунке 3.3.10 в качестве иллюстрации показано решение системы трех линейных уравнений относительно трех неизвестных.

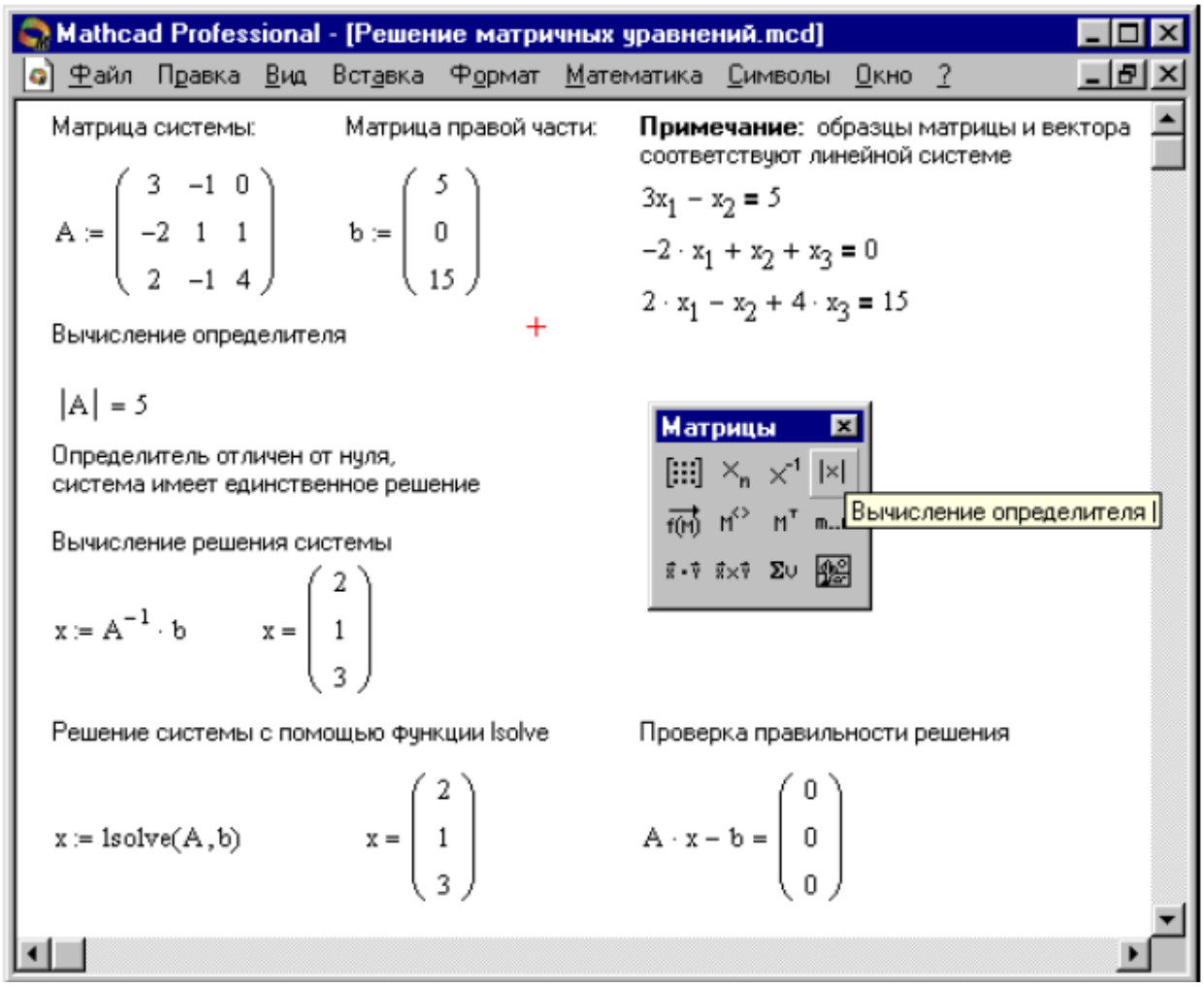


Рисунок 3.3.10 - Решение системы трех линейных уравнений относительно трех неизвестных

Аналогично, для исследуемой системы уравнений реализация на MathCad приводит к следующему решению (рисунок 3.3.11):

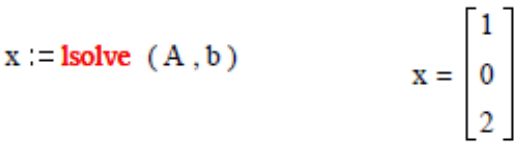


Рисунок 3.3.11 - Скриншот фрагмента рабочего документа исследуемой системы уравнений, реализованного в системе MathCad

Легко видеть, что и в этом случае полученное решение в точности совпадает со всеми предыдущими.

3.3.18 Решение системы линейных алгебраических уравнений методом простых итераций

Предполагается в дальнейшем, что матрица A квадратная и невырожденная. Предварительно систему (3.3.16) приводится к итерационному виду:

$$x=Cx+f. \qquad (3.3.18)$$

Известно, что для произвольного начального вектора x0 итерационный процесс:

$$x^{n+1}=Cx^{n}+f, \qquad (3.3.19)$$

сходится, если выполнено одно из условий:

$$\text{а)}\sum_{j=1}^{n}\left|c_{i,j}\right|=\alpha<1,\ 1\le i\le n, \qquad (3.3.20)$$

$$\text{б)}\sum_{i=1}^{n}\left|c_{i,j}\right|=\alpha<1,\ 1\le j\le n, \qquad (3.3.21)$$

$$\text{в)}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}^{2}}=\alpha<1. \qquad (3.3.22)$$

Процесс вычислений заканчиваем при выполнении условия:

$$p_i(x^{k-1},x^{k})\le\varepsilon(1-\alpha)/\alpha \qquad (3.3.23)$$

где ε − заданная точность (например, $\varepsilon=10^{-4}$); ρ (i=1,2,3) − одна из метрик, определяемая левой частью (3.3.20)−(3.3.22), по которой была установлена сходимость.

Алгоритм решения системы линейных алгебраических уравнений методом простых итераций состоит в следующем:

1. Ввести матрицы C и d.

2. Преобразовать известными методами исходную систему Cx=d к виду x = b + Ax.

3. Определить нулевое приближение решения.

4. Задать количество необходимых итераций.

5. Вычислить последовательные приближения.

Реализация алгоритма решения системы линейных алгебраических уравнений методом простых итераций в системе MathCad представлена на рисунке 3.3.12.

$$ORIGIN := 1$$

$$C := \begin{bmatrix} 100 & 6 & -2 \\ 6 & 200 & -10 \\ 1 & 2 & 100 \end{bmatrix} \qquad d := \begin{bmatrix} 200 \\ 600 \\ 500 \end{bmatrix}$$

$$i := 1..3 \qquad j := 1..3$$

$$b_i := \frac{d_i}{C_{i,i}} \qquad A_{i,j} := \frac{-C_{i,j}}{C_{i,i}} \qquad A_{i,i} := 0$$

$$A = \begin{bmatrix} 0 & -0.06 & 0.02 \\ -0.03 & 0 & 0.05 \\ -0.01 & -0.02 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

$$x^{\langle 1 \rangle} := b \qquad k := 2..10 \qquad x^{\langle k \rangle} := b + A \cdot x^{\langle k-1 \rangle}$$

$x =$

	1	2	3	4	5	6	7	8	9	10
1	2	1.92	1.907	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	3.189
3	5	4.92	4.917	4.917	4.917	4.917	4.917	4.917	4.917	4.917

$$X := x^{\langle 10 \rangle} \qquad X = \begin{bmatrix} 1.907 \\ 3.189 \\ 4.917 \end{bmatrix}$$

Рисунок 3.3.12 - Реализация алгоритма решения системы линейных алгебраических уравнений методом простых итераций в системе MathCad

3.3.19 Решение системы линейных алгебраических уравнений методом Зейделя

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. Вычисления ведутся по формуле

$$x_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_i^{(k+1)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_i^{k} + \frac{b_i}{a_{ii}} \qquad (3.3.24)$$

Каждое из условий (3.3.20)−(3.3.22) является, при этом, достаточным для сходимости итерационного процесса по методу Зейделя. Практически же удобнее следующее преобразование, а именно, умножив слева обе части (3.3.19) на A^T и определи матрицу C и вектор d как

$$C = A^T A \quad d = A^T b, \qquad (3.3.25)$$

получим эквивалентную ей систему:

$$Cx=d \qquad (3.3.26)$$

Разделив каждое полученное соотношение на c_{ii} , приведем систему к виду (3.3.24). Указанное преобразование гарантирует сходимость итерационного процесса.

Алгоритм решения системы линейных алгебраических уравнений методом Зейделя по форме ничем не отличается от предыдущего и состоит в следующем:

1. Ввести матрицы C и d.
2. Преобразовать известными методами исходную систему Cx=d к виду x=b+Ax.
3. Определить нулевое приближение решения.
4. Задать количество необходимых итераций.
5. Вычислить последовательные приближения.

Реализация алгоритма решения системы линейных алгебраических уравнений методом Зейделя в системе MathCad представлена на рисунке 3.3.13.

$$ORIGIN := 1$$

$$C := \begin{bmatrix} 100 & 6 & -2 \\ 6 & 200 & -10 \\ 1 & 2 & 100 \end{bmatrix} \qquad d := \begin{bmatrix} 200 \\ 600 \\ 500 \end{bmatrix}$$

$$i := 1..3 \qquad b_i := \frac{d_i}{C_{i,i}} \qquad i := 2..3 \qquad j := 1..2$$

$$A1_{i,j} := \frac{-C_{i,j}}{C_{i,i}} \qquad A2_{j,i} := \frac{-C_{j,i}}{C_{j,j}}$$

$$A1_{i,i} := 0 \quad A1_{j,i} := 0 \quad A2_{i,i} := 0 \quad A2_{i,j} := 0 \quad A := A1 + A2$$

$$A1 = \begin{bmatrix} 0 & 0 & 0 \\ -0.03 & 0 & 0 \\ -0.01 & -0.02 & 0 \end{bmatrix} \quad A2 = \begin{bmatrix} 0 & -0.06 & 0.02 \\ 0 & 0 & 0.05 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & -0.06 & 0.02 \\ -0.03 & 0 & 0.05 \\ -0.01 & -0.02 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

$$x^{\langle 1 \rangle} := b \qquad y^{\langle 1 \rangle} := b \qquad k := 2..10$$

$$x^{\langle k \rangle} := b + A2 \cdot x^{\langle k-1 \rangle} \qquad x^{\langle k \rangle} := x^{\langle k \rangle} + A1 \cdot x^{\langle k-1 \rangle}$$

$x =$

	1	2	3	4	5	6	7	8	9	10
1	2	1.92	1.905	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	3.193
3	5	4.92	4.917	4.917	4.917	4.917	4.917	4.917	4.917	4.917

Рисунок 3.3.13 – Реализация алгоритма решения системы линейных алгебраических уравнений методом Зейделя в системе MathCad

3.3.20 Порядок выполнения работы

Упражнение 3.3.1

Построить график функции f(x) (таблица 3.3.1) и приблизительно определить один из корней уравнения. Решить уравнение f(x)=0 с точностью $\varepsilon = 10^{-4}$ с помощью встроенной функции MathCad *root*;

Таблица 3.3.1– Варианты заданий

№ вар.	f(x)	№ вар.	f(x)
1	$e^{x-1} - x^3 - x$ $x \in [0, 1]$	9	$0.25x^3 + x - 2$ $x \in [0, 2]$
2	$x - \frac{1}{3 + \sin(3.6x)}$ $x \in [0, 1]$	10	$\arccos\frac{1 - x^2}{1 + x^2} - x$ $x \in [2, 3]$
3	$\arccos x - \sqrt{1 - 0.3x^3}$ $x \in [0, 1]$	11	$3x - 4\ln x - 5$ $x \in [2, 4]$
4	$\sqrt{1 - 0.4x^2} - \arcsin x$ $x \in [0, 1]$	12	$e^x - e^{-x} - 2$ $x \in [0, 1]$
5	$3x - 14 + e^x - e^{-x}$ $x \in [1, 3]$	13	$\sqrt{1 - x} - tgx$ $x \in [0, 1]$
6	$\sqrt{x^2 + 1.2 - \cos x} - 1$ $x \in [0, 1]$	14	$1 - x + \sin x - \ln(1 + x)$ $x \in [0, 2]$
7	$\cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x}$ $x \in [1, 2]$	15	$x^5 - x - 0.2$ $x \in [1, 2]$
8	$0.1x^2 - x\ln x$ $x \in [1, 2]$		

Упражнение 3.3.2

Выполнить отделение корней: графически и по программе (точность $\varepsilon = 10^{-1}$). Индивидуальные задания приведены в таблице 3.3.2.

Таблица 3.3.2 – Индивидуальные задания

№ вар.	Функция	№ вар.	Функция
1.	$x + x\ln(x+0.5) - 0.5 = 0$	2.	$x2^x - 1 = 0$
3.	$x^3 - 2x^2 + x - 3 = 0$	4.	$x^3 + 12x - 2 = 0$
5.	$5x - 8\ln(x) - 8 = 0$	6.	$x - \sin(x) + 0.25 = 0$
7.	$x^3 - 6x^2 + 20 = 0$	8.	$5x^3 + 10x^2 + 5x - 1 = 0$
9.	$0.1x^2 - x\ln(x) = 0$	10.	$x^4 + 0.5x^3 - 4x^2 - 3x + 0.5 = 0$

Упражнение 3.3.3

Для полинома g(x) выполнить следующие действия (таблица 3.3.3):

1. с помощью команды **Символы → Коэффициенты полинома** создать вектор V, содержащий коэффициенты полинома;
2. решить уравнение g(x)=0 с помощью функции *polyroots*;
3. решить уравнение символьно, используя команду **Символы → Переменные → Вычислить**.

Таблица 3.3.3 – Индивидуальные задания

№ вар.	$g(x)$	№ вар.	$g(x)$
1	$x^4 - 2x^3 + x^2 - 12x + 20$	9	$x^4 + x^3 - 17x^2 - 45x - 100$
2	$x^4 + 6x^3 + x^2 - 4x - 60$	10	$x^4 - 5x^3 + x^2 - 15x + 50$
3	$x^4 - 14x^2 - 40x - 75$	11	$x^4 - 4x^3 - 2x^2 - 20x + 25$
4	$x^4 - x^3 + x^2 - 11x + 10$	12	$x^4 + 5x^3 + 7x^2 - 5x + 100$
5	$x^4 - x^3 - 29x^2 - 71x - 140$	13	$x^4 - 7x^3 + 7x^2 - 5x + 100$
6	$x^4 + 7x^3 + 9x^2 + 13x - 30$	14	$x^4 + 10x^3 + 36x^2 + 704x + 75$
7	$x^4 + 3x^3 - 23x^2 - 55x - 150$	15	$x^4 + 9x^3 + 31x^2 + 594x + 60$
8	$x^4 - 6x^3 + 4x^2 + 10x + 75$		

Упражнение 3.3.4

Решить систему линейных уравнений (таблица 3.3.4):
1) используя функцию *Find*;
2) матричным способом и используя функцию *lsolve*.

Таблица 3.3.4 – Индивидальные задания

№ вар.	Система линейных уравнений	№ вар.	Система линейных уравнений
1	2	3	4
1	$\begin{cases} 2x_1 + x_2 + 2x_3 + 3x_4 = 8 \\ 3x_1 + 3x_2 = 6 \\ 2x_1 - x_2 + 3x_4 = 4 \\ x_1 + 2x_2 - x_3 + 2x_4 = 4 \end{cases}$	9	$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = -4 \\ x_1 - 3x_2 - 6x_4 = -7 \\ 2x_2 - x_3 + 2x_4 = 2 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -2 \end{cases}$
2	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 22 \\ 2x_1 + 3x_2 + x_3 + 2x_4 = 17 \\ x_1 + x_2 + x_3 - x_4 = 8 \\ x_1 - 2x_3 - 3x_4 = -7 \end{cases}$	10	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$
3	$\begin{cases} 9x_1 + 10x_2 - 7x_3 - x_4 = 23 \\ 7x_1 - x_3 + 5x_4 = 37 \\ 5x_1 - 2x_3 + x_4 = 22 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$	11	$\begin{cases} 2x_1 - 8x_2 - 3x_3 - 2x_4 = -18 \\ x_1 - 2x_2 + 3x_3 - 2x_4 = 28 \\ x_2 + x_3 + x_4 = 10 \\ 11x_1 + x_3 + 2x_4 = 21 \end{cases}$
4	$\begin{cases} 6x_1 - x_2 + 10x_3 - x_4 = 158 \\ 2x_1 + x_2 + 10x_3 + 7x_4 = 128 \\ 3x_1 - 2x_2 - 2x_3 - x_4 = 7 \\ x_1 - 12x_2 + 2x_3 - x_4 = 17 \end{cases}$	12	$\begin{cases} 2x_1 - x_2 + 4x_3 + x_4 = 66 \\ 2x_2 - 6x_3 + x_4 = -63 \\ 8x_1 - 3x_2 + 6x_3 - 5x_4 = 146 \\ 2x_1 - 7x_2 + 6x_3 - x_4 = 80 \end{cases}$
5	$\begin{cases} x_1 - 2x_2 + 6x_3 + x_4 = 88 \\ 5x_1 + 2x_3 - 3x_4 = 88 \\ 7x_1 - 3x_2 + 7x_3 + 2x_4 = 181 \\ x_1 - 12x_2 + 2x_3 - x_4 = 17 \end{cases}$	13	$\begin{cases} 2x_1 - 3x_3 - 2x_4 = -16 \\ 2x_1 - x_2 + 13x_3 + 4x_4 = 213 \\ 3x_1 + x_2 + 2x_3 + x_4 = 72 \\ x_1 - 12x_3 - 5x_4 = -159 \end{cases}$

Продолжение таблицы 3.3.4

1	2	3	4
6	$\begin{cases} x_1 - 2x_2 - 8x_4 = -7 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -8 \\ x_1 + x_2 - 5x_3 + x_4 = -10 \\ 2x_1 - x_2 + 2x_4 = 7 \end{cases}$	14	$\begin{cases} 7x_1 + 7x_2 - 7x_3 - 2x_4 = 5 \\ 3x_1 + 4x_2 + 5x_3 + 8x_4 = 60 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 27 \\ 2x_1 - 2x_3 - x_4 = -1 \end{cases}$
7	$\begin{cases} 2x_1 + 2x_2 + 6x_3 + x_4 = 15 \\ -x_2 + 2x_3 + x_4 = 18 \\ 4x_1 - 3x_2 + x_3 - 5x_4 = 37 \\ 3x_1 - 5x_2 + x_3 - x_4 = 30 \end{cases}$	15	$\begin{cases} 6x_1 - 9x_2 + 5x_3 + x_4 = 124 \\ 7x_2 - 5x_3 - x_4 = -54 \\ 5x_1 - 5x_2 + 2x_3 + 4x_4 = 83 \\ 3x_1 - 9x_2 + x_3 + 6x_4 = 45 \end{cases}$
8	$\begin{cases} 4x_1 - 5x_2 + 7x_3 + 5x_4 = 165 \\ 2x_1 + x_2 - 3x_3 - x_4 = -15 \\ 9x_1 + 4x_3 - x_4 = 194 \\ x_1 - x_2 - 2x_3 - 3x_4 = -19 \end{cases}$		

Упражение 3.3.5

Решить методом Гаусса системы линейных алгебраических уравнений, указанные в таблице 3.3.5.

Таблица 3.3.5 – Индивидуальные задания

№ вар.	a_{1i}	a_{2i}	a_{3i}	b_{1i}
1	2	3	4	5
1	0.35 0.12 -0.13	0.12 0.71 0.15	-0.13 0.15 0.63	0.10 0.26 0.38
2	0.71 0.10 -0.10	0.10 0.34 0.64	0.12 -0.04 0.56	0.29 0.32 -0.10
3	0.34 -0.04 0.06	-0.04 0.44 0.56	0.10 -0.12 0.39	0.33 -0.05 0.28
4	0.10 -0.04 -0.43	-0.04 0.34 0.05	-0.63 0.05 0.13	-0.15 0.31 0.37

Продолжение таблицы 3.3.5

1	2	3	4	5
5	0.63 0.05 0.15	0.05 0.34 0.10	0.15 0.10 0.71	0.34 0.32 0.42
6	1.20 -0.50 -0.30	-0.20 1.70 0.10	0.30 -1.60 -1.50	-0.60 0.30 0.40
7	0.30 -0.10 -1.50	1.20 -0.20 -0.30	-0.20 1.60 0.10	-0.60 0.30 0.70
8	0.20 0.58 0.05	0.44 -0.29 0.34	0.91 0.05 0.10	0.74 0.02 0.32
9	6.36 7.42 1.77	1.75 19.03 0.42	1.0 1.75 6.36	41.70 49.49 27.67
10	3.11 -1.65 0.60	-1.66 3.15 0.78	-0.60 -0.78 -2.97	-0.92 2.57 1.65

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 влияниях 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.

Упражнение 3.3.6

Преобразовать нелинейные уравнения системы из таблицы 3.3.6 к виду

$$f_1(x)=y$$

и

$$f_2(y)=x.$$

Построить их графики и определить начальное приближение решения. Решить систему нелинейных уравнений с помощью функции *Minerr*.

Таблица 3.3.6 – Индивидуальные задания

№ вар.	Система нелинейных уравнений	№ вар.	Система нелинейных уравнений
1	$\begin{cases}\sin x + 2y = 2,\\ \cos(y-1) + x = 0{,}7.\end{cases}$	9	$\begin{cases}\sin y + x = -0{,}4,\\ 2y - \cos(x+1) = 0.\end{cases}$
2	$\begin{cases}\sin(x+0{,}5) - y = 1,\\ \cos(y-2) + x = 0.\end{cases}$	10	$\begin{cases}\sin(x+2) - y = 1{,}5,\\ \cos(y-2) + x = 0{,}5.\end{cases}$
3	$\begin{cases}\cos x + y = 1{,}5,\\ 2x - \sin(y-0{,}5) = 1.\end{cases}$	11	$\begin{cases}\cos(x+0{,}5) - y = 2,\\ \sin y - 2x = 1.\end{cases}$
4	$\begin{cases}\cos(x+0{,}5) + y = 0{,}8\\ \sin y - 2x = 1{,}6.\end{cases}$	12	$\begin{cases}\cos(x-2) + y = 0,\\ \sin(y+0{,}5) - x = 1.\end{cases}$
5	$\begin{cases}\sin(x-1) = 1{,}3 - y,\\ x - \sin(y+1) = 0{,}8.\end{cases}$	13	$\begin{cases}\cos(x+0{,}5) + y = 1,\\ \sin(y+0{,}5) - x = 1.\end{cases}$
6	$\begin{cases}\cos(x+0{,}5) + y = 1,\\ \sin y - 2x = 2.\end{cases}$	14	$\begin{cases}\sin(x) - 2y = 1,\\ \cos(y+0{,}5) - x = 2.\end{cases}$
7	$\begin{cases}-\sin(x+1) + y = 0{,}8,\\ \sin(y-1) + x = 1{,}3.\end{cases}$	15	$\begin{cases}2y - \sin(x-2) = 1,\\ \cos(y) + x = 1{,}5.\end{cases}$
8	$\begin{cases}\sin(x) - 2y = 1,\\ \sin(y-1) + x = 1{,}3.\end{cases}$		

Упражнение 3.3.7

Символьно решить системы уравнений:

$$\begin{cases} 3x + 4\pi y = a, \\ 2x + y = b. \end{cases} \qquad \begin{cases} 2y - \pi z = a, \\ \pi z - z = b, \\ 3y + x = c. \end{cases}$$

3.3.21 Контрольные вопросы

1. Назовите способы нахождения начального приближения.
2. Какие функции для решения одного уравнения в MathCad вы знаете? В чем их отличие?
3. Какие аргументы функции *root* не обязательны?
4. В каких случаях MathCad не может найти корень уравнения?
5. Какая системная переменная отвечает за точность вычислений?
6. Как изменить точность, с которой функция *root* ищет корень?
7. Как системная переменная **TOL** влияет на решение уравнения с помощью функции root?
8. Назовите функции для решения систем уравнений в MathCad и особенности их применения.
9. Опишите структуру блока решения уравнений.
10. Какой знак равенства используется в блоке решения? Какой комбинацией клавиш вставляется в документ?
11. Какие выражения не допустимы внутри блока решения уравнения?
12. Опишите способы использования функции *Find*.
13. В каких случаях MathCad не может найти решение системы уравнений?
14. Дайте сравнительную характеристику функциям *Find* и *Minerr*.
15. Какие уравнения называются матричными?

16. Как решать матричные уравнения? Назовите способы решения матричных уравнений.

17. Этапы решения уравнения с одной неизвестной.

18. Способы отделения корней.

19. Каким образом графическое отделение корней уточняется с помощью вычислений?

20. Дать словесное описание алгоритма метода половинного деления.

21. Необходимые условия сходимости метода половинного деления.

22. Условие окончания счета метода простой итерации. Погрешность метода.

23. Словесное описание алгоритма метода хорд. Графическое представление метода. Вычисление погрешности.

24. Словесное описание алгоритма метода касательных (Ньютона). Графическое представление метода. Условие выбора начальной точки.

25. К какому типу - прямому или итерационному - относится метод Гаусса?

26. В чем заключается прямой и обратный ход в схеме единственного деления?

27. Как организуется, контроль над вычислениями в прямом и обратном ходе?

28. Как строится итерационная последовательность для нахождения решения системы линейных уравнений?

29. Как формулируется достаточные условия сходимости итерационного процесса?

30. Как эти условия связаны с выбором метрики пространства?

31. В чем отличие итерационного процесса метода Зейделя от аналогичного процесса метода простой итерации?

Лабораторная работа №4
Интерполирование функций MathCad

4.1 Постановка задачи интерполирования функций

Пусть функция f(x) задана таблично, либо вычисление ее требует громоздких выкладок. Заменим приближенно функцию f(x) на какую-либо функцию F(x), так, чтобы отклонение f(x) от F(x) было в заданной области в некотором смысле минимальным. Подобная замена называется аппроксимацией функции f(x), а функция F(x) – аппроксимирующей (приближающей) функцией.

Классический подход к решению задачи построения приближающей функции основывается на требование строгого совпадения значений f(x) и F(x) в точках x_i (i = 0, 1, 2, … n), т.е.

$$F(x_0) = y_0,\ \ F(x_1) = y_1, ...,\ \ F(x_n) = y_n. \qquad (4.1)$$

В этом случае нахождение приближенной функции называют интерполяцией (или интерполированием), точки $x_0, x_1, ..., x_n$ – узлами интерполяции.

Часто интерполирование ведется для функций, заданных таблицами с равноотстоящими значениями аргумента x. В этом случае шаг таблицы $h = x_{i+1} - x_i$ (i = 0, 1, 2, …) является величиной постоянной. Для таких таблиц построение интерполяционных формул (как, впрочем, и вычисление по этим формулам) заметно упрощается.

4.2 Порядок выполнения работы

Задание 1

По заданной таблице значений функции составить формулу интерполяционного многочлена Лагранжа (4.2) и построить график $L_2(x)$. Исходные данные берутся из таблицы 4.1.

$$L_2(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + \\ = y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \quad (4.2)$$

Таблица 4.1 - Исходные данные

№	x_0	x_1	x_2	y_0	y_1	y_2
1	2	3	5	4	1	7
2	4	2	3	5	2	8
3	0	2	3	-1	-4	2
4	7	9	13	2	-2	3
5	-3	-1	3	7	-1	4
6	1	2	4	-3	-7	2
7	-2	-1	2	4	9	1
8	2	4	5	9	-3	6
9	-4	-2	0	2	8	5
10	-1	1.5	3	4	-7	1
11	2	4	7	-1	-6	3
12	-9	-7	-4	3	-3	4
13	0	1	4	7	-1	8
14	8	5	0	9	2	4
15	-7	-5	-4	4	-4	5

Задание 2

Вычислить одно значение заданной функции для промежуточного значения аргумента (а) с помощью интерполяционного многочлена Лагранжа (4.3) и оценить погрешность интерполяции. Для выполнения задания исходные данные берутся из таблицы 4.2, 4.3 или 4.4.

$$L_n(x) = \sum_{i=0}^{n} y_i \frac{(x-x_0)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)} \quad (4.3)$$

Для погрешности $R_n(x)$ выполняется неравенство

$$|R_n(x)| \le \frac{M_{n+1}}{(n+1)!}\left|\prod_{n+1}(x)\right|, \quad x \in [x_0, x_n] \qquad (4.3)$$

где $M_{n+1} = \max\left|f^{(n+1)}(x)\right|$.

Таблица 4.2 – Исходные данные

№ варианта	Значение аргумента (a)	№ таблицы
1	-2.0	4.3
2	3.77	4.4
3	0.55	4.3
4	4.83	4.4
5	3.5	4.3
6	5.1	4.4
7	1.75	4.3
8	4.2	4.4
9	-1.55	4.3
10	6.76	4.4

Таблица 4.3 – Исходные данные

x	-3.2	-0.8	0.4	2.8	4.0	6.4	7.6
$f(x) = 2.1\sin(0.37x)$	-1.94	-0.61	0.31	1.81	2.09	1.47	0.68

Таблица 4.4 – Исходные данные

x	1.3	2.1	3.7	4.5	6.1	7.7	8.5
$f(x) = \lg(x)/x + x^2$	1.777	4.563	13.84	20.39	37.34	59.41	72.4

Таблица 4.5 – Исходные данные

x	0.10	0.15	0.20	0.25	0.30	0.35	0.40
$f(x) = \cos(x)$	0.995	0.988	0.980	0.969	0.955	0.939	0.921

Таблица 4.6 – Исходные данные

x	0.65	0.70	0.75	0.80	0.85	0.90	0.95
$f(x) = \sin(x)$	0.605	0.644	0.681	0.71	0.75	0.783	0.813

Задание 3

Уплотнить часть таблицы заданной на отрезке [a, b] функции, используя интерполяционный многочлен Ньютона (4.5) и оценить погрешность интерполяции D (формула (4.6)). Таблицу 4.7 конечных разностей просчитать вручную на отрезке [a, b] с шагом h. Для выполнения задания исходные данные берутся из таблиц 4.8, 4.5 и 4.6.

$$P_2(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_0 \qquad (4.5)$$

где $t = \frac{x - x_0}{h}$

$$D \approx \frac{t(t-1)(t-2)}{3!} f'''(\xi) \qquad (4.6)$$

где ξ – некоторая внутренняя точка наименьшего промежутка, содержащего все узлы $x_i (i = \overline{0,n})$ и x.

Таблица 4.7 – Исходные данные

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
x_0	y_0	$\Delta y_0 = y_1 - y_0$	$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$	$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$
$x_1 = x_0 + h$	y_1	$\Delta y_1 = y_2 - y_1$	$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$	
$x_2 = x_1 + h$	y_2	$\Delta y_2 = y_3 - y_2$		
$x_3 = x_2 + h$	y_3			

Формула (4.5) называется первой интерполяционной формулой Ньютона. Если вычисляемое значение переменной ближе к концу отрезка [a;b], то применяют вторую формулу Ньютона – интерполирование назад (формула (4.7)).

$$P_n(x) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \frac{t(t+1)(t+2)}{3!}\Delta^3 y_{n-3} \qquad (4.7)$$

где $t = \frac{x - x_n}{h}$ и $D = \frac{t(t+1)(t+2)}{3!} f'''(\xi)$

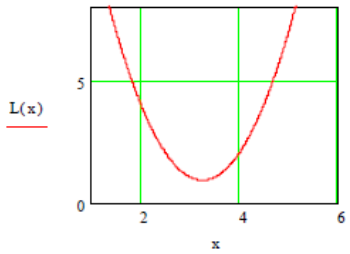
Таблица 4.8 – Исходные данные

№	a	b	h_0	h	№ таблицы
1	0.65	0.80	0.05	0.01	3.6
2	0.25	0.40	0.05	0.025	3.5
3	0.75	0.90	0.05	0.01	3.6
4	0.70	0.85	0.05	0.025	3.6
5	0.80	0.95	0.05	0.025	3.6
6	0.1	0.25	0.05	0.025	3.5
7	0.15	0.3	0.05	0.025	3.5
8	0.7	0.85	0.05	0.025	3.6
9	0.2	0.35	0.05	0.01	3.5
10	0.80	0.95	0.05	0.01	3.6

Примерный фрагмент выполнения работы представлен на рисунке 4.1.

$x0 := 2 \quad x1 := 3 \quad x2 := 5 \quad y0 := 4 \quad y1 := 1 \quad y2 := 7$

$$L(x) := \left[\frac{y0\cdot(x - x1)\cdot((x - x2))}{(x0 - x1)\cdot(x0 - x2)} + \frac{y1\cdot(x - x0)\cdot((x - x2))}{(x1 - x0)\cdot(x1 - x2)} + \frac{y2\cdot(x - x0)\cdot((x - x1))}{(x2 - x0)\cdot(x2 - x1)}\right]$$



$x0 := 2 \quad x1 := 3 \quad x2 := 5 \quad y0 := 4 \quad y1 := 1 \quad y2 := 7$

$$L(x) := \left[\frac{4\cdot(x - 3)\cdot((x - 5))}{(2 - 3)\cdot(2 - 5)} + \frac{1\cdot(x - 2)\cdot((x - 5))}{(3 - 2)\cdot(3 - 5)} + \frac{7\cdot(x - 2)\cdot((x - 3))}{(5 - 2)\cdot(5 - 3)}\right]$$

$2\cdot x^2 - 13\cdot x + 22$

Рисунок 4.1 - Примерный фрагмент выполнения работы

4.3 Контрольные вопросы

1. В чем особенность приближения таблично заданной функции методом интерполирования?
2. Как обосновывается существование и единственность интерполяционного многочлена?
3. Как связана степень интерполяционного многочлена с количеством узлов интерполяции?
4. Как строятся интерполяционные многочлены Лагранжа и Ньютона?
5. В чем особенности этих двух способов интерполяции?
6. Как производится оценка погрешности метода интерполяции многочленом Лагранжа?
7. Как используется метод интерполирования для уточнения таблиц функций?
8. В чем отличие между первой и второй интерполяционными формулами Ньютона?

Лабораторная работа №5

5.1 Интегрирование в пакете MathCad

5.1.1 Постановка задачи приближенного интегрирования функций

Геометрический смысл определенного интеграла функции f(x) заключается в площади фигуры, образованной этой функцией и осью OX.

Формулы, используемые для приближенного вычисления однократных интегралов, называют квадратурными формулами.

5.1.2 Приближенное вычисление интеграла функции методом прямоугольника

Самый простой способ вычислить определенный интеграл от «хорошей» (т.е. гладкой, неосциллирующей, без особенностей) функции – применить формулу прямоугольников. С помощью указанной формулы площадь искомой фигуры вычисляется как сумма элементарных прямоугольников, множеством которых заменяется подынтегральная функция f(x). Иллюстрация метода прямоугольников приведена на рисунке 5.1.1.

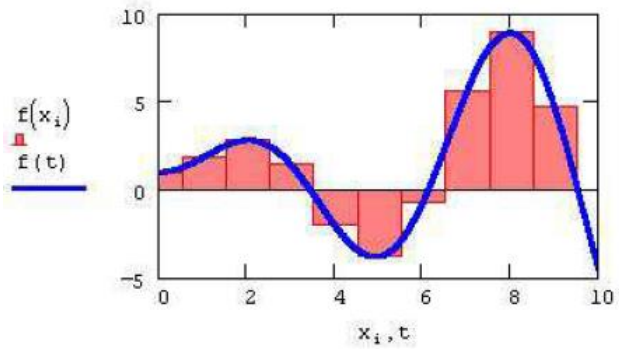


Рисунок 5.1.1 - Метод прямоугольников

Для вычисления интервал интегрирования [a,b] разбивается на n отрезков и на каждом из полученных отрезков функция f(x) заменяется прямоугольником шириной:

$$h := \frac{b - a}{N}$$

И высотой $f(x_1)$. При этом точка x_1 может выбираться, к примеру, как начало каждого элементарного отрезка, либо как его центр. Как несложно убедиться, формулы прямоугольников для этих двух случаев запишутся в виде:

- формула прямоугольников 1

$$Rect1 := h \cdot \sum_{i=1}^{N-1} f(x_i);$$

- формула прямоугольников 2.

$$- \quad Rect2 := h \cdot \sum_{i=1}^{N-1} f(x_i + \frac{h}{2});$$

Вторая формула дает намного лучшую точность интегрирования, из-за чего первая практически никогда не применяется.

5.1.3 Приближенное вычисление интеграла функции методом трапеций

Альтернативный вариант – замена f(x) ломаной линией (с вершинами в концах элементарных отрезков, на которые разбивается интервал интегрирования), т.е. аппроксимация искомого интеграла множеством элементарных трапеций. Формула трапеций такова:

$$Trapz := h\left(\frac{f(a)}{2} + \sum_{i=1}^{N-1} f(x_i) + \frac{f(b)}{2}\right)$$

Очевидно, что при стремлении h к 0, множество прямоугольников (или трапеций) стремится к искомой фигуре,

образованной подынтегральной функцией, а численный результат – к истинному значению интеграла. Можно показать, что формулы прямоугольников обеспечивают второй порядок аппроксимации интеграла, т.е. погрешность его вычисления пропорциональна h^2, а формула трапеций – h^3. Результат зависимости от числа n погрешности вычисления интеграла от тестовой функции по трем формулам численного интегрирования показан на рисунке 5.1.2:

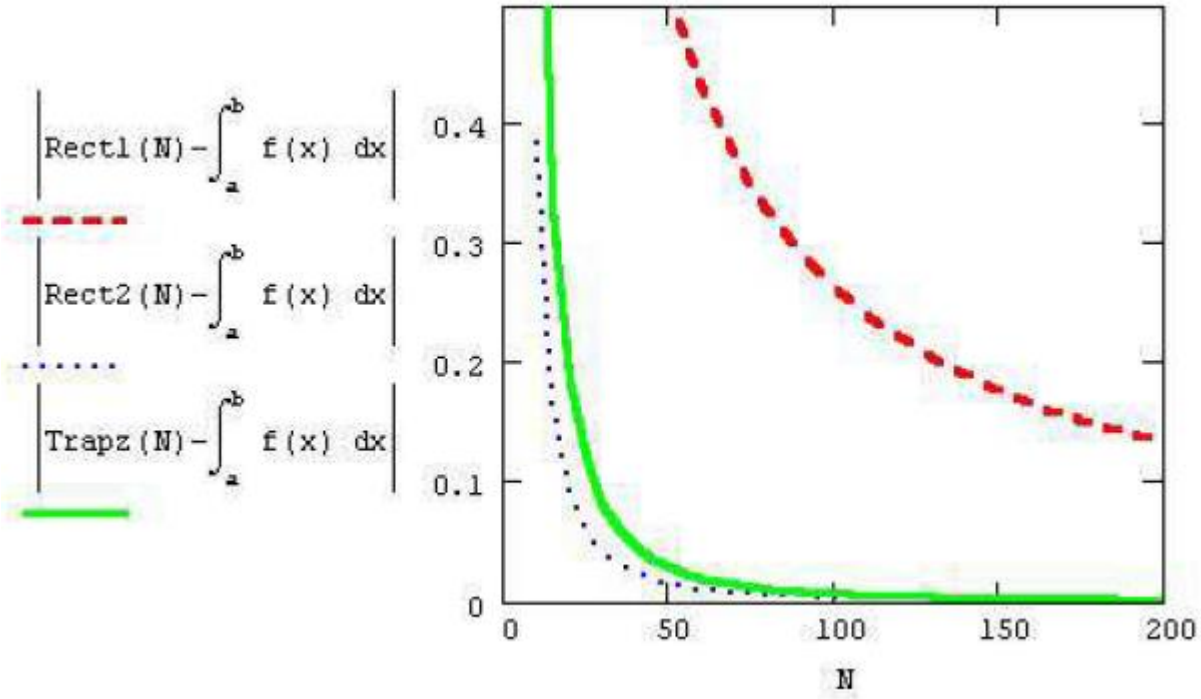


Рисунок 5.1.2 - Результат зависимости от числа n погрешности вычисления интеграла от тестовой функции по трем формулам численного интегрирования

5.1.4 Приближенное вычисление интеграла функции методом Симпсона

Простой прием построения квадратурных формул состоит в том, что подынтегральная функция f(x) заменяется на отрезке [a,b] интерполяционным многочленом, например, многочленом Лагранжа $L_n(x)$; для интеграла имеем приближенное равенство (5.1.1). Предполагается, что отрезок [a,b] разбит на n частей точками (узлами) x_i, наличие которых подразумевается при построении многочлена $L_n(x)$. Для равноотстоящих узлов:

$$x_i = x_0 + ih,\ h = \frac{b-a}{n},\ x_0 = a,\ x_n = b. \qquad (5.1.1)$$

$$\int_a^b f(x)dx = \int_a^b L_n(x)dx \qquad (5.1.2)$$

При определенных допущениях получаем формулу трапеций

$$\int_a^b f(x)dx = h\left(\frac{y_0 + y_n}{2} + y_1 + y_2 + \ldots + y_{n-1}\right), \qquad (5.1.3)$$

где y_i – значения функции в узлах интерполяции.

Имеем следующую оценку погрешности метода интегрирования по формуле трапеций (5.3):

$$|R_n| \le M\frac{|b-a|\cdot h^2}{12},\ \text{где } M = \max\left|f^{(2)}(x)\right|,\ x \in [a,b]. \qquad (5.1.4)$$

Во многих случаях более точной оказывается формула Симпсона (формула парабол):

$$\int_a^b f(x)dx \approx \frac{2h}{3}\left(\frac{y_0 + y_{2m}}{2} + 2y_1 + y_2 + \ldots + 2y_{2m-1}\right). \qquad (5.1.5)$$

Для формулы Симпсона имеем следующую оценку погрешности:

$$|R_n| \le M\frac{|b-a|\cdot h^4}{180},\ \text{где } M = \max\left|f^{(4)}(x)\right|,\ x \in [a,b]. \qquad (5.1.6)$$

5.1.5 Приближенное вычисление интеграла функции в MathCad

Фрагмент документа MathCad вычисления интеграла от заданной функции на отрезке [a, b] по формуле трапеций и прямым способом (рисунке 5.1.3).

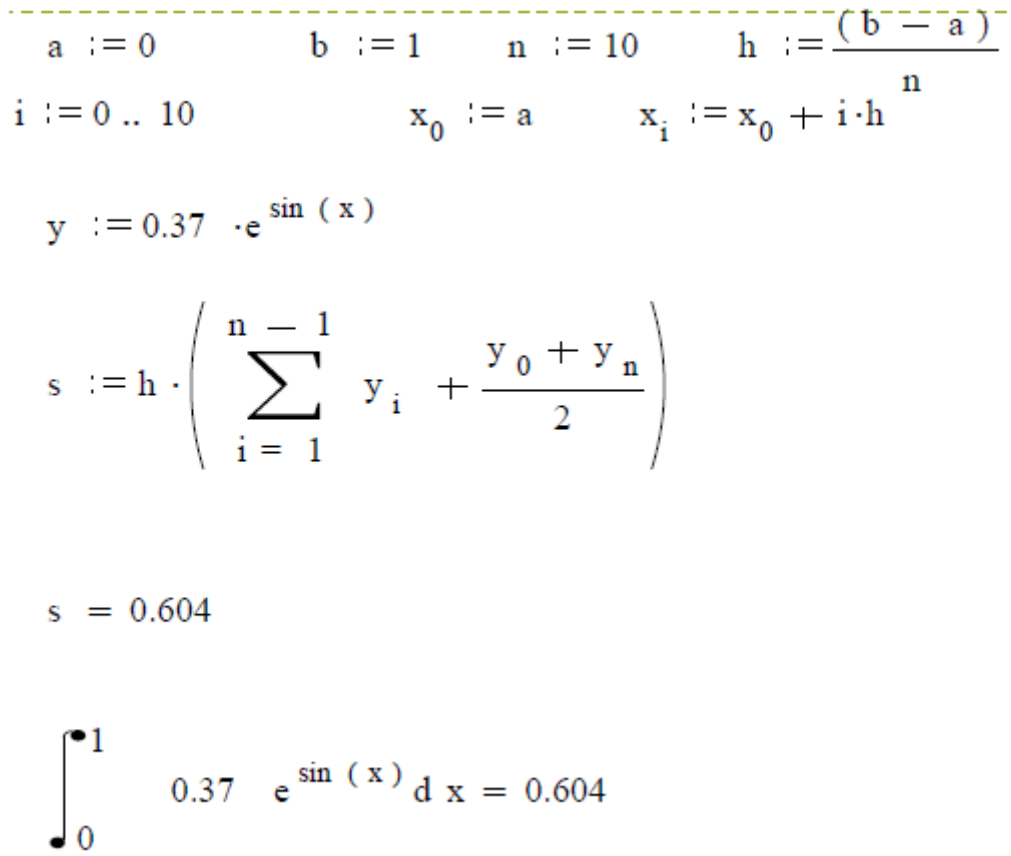


Рисунок 5.1.3 - Вычисления интеграла от заданной функции на отрезке [a,b]

5.1.6 Порядок выполнения работы

Задание 5.1.1

Составить программу вычисления интеграла от заданной функции на отрезке [a, b] по формуле трапеций с шагом h=0.1 и h=0.05. Сравнить результаты. Оценить точность по формуле (5.1.3). Сравнить результаты. Исходные данные для выполнения задания берутся из таблицы 5.1.1.

Задание 5.1.2

Составить программу вычисления интеграла от заданной функции на отрезке [a, b] по формуле Симпсона методом повторного счета с точностью $\varepsilon=10^{-6}$. Исходные данные для выполнения задания берутся из таблицы 5.1.1.

Таблица 5.1.1 – Исходные данные

N	Функция	a	b
1	$0{,}37e^{\sin x}$	0	1
2	$0{,}5x+x\ln x$	1	2
3	$(x+1{,}9)\sin(x/3)$	1	2
4	$\frac{1}{x}\ln(x+2)$	2	3
5	$\frac{3\cos x}{2x+1{,}7}$	0	1
6	$(2x+0{,}6)\cos(x/2)$	1	2
7	$2{,}6x^2\ln x$	1,2	2,2
8	$(x^2+1)\sin(x-0{,}5)$	1	2
9	$x^2\cos(x/4)$	2	3
10	$\frac{\sin(0{,}2x-3)}{x^2+1}$	3	4

5.1.7 Контрольные вопросы

1. Каковы преимущества формулы парабол по сравнению с формулой трапеций и следствием чего являются эти преимущества?
2. Верны ли формулы (5.1.2), (5.1.4) для неравноотстоящих узлов?
3. В каких случаях приближенные формулы трапеций и парабол оказываются точными?
4. Как влияет на точность численного интегрирования величина шага?

5. Каким способом можно прогнозировать примерную величину шага для достижения заданной точности интегрирования?

Можно ли добиться неограниченного уменьшения погрешности интегрирования путем последовательного уменьшения шага?

5.2 Дифференцирование в пакете MathCad

5.2.1 Основные понятия и определения

Пусть дано дифференциальное уравнение первого порядка

$$y' = f(x, y) \tag{5.2.1}$$

Требуется найти на отрезке [a,b] решение y(x), удовлетворяющее начальному условию

$$y(a) = y_0 \tag{5.2.2}$$

Будем предполагать, что условия теоремы существования и единственности выполнены. Для решения используем метод Эйлера (метод первого порядка точности, расчетные формулы (5.2.3)) и метод Рунге-Кутта (метод четвертого порядка точности, расчетные формулы (5.2.4)) с шагом h и 2h. Отметим, что результаты могут сильно отличаться, ввиду того, что метод Эйлера, имея только первый порядок точности, используется, как правило, для оценочных расчетов. Ориентировочную оценку погрешности метода Рунге-Кутта ε вычислить по формуле (5.2.5).

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{5.2.3}$$

$$y_{i+1} = y_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \tag{5.2.4}$$

где h – шаг дискретизации;

$$k_1 = hf(x_i, y_i),\ k_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right),\ k_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \quad (5.2.5)$$

$$k_4 = hf(x_i + h, y_i + k_3).$$

$$\varepsilon = \frac{|y_{2b} - y_b|}{15}.$$

Кроме того, в MathCad имеется тринадцать встроенных функций для решения обыкновенных дифференциальных уравнений различными методами. Большинство из них требуют предварительного представления дифференциального уравнения в виде системы дифференциальных уравнений первого порядка.

5.2.2 Решение дифференциальных уравнений с помощью функции odesolve

Функция **odesolve(x,b)** позволяет решать уравнение без его преобразования. Здесь x - переменная интегрирования, b - верхняя граница изменения аргумента. Нижняя граница равна нулю.

Задача 5.2.1

Используя встроенную функцию **odesolve** решить в MathCad следующее нелинейное обыкновенное дифференциальное уравнение второго порядка с нулевыми начальными условиями:

$$100y'' + 10(y')^2 + 100y = 50\left(\frac{\sin x}{4}\right)$$

На рисунке 5.2.1 показано решение в MathCad обыкновенного дифференциального уравнения с помощью встроенной функции odesolve.

Итак, для решения с использованием этой функции нужно:

1. Ввести директиву given
2. Набрать дифференциальное уравнение. Знак производной набирается клавишей Ё английской клавиатуры, знак «=» - с логической панели,

3. набрать начальные условия,
4. набрать функцию odesolve,
5. сформировать график,
6. Нажать клавишу F9.

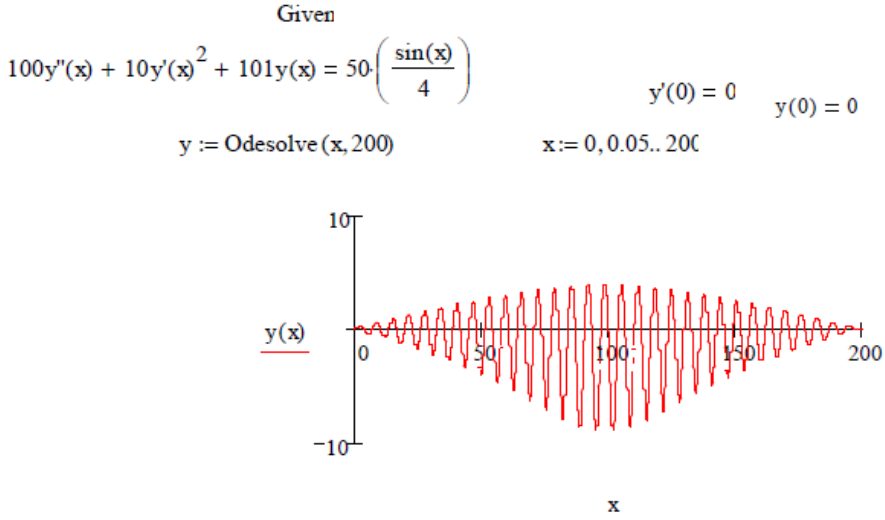


Рисунок 5.2.1 – Решение в MathCad обыкновенного дифференциального уравнения с помощью встроенной функции odesolve

Используя функцию odesolve, можно решать и системы дифференциальных уравнений первого порядка. На рисунке 5.2.2 приведено решение системы двух уравнений. Из рисунка видно, что в функцию odesolve помимо прежних данных вводится вектор имен решаемых функций.

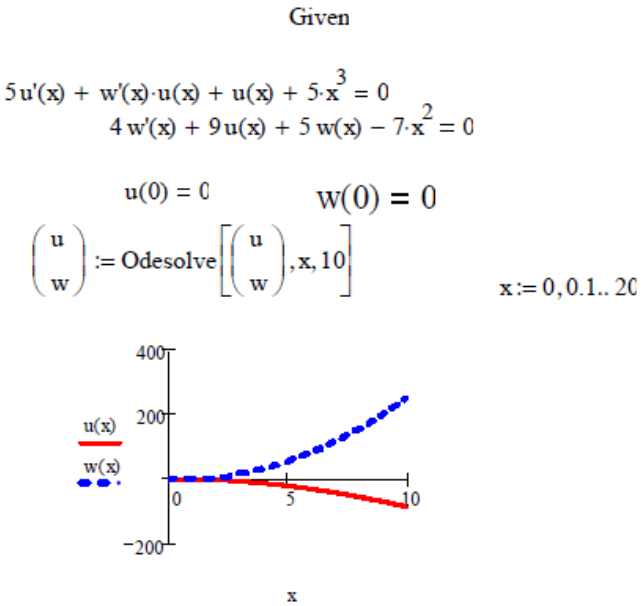


Рисунок 5.2.2 – Решение системы уравнений первого порядка

Недостатком этой функции является фиксированная нижняя граница аргумента.

5.2.3 Решение дифференциальных уравнений с помощью функции rkfixed

Среди в строенных функций MathCad для решения дифференциальных уравнений есть функция их решения методом Рунге - Кутты с постоянным фиксированным шагом. Она имеет вид:

rkfixed(v,x_0,x_k,n,F)

Здесь v - начальные условия, записанные в виде вектора, x_0, x_k - начальное и конечное значения аргумента, n - число шагов, F - правые части системы, записанные в виде вектора.

Уравнения порядка выше первого требуется преобразовывать в систему первого порядка. Разберем сначала, как это делается применительно к пакету MathCad.

Любое обыкновенное дифференциальное уравнение выше первого порядка может быть представлено в виде системы дифференциальных уравнений первого порядка, число которых равно порядку преобразуемого уравнения. Покажем это на примере дифференциального уравнения третьего порядка.

Дано

$$ay'''+by''+cy'+dy = f(x) \qquad (5.2.6)$$

Введем подстановки

$$y'' = y_0 \qquad (5.2.7)$$

$$y' = y_1 \qquad (5.2.8)$$

Тогда уравнение (5.2.6) запишется в виде:

$$ay_0'+by_0+cy_1+dy_2=f(x) \qquad (5.2.9)$$

Это первое уравнение первого порядка будущей системы.
Продифференцируем уравнение (5.2.8). Получим

$$y''=y_1' \qquad (5.2.10)$$

Левые части уравнений (5.2.7) и (5.2.9) равны. Следовательно, равны и их правые части. Отсюда

$$y_1=y_0 \qquad (5.2.11)$$

Продифференцируем уравнение (5.2.9). Получим

$$y'=y_2' \qquad (5.2.12)$$

Левые части уравнений (5.2.8) и (5.2.12) равны, следовательно, равны и их правые части. Тогда

$$y_2'=y_1 \qquad (5.2.13)$$

Мы получили системы дифференциальных уравнений первого порядка

$$\begin{aligned} &ay_0'+by_0+cy_1+dy_2=f(x)\\ &y_1'=y_0\\ &y_2'=y_1 \end{aligned} \qquad (5.2.14)$$

Разрешив первое уравнение относительно производной, окончательно получим:

$$\begin{aligned} &y_0'=-by_0/a-cy_1/a-dy_2/a+f(x)/a\\ &y_1'=y_0\\ &y_2'=y_1 \end{aligned} \qquad (5.2.15)$$

В качестве иллюстрационного примера рассмотрим метод Рунге-Кутты с фиксированным шагом применительно к дифференциальному уравнению второго порядка:

$$\frac{d^2y}{dt} + 1{,}33\frac{dy}{dt} + 1{,}667y = 0$$

при заданных начальных условиях

$$t_0 = 0, y(0) = 1, \frac{dy}{dt} = 0$$

и заданном конце вычислений:

$$t_k = 13.$$

Проведя преобразование в систему уравнений первого порядка, получим:

$$\frac{dy}{dt} = -1{,}33y_0 - 1{,}667y_1$$
$$\frac{dy_1}{dt} = y_0.$$

Начальные условия примут вид

$$y_0(0) = 0, y_1(0) = 1.$$

Записав правые части и начальные условия в виде векторов, получим

$$f(t, y) = \begin{bmatrix} -1{,}33y_0 - 1{,}66y_1 \\ y_0 \end{bmatrix} v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

где f(t, y) - это вектор правых частей системы.

При его формировании надо обратить внимание на следующее:

1 .Вместо буквы f можно использовать любую другую букву. Но тогда и во встроенной функции нужно, естественно, использовать ту же букву.

2. Внутри скобок первое имя (в нашем случае t) является именем аргумента, по которому происходит интегрирование дифференциального уравнения. Т. е. опять - таки это может быть и другая буква.

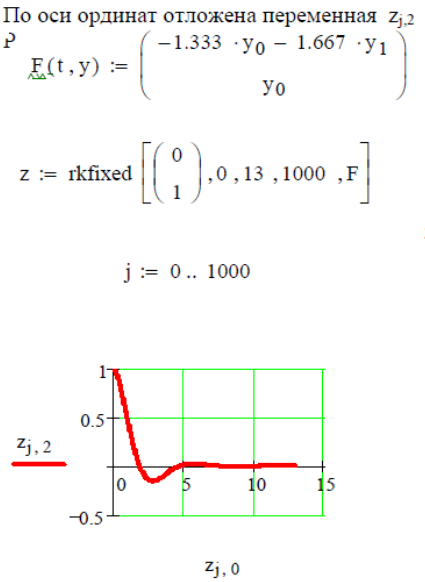
3. Вторая буква внутри скобок - это вектор имен зависимых переменных. Им эти имена полностью определяются. Если принято имя *y,* то именами переменных должны являться y_0, y_1, y_2 и т.д., причем первое уравнение - это $dy_0/dt =$, второе уравнение - это $dy_1/dt =$ и т. д.

На рисунке 5.2.2 приведено решение этой системы.

Ответ получен в виде вектора и в виде графика. В первой строке этого вектора показаны номера переменных: z_0 это время, z_1 - производная y_0, z_2 - сама функция y. Выведены только первые 11 значений вектора ответа.

Начальные условия заданы встроенным вектором v.

Ранжировка j := 0..1000 относится не к функции rkfixed , а к графику. Для встроенной функции число точек решения задано числом 1000 внутри нее.



$z =$

	0	1	2
0	0	0	1
1	0.013	-0.021	1
2	0.026	-0.043	0.999
3	0.039	-0.063	0.999
4	0.052	-0.084	0.998
5	0.065	-0.104	0.997
6	0.078	-0.123	0.995
7	0.091	-0.143	0.993
8	0.104	-0.161	0.991
9	0.117	-0.18	0.989
10	0.13	-0.198	0.987
11	0.143	-0.216	0.984
12	0.156	-0.233	0.981
13	0.169	-0.25	0.978
14	0.182	-0.267	0.975
15	0.195	-0.283	0.971

Рисунок 5.2.3 – Решение дифференциального уравнения с помощью функции rxfixed

5.2.4 Построение фазовых портретов

Фазовым портретом называется график функции в координатах y (y) (рисунок 5.2.4). Фазовые портреты используются в теории автоматического управления для определения переходных процессов в автоматических системах.

$T := 10 \qquad \xi := 0$

$$f1(t, y1) := \begin{pmatrix} \dfrac{-2\xi \cdot T \cdot y1_0 - y1_1}{T^2} \\ y1_0 \end{pmatrix} \qquad z1 := \text{rkfixed}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}, 0, 100, 1000, f1\right]$$

$\xi := 0.1$

$$f2(t, y2) := \begin{pmatrix} \dfrac{-2\xi \cdot T \cdot y2_0 - y2_1}{T^2} \\ y2_0 \end{pmatrix} \qquad z2 := \text{rkfixed}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}, 0, 100, 1000, f2\right]$$

$= -0.1$

$$f3(t, y3) := \begin{pmatrix} \dfrac{-2\xi \cdot T \cdot y3_0 - y3_1}{T^2} \\ y3_0 \end{pmatrix} \qquad z3 := \text{rkfixed}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}, 0, 100, 1000, f3\right]$$

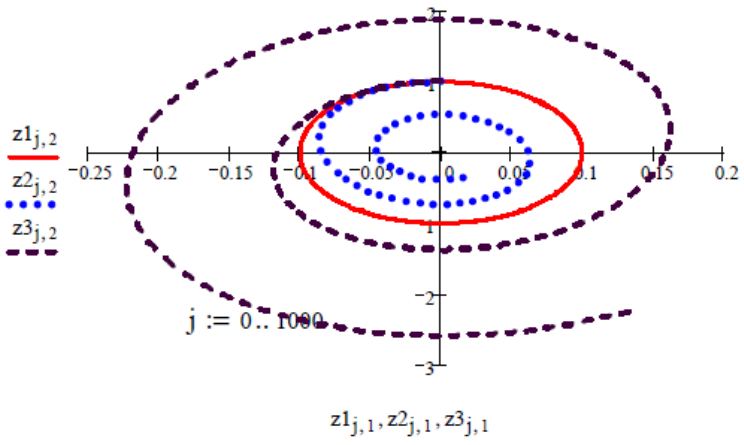


Рисунок 5.2.4 – Фазовые портреты

Построим фазовый портрет для дифференциального уравнения второго порядка, записанного в форме:

Здесь T- постоянная времени, ξ - коэффициент затухания.

Решаются три одинаковых дифференциальных уравнения с одинаковыми начальными условиями, но с разными коэффициентами затухания.

Все три решения начинаются в одной точке (0,1). Но первая сплошная кривая при ξ=0 - эллипс, характеризует консервативную систему, незатухающие гармонические колебания. Вторая пунктирная кривая при ξ=0.1 - затухающая кривая, характеризует затухающие колебания, такая система является устойчивой (это характерно для всех 0 <ξ<1).

Третья кривая пунктиром - при ξ=-0.1 характеризует неустойчивую колебательную систему.

Возможно решение тем же методом с автоматическим выбором шага. Для этого служит функция **rkadapt(y, x1, x2, n points, D).**

5.2.5 Решение дифференциальных уравнений методом Буль-Штейера

Численный метод Буль - Штейера дает более точное решение, чем метод Рунге Кутта. Ниже на рисунке 5.2.5 приведено решение этим методом нелинейной системы второго порядка: с начальными условиями

$$x_0(0) = 50$$
$$x_1(0) = 2000$$
$$\frac{dx_0}{dt} = x_0(0{,}01x_1 - 1)$$
$$\frac{dx_1}{dt} = -0{,}01x_0x_1$$

Из рисунка видно, что последовательность решения не отличается от последовательности решения методом Рунге-Кутта.

$$f(t,x) := \begin{bmatrix} x_0 \cdot (0.01 \cdot x_1 - 1) \\ -0.01 \cdot x_0 \cdot x_1 \end{bmatrix} \qquad j := 0..100 \qquad z := \text{Bulstoer}\left[\begin{pmatrix} 50 \\ 2000 \end{pmatrix}, 0, 13, 1000, f\right]$$

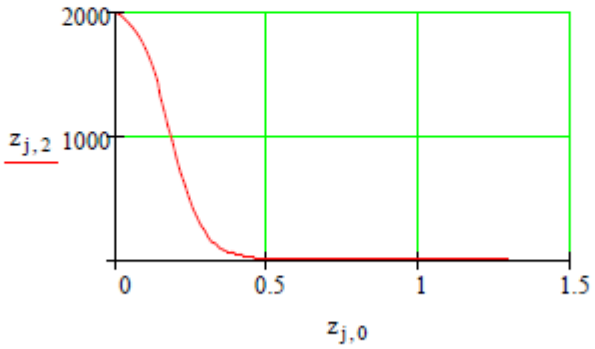


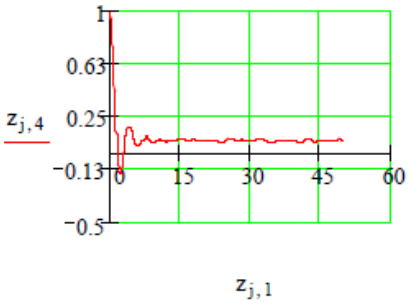
Рисунок 5.2.5 – Решение дифференциального уравнения методом Буль-Штейера

5.2.6 Дифференциальные уравнения со случайной составляющей

MathCad позволяет решать дифференциальные уравнения, коэффициенты которого являются случайными функциями. При этом, конечно, каждая реализация решения будет отличаться от предыдущей.

Ниже, на рисунке 5.2.6 показано решение уравнения третьего порядка, свободный член которого реализуется функцией rnd - возвращающей равномерно распределенную случайную величину в пределах 0-1.

$j := 1..1024$ $\quad z := rkfixed\left[\begin{pmatrix}0\\0\\1\end{pmatrix}, 0, 50, 1024, f\right]$ $\quad f(t,y) := \begin{pmatrix}-3\cdot y_1 - 5y_2 - 6y_3 + rnd(1)\\ y_1\\ y_2\end{pmatrix}$



$z =$

	1	2	3	4
1	0	0	0	1
2	0.049	-0.249	$-6.172\cdot10^{-3}$	1
3	0.098	-0.455	-0.023	0.999
4	0.146	-0.629	-0.05	0.997
5	0.195	-0.768	-0.084	0.994
6	0.244	-0.88	-0.125	0.989
7	0.293	-0.969	-0.17	0.982
8	0.342	-1.052	-0.219	0.972
9	0.391	-1.092	-0.272	0.961
10	0.439	-1.112	-0.326	0.946
11	0.488	-1.114	-0.38	0.929
12	0.537	-1.097	-0.434	0.909
13	0.586	-1.071	-0.487	0.886
14	0.635	-1.015	-0.538	0.861
15	0.684	-0.964	-0.587	0.834
16	0.732	-0.891	-0.632	0.804

Рисунок 5.2.6 – Решение дифференциального уравнения со случайными параметрами

5.2.7. Решение линейных дифференциальных уравнений высших порядков

Дифференциальное уравнение порядка n

$$a_n \frac{d^n y}{dx_n} + a_{n-1} \frac{d^{n-1} y}{dx_n} + \ldots a_1 \frac{dy}{dx} + a_0 = f(x)$$

называется линейным, если все его коэффициенты a_n (n=0, 1,…) являются числами или функциями аргумента x.

Дифференциальное уравнение называется нелинейным, если хотя бы один из этих коэффициентов зависит от искомой функции y, или ее производных. Рассмотрим линейное дифференциальное уравнение второго порядка:

$$T^2 \frac{d^2y}{dt^2} + 2\xi T \frac{dy}{dt} + y = 0$$

$$y(0)=0$$

$$y'(0)=1$$

При заданных параметрах T и ξ указанное дифференциальное уравнение является линейным с постоянными коэффициентами. В теории автоматического управления линейное уравнение, записанное в такой форме, называется типовым динамическим звеном второго порядка.

При T=10 исследуем влияние на решение изменения численного значения коэффициента ξ**,** определенного в диапазоне как

$$\xi=\{-0.5, 0, 0.5, 5\}$$

Убедимся, что

при ξ=5 решение - монотонно затухающая кривая,

при ξ=0 решение - чистая синусоида,

при ξ= -5 решение - монотонно расходящаяся кривая.

при ξ= -0,5 решение расходящаяся синусоида;

при ξ=0,5 - решение затухающее колебательное,

Следовательно, если данным дифференциальным. уравнением описывается какая - либо реальная система (а это бывает сплошь и рядом), то при ξ=0, -0.5, -5 система неработоспособна, т.к. является неустойчивой.

Коэффициент ξ в данном уравнении называют коэффициентом затухания.

При ξ=0.5 исследуем теперь влияние на решение изменения численного значения величины коэффициента T, определенного в диапазоне как

$$T=\{1, 5, 10, 25\}.$$

Убедимся, что увеличение T ведет к увеличению запаздывания системы. Коэффициент T называют линейным запаздыванием.

Если в любом дифференциальном уравнении и правая часть равна 0 и заданы нулевые начальные условия, то его решение будет также тождественно равно 0. Начальные условия и правая часть являются возмущениями.

При T=10, ξ=0.5 исследуем влияние на решение изменения численного значения величины начальных условий (возмущений), определенных в диапазоне как

$$y_0=\{10, 100\}.$$

Мы видим, что в этом случае характер решения не изменился, устойчивая система продолжает оставаться устойчивой при увеличении возмущений.

Пример решения линейного уравнения

Задано одно и то же уравнение, но разные начальные условия (различное возмущение) (рисунок 5.2.7).

$$3d^2y/dt^2+0{,}8dy/dt+6y=0 \qquad 3d^2z/dt^2+0{,}8dz/dt+6z=0$$
$$t_0=0,\ y_0=10,\ dy/dt_{t=0}=0 \qquad t_0=0,\ z_0=100,\ dz/dt_{t=0}=0$$

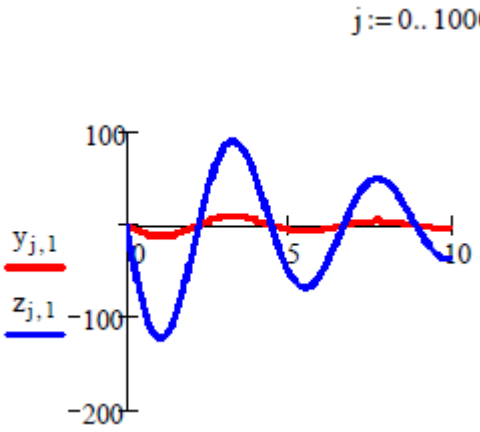


Рисунок 5.2.7 - Влияние начальных условий на решение линейного дифференциального уравнения.

Легко видеть, что характер решения не изменился. В обоих случаях - затухающие колебания с одинаковой частотой.

5.2.8 Решение нелинейных дифференциальных уравнений

Опыт показывает, что решение дифференциальных уравнений в численном виде приводит к тому, что студенты не понимают принципиального отличия нелинейных систем от линейных.

Аналитическое решение подчеркивает это различие, а численное решение – стирает.

Если решение линейных дифференциальных уравнений полностью определяется параметрами (коэффициентами) этих уравнений, то решение нелинейных уравнений зависит также и от величины возмущений.

Пример решения нелинейного уравнения

Рассмотрим решение двух одинаковых нелинейных дифференциальных уравнений с различными начальными условиями.

$$d^2y/dt^2 + 1{,}53dy/dt * y + 0{,}16y = 0 \qquad d^2z/dt^2 + 1{,}53dz/dt * z + 0{,}166z = 0$$
$$t_0 = 0,\ y_0 = 0{,}5,\ dy/dt_{t=0} = 0 \qquad t_0 = 0,\ z_0 = 3,\ dz/dt_{t=0} = 0$$

На рисунке 5.2.8 показаны решения обоих уравнений.

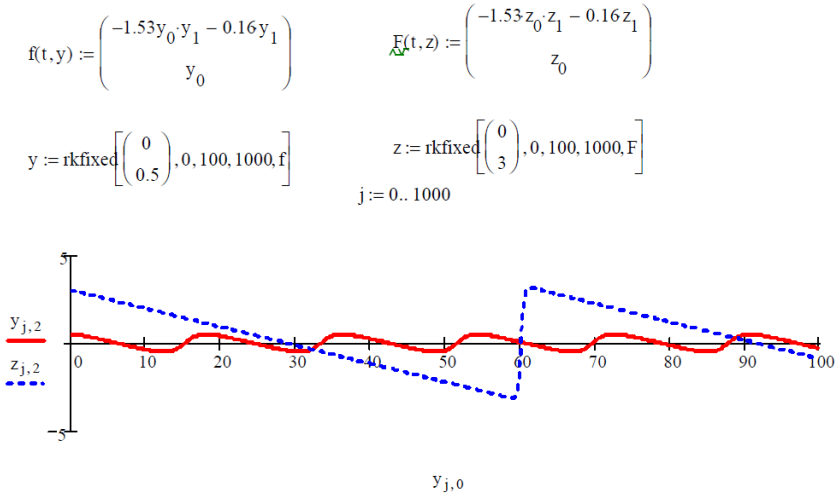


Рисунок 5.2.8 - Влияние начальных условий на решение нелинейного дифференциального уравнения

Вывод: При изменении начальных условий изменилась форма кривой и период колебаний.

5.2.9 Символьное решение линейных дифференциальных уравнений

Символьное решение линейных дифференциальных уравнений проводится классическим способом, через решение характеристического уравнения с последующим определением констант методом вариации постоянных.

Разберем символьное решение следующего линейного дифференциального уравнения третьего порядка

$$0.01\frac{d^3y}{dt^3} + 0.1\frac{d^2y}{dt^2} + 0.1\frac{dy}{dt} + 0.01 = 7 \qquad \begin{matrix} y(0) = 8.1 \\ y'(0) = 0 \\ y''(0) = 0 \end{matrix}$$

Его характеристическое уравнение:

$$0.01s^3 + 0.1s^2 + 0.1s + 0.01 = 0$$

Мы решаем его символьно

$$0.01s^3 + 0.1s^2 + 0.1s + 0.01\,\text{solve},s \rightarrow \begin{pmatrix} -1. \\ -8.887482193696606 10302 \\ -0.11251780630393869 79 \end{pmatrix}$$

Однако полученные корни имеют слишком много знаков. Для уменьшения числа значащих цифр найдем эти корни с одновременным переводом их с помощью оператора float (плавающая точка) в форму с плавающей точкой.

$$0.01s^3 + 0.1s^2 + 0.1s + 0.01\left|\begin{matrix}\text{solve},s \\ \text{float},3\end{matrix}\right. \rightarrow \begin{pmatrix} -1. \\ -8.89 \\ -0.113 \end{pmatrix}$$

Поставив float 3, мы ограничились тремя значащими знаками. Установка двух операторов производится автоматически нажатием последовательно двух кнопок на панели символьных вычислений.

Отсюда частные решения имеют вид:

$$y1(t) := \exp(-t) \qquad y2(t) := \exp(-8.89 \cdot t) \qquad y3(t) := \exp(-0.113 \cdot t)$$

и общее решение равно

$$y = C1e^{-8.89t} + C2e^{-t} + C3e^{-0.113t}$$

Решая систему алгебраических уравнений, определяем C1, C2, C3.

Given

$$C1 \cdot y1(t) + C2 \cdot y2(t) + C3 \cdot y3(t) = 0$$

$$C1 \cdot \frac{d}{dt} y1(t) + C2 \cdot \frac{d}{dt} y2(t) + C3 \cdot \frac{d}{dt} y3(t) = 0$$

$$C1 \cdot \frac{d^2}{dt^2} y1(t) + C2 \cdot \frac{d^2}{dt^2} y2(t) + C3 \cdot \frac{d^2}{dt^2} y3(t) = 7$$

$$\text{find}(C1, C2, C3)\,\text{float}, 3 \rightarrow \begin{bmatrix} \dfrac{-1.00}{\left[e^{\left(-0.100e^{-2}\right)\cdot t}\right]^{1000}} \\ \dfrac{0.101}{\left[e^{\left(-0.100e^{-2}\right)\cdot t}\right]^{8890}} \\ \dfrac{0.899}{\left[e^{\left(-0.100e^{-2}\right)\cdot t}\right]^{113}} \end{bmatrix}$$

Используя оператор simplify (упростить. Соответствующая кнопка символьной панели), упрощаем найденное решение

$$\begin{bmatrix} \frac{-1.00}{\left[e^{\left(-0.100e^{-2}\right)\cdot t}\right]^{1000}} \\ \frac{0.101}{\left[e^{\left(-0.100e^{-2}\right)\cdot t}\right]^{8890}} \\ \frac{0.899}{\left[e^{\left(-0.100e^{-2}\right)\cdot t}\right]^{113}} \end{bmatrix} \left| \begin{matrix} \text{simplify} \\ \text{float},3 \end{matrix} \right. \rightarrow \begin{bmatrix} (-1.)\cdot e^{t} \\ 0.101\cdot e^{8.89\cdot t} \\ 0.889\cdot e^{0.113\cdot t} \end{bmatrix}$$

Интегрируем

$$\int -e^{t}\,dt\,\text{float},3 \rightarrow (-1.)\cdot e^{t}$$

$$\int 0.101e^{8.89\cdot t}\,dt\,\text{float},3 \rightarrow 0.114e^{-1}\cdot e^{8.89\cdot t}$$

$$\int 0.899e^{0.113\cdot t}\,dt\,\text{float},3 \rightarrow 7.96\cdot e^{0.113\cdot t}$$

Прибавляем постоянные интегрирования

$$\begin{bmatrix} (-1.)\cdot e^{t} \\ 0.114e^{-1}\cdot e^{8.89\cdot t} \\ 7.96\cdot e^{0.113\cdot t} \end{bmatrix} + \begin{pmatrix} C11 \\ C12 \\ C13 \end{pmatrix} \rightarrow \begin{bmatrix} (-1.)\cdot e^{t} + C11 \\ 0.114e^{-1}\cdot e^{8.89\cdot t} + C12 \\ 7.96\cdot e^{0.113\cdot t} + C13 \end{bmatrix}$$

Формируем общее решение

$$\begin{pmatrix} y1(t) \\ y2(t) \\ y3(t) \end{pmatrix}^{T} \cdot \begin{bmatrix} (-1.)\cdot e^{t} + C11 \\ 0.114e-1\cdot e^{8.89\cdot t} + C12 \\ 7.96\cdot e^{0.113\cdot t} + C13 \end{bmatrix} \left| \begin{matrix} \text{simplify} \\ \text{float},3 \end{matrix} \right. \rightarrow$$

$$\rightarrow 0.200e^{-3}\begin{bmatrix} 0.349e^{5}e^{t} + 0.500e^{4}C11 + 0.500e^{4}e^{(-7.89)t}C12 + \\ +0.500e^{4}e^{0.887t}C13 \end{bmatrix}\cdot e^{(-1.)t}$$

Для определения констант дифференцируем два раза с упрощением

$$\frac{d}{dt}\left[0.200e^{-3}\left[0.349e^{5}e^{t}+0.500e^{4}C11+0.500e^{4}e^{(-7.89)t}C12+\right.\right.$$
$$\left.\left.+0.500e^{4}e^{0.887t}C13\right]\cdot e^{(-1.)t}\right]\left|\begin{matrix}\text{simplify}\\ \text{float},3\end{matrix}\right.\rightarrow$$
$$\rightarrow(-8.89)\cdot e^{(-8.89)t}\cdot C12-0.113\cdot e^{(-0.113)t}\cdot C13-1.\cdot e^{(-1.)t}\cdot C11$$

Приравнивая нулю время t, составляем уравнения для определения констант

$$\frac{d}{dt}\left[0.200e^{-3}\left[0.349e^{5}e^{t}+0.500e^{4}C11+0.500e^{4}e^{(-7.89)t}C12+\right.\right.$$
$$\left.\left.+0.500e^{4}e^{0.887t}C13\right]\cdot e^{(-1.)t}\right]\left|\begin{matrix}\text{simplify}\\ \text{float},3\end{matrix}\right.\rightarrow$$

$$\rightarrow 79.0\cdot e^{(-8.89)t}\cdot C12-0.128e^{-1}\cdot e^{(-0.113)t}\cdot C13+e^{(-1.)t}\cdot C11$$

Given

$$C11+C12+C13=8.1$$

$$-8.89*C12-0.113*C13-C11=0$$
$$-79*C12-0.128*C13+C11=0$$

$$\text{find}(C11,C12,C13)\,\text{float},3\rightarrow\begin{pmatrix}-1.05\\ 0.156e^{-2}\\ 9.15\end{pmatrix}$$

Общее решение принимает вид:

$$y(t):=0.200e^{-3}\left[0.349e^{5}e^{t}+0.500e^{4}(-1.05)+0.500e^{4}e^{(-7.89)t}(0.00156)+\right.$$

$$+0.500e^{4}e^{0.887t}9.15\Big]\cdot e^{(-1.)t}$$

Упрощаем выражение:

$$y(t)\left|\begin{array}{l}\text{simplify}\\ \text{float},3\end{array}\right.\rightarrow 0.400e^{-1}\Big[0.175e^{6}e^{t}-0.263e^{5}+$$
$$+39.e^{(-7.89)t}+0.229e^{6}e^{0.887t}\Big]\cdot e^{(-1.)t}$$

Окончательное выражение для общего решения:

$$y(t):=0.400e^{-1}\Big[0.175e^{6}e^{t}-0.263e^{5}+39.e^{(-7.89)t}+0.229e^{6}e^{0.887t}\Big]\cdot e^{(-1.)t}$$

Строим график (рисунок 5.2.9):

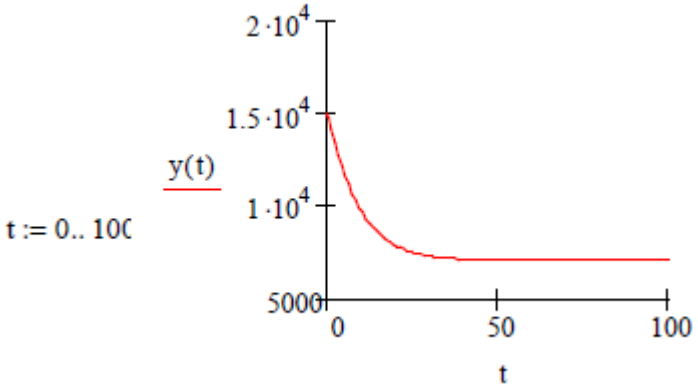


Рисунок 5.2.9 – График, полученный по результатам вычислений

5.2.10 Решение краевых задач для обыкновенных дифференциальных уравнений

Краевая задача решается в MathCad методом пристрелки с помощью встроенной функции **sbval**. Эта функция на основании заданных конечных условий вычисляет начальные условия. После этого задача сводится к задаче Коши и ее можно решить, используя известную функцию **rkfixed.**

Функция **sbval** имеет вид:

sbval(v,x1,x2,f,load,score).

Здесь: **v**-вектор незаданных начальных условий, т.е. тех начальных условий, вместо которых заданы конечные условия. Обычно выбираем все компоненты v, равными 1; **x1, x2**- начальное и конечное значения аргумента, т.е. интервал решения дифференциального уравнения; **f(x,y)-** векторная функция, содержащая правые части дифференциального уравнения , та же, что используется при решении дифференциального уравнения с помощью **rkfixed**; **load(x1,v)** -вектор всех начальных условий. Сюда помещаются все заданные начальные условия, а вместо незаданных помещаются компоненты вектора v; **score(x2, y**)-вектор конечных условий, в который помещаются разности между текущими значениями тех переменных, для которых заданы конечные условия, и их численными значениями.

В качестве иллюстративного примера рассмотрим однородное дифференциальное уравнение пятого порядка

$$\frac{d^5y}{dx^5} + y = 0$$

Задан интервал решения: 0, 1.

Для уравнения пятого порядка должны быть заданы пять граничных условий.

Заданы начальные условия $y(0)=0$, $dy/dx_{x=0}=7$ и конечные условия $y(1)=1$, $dy/dx_{x=1}=10$, $d2y/dx^2{}_{x=1}=5$.

Сформируем для трех не заданных начальных условий вектор v ,присвоив всем его элементам единичные значения

$$v = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Сформируем вектор f(x,y). Для этого введем подстановки:

$$\frac{d^4y}{dx^4} = y_0$$

$$\frac{d^3y}{dx^3} = y_1$$

$$\frac{d^2y}{dx^2} = y_2$$

$$\frac{dy}{dx} = y_3$$

$$y = y_4$$

Вместо одного уравнения пятого порядка мы имеем теперь систему из пяти уравнений первого порядка:

$$\frac{dy_0}{dx} + y_4 = 0$$

$$\frac{dy_1}{dx} = y_0$$

$$\frac{dy_2}{dx} = y_1$$

$$\frac{dy_3}{dx} = y_2$$

$$\frac{dy_4}{dx} = y_3$$

Разрешив первое уравнение этой системы относительно производной, получим:

$$\frac{dy_0}{dx} = -y_4$$

$$\frac{dy_1}{dx} = y_0$$

$$\frac{dy_2}{dx} = y_1$$

$$\frac{dy_3}{dx} = y_2$$

$$\frac{dy_4}{dx} = y_3$$

Фрагмент решения краевой задачи представлен на рисунке 5.2.10.

$xn := 0 \quad xk := 1 \quad v := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad load(xn, v) := \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ 7 \\ 0 \end{pmatrix} \quad score(xk, y) := \begin{pmatrix} y_2 - 5 \\ y_3 - 10 \\ y_4 - 1 \end{pmatrix}$

Рисунок 5.2.10 – Фрагмент решения краевой задачи

Следовательно,

$$f(t, y) = \begin{bmatrix} -y_4 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Сформируем вектор

$$\text{load}(xn, v) = \begin{bmatrix} 0 \\ 7 \\ v_0 \\ v_1 \\ v_2 \end{bmatrix}$$

Здесь xn - начальное значение x, а вектор заполняется следующим образом: была проведена подстановка

$$\frac{d^4 y}{dx^4} = y_0$$

$$\frac{d^3 y}{dx^3} = y_1$$

$$\frac{d^2 y}{dx^2} = y_2$$

$$\frac{dy}{dx} = y_3$$

$$y = y_4$$

После которой вектор переменных приобрел следующий вид:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

С учетом подстановки нам заданы y_0=0 и y_1=7. Остальные начальные условия неизвестны и мы заполняем их вектором v.

Сформируем вектор score(xk,y) для заданных конечных условий:

$$\text{score}(xk, y) = \begin{bmatrix} y_0 - 1 \\ y_1 - 10 \\ y_2 - 5 \end{bmatrix}$$

Вектор включает те переменные, для которых заданы конечные значения.

Все вышеперечисленные действия были произведены в MathCad, и было получено искомое решение, приведенное на рисунке 5.2.11.

$xn := 0 \qquad xk := 1 \qquad \text{score}(xk, y) := \begin{pmatrix} y_2 - 5 \\ y_3 - 10 \\ y_4 - 1 \end{pmatrix} \qquad \text{load}(xn, v) := \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ 7 \\ 0 \end{pmatrix}$

$v := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad f(x, y) := \begin{pmatrix} -y_4 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \qquad S := \text{sbval}(v, 0, 1, f, \text{load}, \text{score})$

$y := \text{rkfixed}\left[\begin{pmatrix} -516.257 \\ 348.107 \\ -85.014 \\ 7 \\ 0 \end{pmatrix}, xn, xk, 1000, f\right] \qquad j := 0..1000$

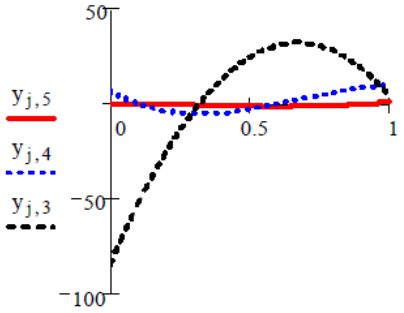


Рисунок 5.2.11 - Решение краевой задачи в MathCad

Функция sbval вычисляет неизвестные начальные условия, а мы затем вводим их в функцию rkfixed и получаем решение.

На графиках приведены те кривые, для которых заданы конечные условия, чтобы можно было убедиться в их выполнении, а именно $y_3(x)$, $y_4(x)$, $y_5(x)$.

Ниже приведены вычисленные начальные и конечные значения переменных. Весь диапазон решения разбит нами на 1000 точек (j=0..1000), поэтому $y_{0,0}$ - это начальное, а $y_{1000,0}$ - конечное значения y_0. То же относится к другим переменным.

$y_{0\,5}=0$ $y_{1000,5}=1$ $y_{0\,4}=7$ $y_{1000\,4}=10$ $y_{1000,\,3}=5$

Как видим, вычисленные значения совпадают с заданными.

5.2.11 Решение линейных уравнений с переменными коэффициентами

В качестве иллюстративного примера рассмотрим линейное дифференциальное уравнение с переменными коэффициентами следующего вида:

$$y''+xy'+x^2y=x$$

В таких уравнениях коэффициенты при переменной являются функцией независимой переменной - аргумента, в нашем случае x. Задан интервал вычислений $0 \le x \le 5$ 0, заданы конечные условия y(5)=2, y′(5)= 3.

Нестационарные дифференциальные уравнения относятся к классу линейных, однако их аналитическое решение обычно затруднено, и их проще решать численно.

Перейдя к системе дифференциальных уравнений первого порядка, имеем (проделать самостоятельно):

$$y_0' = -xy_0 - x^2y_1 + x$$

$$y_1' = y_0$$

Далее составляем все необходимые функции, как показано на рисунке 5.2.12 и проводим решение.

Это уравнение является сложным для компьютера, что видно из длительности его решения.

Проверим вычисление крайних точек:

$y_{0,2} = 2.073 \times 10^3$ $y_{1000,2} = 2$ $y_{0,1} = 2.673 \times 10^3$ $y_{1000,2} = 3$

Как видим, конечные значения совпали с заданными.

$xn := 0$ $xk := 5$ $v := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $load(xn, v) := v$ $score(xk, y) := \begin{pmatrix} y_0 - 3 \\ y_1 - 2 \end{pmatrix}$

$f(x, y) := \begin{pmatrix} -x \cdot y_0 - x^2 \cdot y_1 + x \\ y_0 \end{pmatrix}$ $s := sbval(v, xn, xk, f, load, score)$ $s =$

$y := rkfixed(s, xn, xk, 1000, f)$ $j := 0 .. 1000$

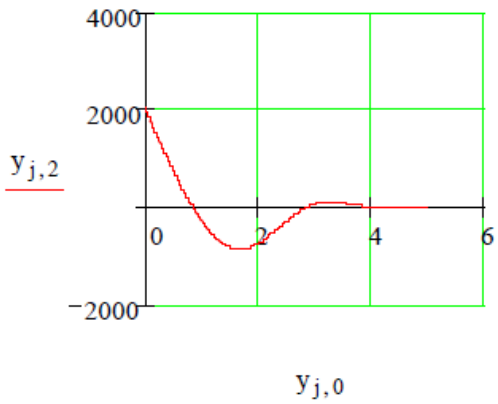


Рисунок 5.2.12 - Решение краевой задачи для нестационарного дифференциального уравнения

5.2.12 Порядок выполнения работы

Задача 5.2.1

Используя функцию odesolve, решить самостоятельно приведенные ниже дифференциальные уравнения. Построить графики решения:

$$y''+5y'+10y = 5x$$
$$y'(0) = 0$$
$$y(0) = 0$$

$$y'''+50y''+y'y+9y = 0$$
$$y''(0) = 0$$
$$y'(0) = 0$$
$$y(0) = 8$$

Задача 5.2.2

Решить в MathCad дифференциальное уравнение второго порядка

$$T^2d^2y/dt^2+\xi Tdy/dt+y=0$$

при начальных условиях $t_0=0$, $y(t_0)=1$, $dy/dt(t_0)=0$ и заданных значениях параметров $T=10$, $\xi=0.5$.

Задача 5.2.3

Решить в MathCad самостоятельно следующие дифференциальные уравнения:

$$3d^2y/dx^2+5dy/dx+6y=0 \qquad x_0=0,$$
$$y(x_0)=2, dy/dx(x_0)=0.$$
$$x_{лон}=20,\ n=500.$$

$$5d^3y/dt^3+9\ d^2y/dt^2-2dy/dt+8y=1/(t+1).$$
$$y(0)=0,$$
$$y`(0)=1,$$
$$y``(0)=0.$$

Задача 5.2.4

Решить самостоятельно приведенные ниже системы уравнений первого порядка

$$y'(x) + y(x)z(x) + 8x = 0 \qquad u'(t)w(t) + u(t) - 3t = 0$$
$$z'(x) + 8z(x) - 10 = 0 \qquad w'(t) - w(t) + t^2 = 0$$
$$y(0) = 1 \qquad u(0) = 0$$
$$z(0) = 5 \qquad w(0) = 0$$

Задача 5.2.5

Решить линейное дифференциальное уравнение:

$$T^2 \frac{d^2y}{dt^2} + 2\xi T \frac{dy}{dt} + y = 0$$
$$y(0) = 0$$
$$y'(0) = 1$$

для T=10 и ξ=-2, -0.5, 0, 0.5, 5.

Убедиться, что:

При ξ=-2 решение – расходящийся апериодический процесс;
при ξ=-0.5 – решение – расходящийся периодический процесс;
при ξ=0 – решение незатухающие гармонические колебания;
при ξ=+0.5 – решение затухающие колебания;
при ξ=+5 – решение апериодический сходящийся процесс.

Задача 5.2.6

Решить линейное дифференциальное уравнение:

$$T^2 \frac{d^2y}{dt^2} + 2\xi T \frac{dy}{dt} + y = 0$$
$$y(0) = 0$$
$$y'(0) = 1$$

для ξ=2 и T=0.5, 5, 10.

Убедиться, что увеличение коэффициента линейного запаздывания T приводит к уменьшению наклона решения - экспоненты.

Задача 5.2.7

Используя функцию odesolve, решить самостоятельно приведенные ниже дифференциальные уравнения. Построить графики решения:

$$y''+5y'+10y = 5x$$
$$y'(0) = 0$$
$$y(0) = 0$$

$$y'''+50y''+y'y+9y = 0$$
$$y''(0) = 0$$
$$y'(0) = 0$$
$$y(0) = 8$$

Задача 5.2.8

Решить в MathCad дифференциальное уравнение второго порядка

$$T^2\frac{d^2y}{dt^2}+2\xi T\frac{dy}{dt}+y=0$$

при начальных условиях $t_0=0$, $y(t_0)=1$, $dy/dt(t_0)=0$ и заданных значениях параметров $T=10$, $\xi=0.5$.

Задача 5.2.9

Решить в MathCad самостоятельно следующие дифференциальные уравнения:

$$3d^2y/dx^2+5dy/dx+6y=0$$
$$x_0=0,$$
$$y(x_0)=2, dy/dx(x_0)=0.$$
$$x_{лон}=20,\ n=500.$$

$5d^3y/dt^3+9d^2y/dt^2–2dy/dt+8y=1/(t+1).$

$y(0)=0,$

$y`(0)=1,$

$y``(0)=0.$

Задача 5.2.10

Решить самостоятельно приведенные ниже системы уравнений первого порядка

$$y'(x)+y(x)z(x)+8x=0 \qquad u'(t)w(t)+u(t)-3t=0$$
$$z'(x)+8z(x)-10=0 \qquad w'(t)-w(t)+t^2=0$$
$$y(0)=1 \qquad u(0)=0$$
$$z(0)=5 \qquad w(0)=0$$

Задача 5.2.11

Так как алгебраические уравнения с буквенными коэффициентами выше третьей степени символьно не решаются в MathCad, то и линейные дифференциальные уравнения с буквенными коэффициентами выше третьего порядка решены быть не могут.

5.2.12 Контрольные вопросы

1. Проверить для дифференциального уравнения условия теоремы существования и единственности.
2. На какие основные группы подразделяются приближенные методы решения дифференциальных уравнений?
3. В какой форме можно получить решение дифференциального уравнения по методу Эйлера?
4. Каков геометрический смысл решения дифференциального уравнения методом Эйлера?
5. В какой форме можно получить решение дифференциального уравнения по методу Рунге- Кутта?

6. Какой способ оценки точности используется при приближенном интегрировании дифференциальных уравнений методами Эйлера и Рунге-Кутта?

7. Как вычислить погрешность по заданной формуле, используя метод двойного пересчета?

Лабораторная работа №6
Основы работы с пакетом Statistica

Краткие сведения из теории статистики

Статистика позволяет компактно описать данные, понять их структуру, провести классификацию, увидеть закономерности в хаосе случайных явлений. Переменная (variable) - это то, что можно измерять, контролировать или то, чем можно манипулировать в исследованиях. Общее представление о значениях переменной дают описательные статистики:

- минимум;
- максимум;
- среднее (сумма значений переменной, поделенная на число значений)

$$\overline{x} = \frac{1}{n}\sum_{j=1}^{n} x_i, \sum_{j=1}^{n}(\overline{x} - x_i) = 0;$$

- дисперсия (изменяется от нуля до бесконечности, нулевое значение показывает, что переменные постоянны, изменений нет)

$$\sigma^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_i - \overline{x})^2;$$

- стандартное отклонение (это корень квадратный из дисперсии; чем выше дисперсия и стандартное отклонение, тем сильнее разбросаны значения переменной относительно среднего)

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(x_i - \overline{x})^2}.$$

Два вида случайных величин:

- независимая - X (фактор);
- зависимая - y (результативный признак).

Для определения аналитического выражения связи между независимой и зависимой случайными величинами используется регрессионный анализ. Форма связи - это уравнение регрессии.

Виды регрессии:

- парная - исследуются две величины: фактор и результативный признак,
-

$$y=A+Bx;$$

- множественная - исследуются несколько факторов и результативный признак,

$$y=A+A_1x_1+A_2x_2+...+A_nx_n.$$

Для составления прогнозов может быть использовано уравнение регрессии, в котором определяются коэффициенты, называемые параметрами регрессии. Построенная линия уравнения регрессии показывает тенденцию изменения данных и называется линией тренда. Для создания линии тренда используются следующие виды аппроксимации:

- линейная, $y=Ax+B$, где A - тангенс угла наклона прямой, B - точка пересечения с осью ординат;
- логарифмическая, $y=C\ln x+B$, где C, B - константы;
- полиномиальная, $y=C_6x^6+...+Qx+B$, где $C_6+...+Q$, B - константы;
- степенная, $y=Cx^B$, где C, B - константы;
- экспоненциальная, $y=Ce^{Bx}$, где C, B - константы.

Достоверность аппроксимации определяется коэффициентом корреляции R, характеризующим связь между двумя множествами в уравнении линейной регрессии.

Если R лежит в диапазоне от 0,9 до 1, то можно применить линейное уравнение регрессии. Если R близко к -1, то между наблюдаемыми зависимостями существует обратная зависимость. При других видах аппроксимации используется индекс корреляции.

При R, равном или меньше, 0,3 присутствует слабая линейная связь; при R, равном 0,3-0,5, - умеренная линейная связь; при R, равном 0,5-0,7, - средняя или заметная линейная связь; при R,

равном или больше 0,7, - сильная или высокая линейная связь; при R, равном или больше 0,9, - очень сильная или весьма высокая линейная связь. При R, равном единице, - полная функциональная зависимость, все точки на прямой.

Для определения степени влияния факторов на результативный признак используется дисперсия следующих видов:

• общая дисперсия, показывающая степень влияния основных и остаточных факторов:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n};$$

где y_i - значение результативного признака, y - среднее значение результативного признака, n - число наблюдений (значений результативного признака);

• факторная дисперсия, показывающая степень влияния основных факторов:

$$\sigma_ф^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{n};$$

где $\hat{y}_i$ - теоретическое значение результативного признака;

• остаточная дисперсия, показывающая степень влияния остаточных факторов, остаток - это разница между реальными значениями и теоретическими прогнозируемыми:

$$\sigma_0^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-(m+1)};$$

где m - число факторов.

Если существует корреляционная связь, то выполняется соотношение:

$$\sigma^2 = \sigma_ф{}^2 + \sigma_0{}^2; \sigma_ф{}^2 < \sigma^2.$$

При анализе с помощью множественной регрессии используется множественный коэффициент детерминации R^2, называемый также квадратом коэффициента множественной корреляции R и определяющий долю вариации результативного признака, обусловленную изменением факторов:

$$R^2 = \frac{\sigma_Ф^2}{\sigma^2}.$$

При R больше 0,7 (критерий Фишера) считается, что вариация обусловлена влиянием факторов.

Кроме того, для оценки результата аппроксимации используется средняя ошибка аппроксимации:

$$\bar{\varepsilon} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}\cdot 100\%.$$

Если некоторые параметры регрессии малы по сравнению со стандартной ошибкой, то их можно исключить из анализа.

Для определения значимости результата используется p - уровень, характеризующий вероятность ошибки. Обычно принимается p-уровень равным 0,05 или меньше, т.е. ошибка должна быть не более 5%.

6.1 Ознакомление с пакетом Statistica

1. Создайте на диске Z свою папку.
2. Запустите программу «Statistica».

3. В диалоговом окне (ДО) «Statistica Module Switcher» выберите модуль «Basic Statistica» и щелкните мышью на кнопке «Switcher», подтвердите.

4. Создайте файл на диске С в своей папке.

5. Подготовьте таблицу к вводу данных:

- дайте заголовок таблице: дважды щелкните мышью на белом поле таблицы под заголовком файла данных;
- настройте размеры таблицы.

Для этого нажмите на кнопку «Variables» панели инструментов и выберите команду «Delete». В ДО «Delete Variables» укажите диапазон удаляемых переменных: в поле «From variable» укажите переменную, начиная с которой будет проводиться удаление; в поле «To variable» укажите переменную, по которую будет проведено удаление переменных. Нажмите на кнопку «Cases» панели инструментов и удалите лишние случаи аналогичным образом.

Откройте ДО «Variables» (для этого щелкните мышью на имени соответствующей переменной):

- в поле «Name» дайте имя переменной (имя переменной не может быть больше 8 символов);
- в поле «Long name» записывается формула для пересчета переменной;
- в поле «Decimals» задайте число разрядов после запятой;
- в поле «Column width» задайте ширину столбца.

Нажмите на кнопку «ОК» ДО.

6. Сохраните файл.

7. Сохраните файл на внешнем устройстве.

8. Создайте отчет: File, Page/Output Setup. Включите опцию «Text Scrollsheet/

9. Spredsheet», в поле «Ouput» включите опцию «Off», включите флажок «Windows» и нажмите на кнопку «ОК».

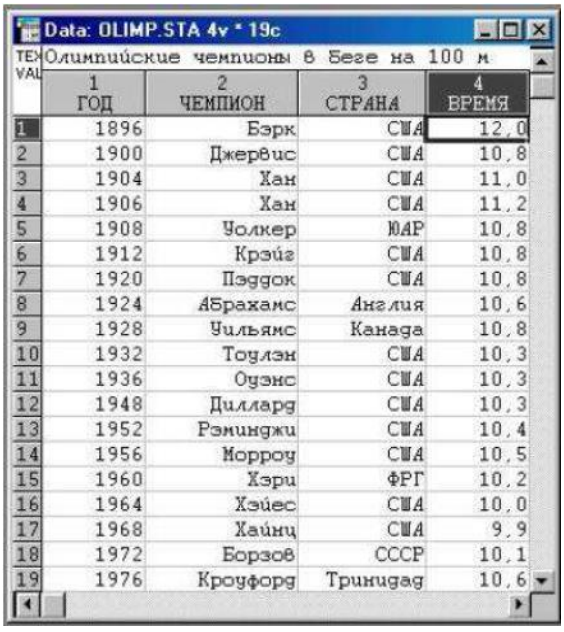
10.Отправьте таблицу исходных данных в окно.

11. Сохраните содержание окна в файл формата .rtf на внешнее устройство.

Ход работы

1. Создайте файл «Cena_rekl.sta» на диске Z в своей папке.
2. Выполните пункты 1-10, приведенные выше.
3. Создайте файл «Olimp.sta» на диске Z в своей папке.
4. Выполните пункты 1-10, приведенные выше.
5. Создайте файл «Kyrs_val.sta» на диске Z в своей папке.
6. Выполните пункты 1-10, приведенные выше.
7. Распечатайте отчет.

Примеры заполненных таблиц представлены на рисунке 6.1.

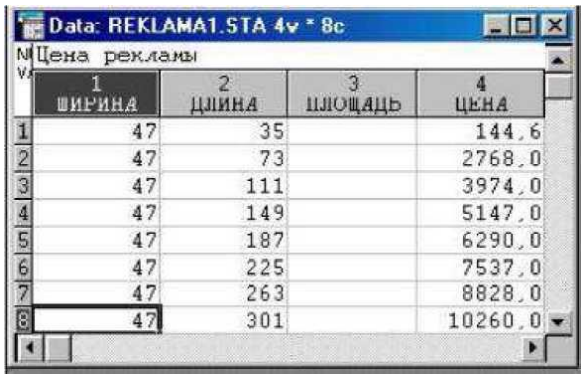


Data: OLIMP.STA 4v * 19c

Олимпийские чемпионы в беге на 100 м

	1 ГОД	2 ЧЕМПИОН	3 СТРАНА	4 ВРЕМЯ
1	1896	Бэрк	США	12,0
2	1900	Джервис	США	10,8
3	1904	Хан	США	11,0
4	1906	Хан	США	11,2
5	1908	Уолкер	ЮАР	10,8
6	1912	Крэйг	США	10,8
7	1920	Пэддок	США	10,8
8	1924	Абрахамс	Англия	10,6
9	1928	Уильямс	Канада	10,8
10	1932	Тоулэн	США	10,3
11	1936	Оуэнс	США	10,3
12	1948	Диллард	США	10,3
13	1952	Рэминджи	США	10,4
14	1956	Морроу	США	10,5
15	1960	Хэри	ФРГ	10,2
16	1964	Хэйес	США	10,0
17	1968	Хайнц	США	9,9
18	1972	Борзов	СССР	10,1
19	1976	Кроуфорд	Тринидад	10,6

А

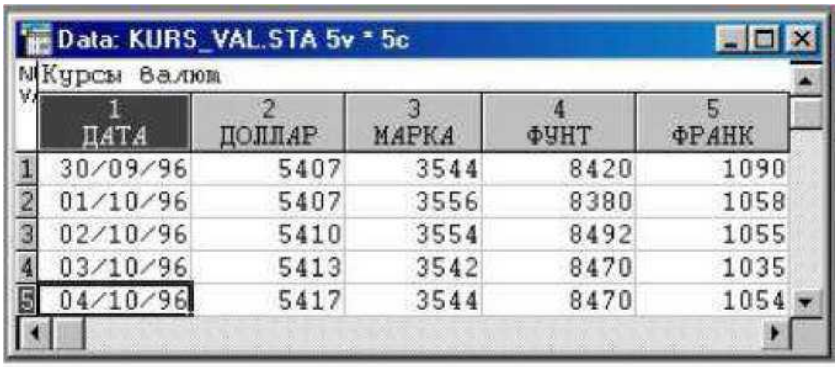


Data: REKLAMA1.STA 4v * 8c

Цена рекламы

	1 ШИРИНА	2 ДЛИНА	3 ПЛОЩАДЬ	4 ЦЕНА
1	47	35		144,6
2	47	73		2768,0
3	47	111		3974,0
4	47	149		5147,0
5	47	187		6290,0
6	47	225		7537,0
7	47	263		8828,0
8	47	301		10260,0

Б



Data: KURS_VAL.STA 5v * 5c

Курсы валют

	1 ДАТА	2 ДОЛЛАР	3 МАРКА	4 ФУНТ	5 ФРАНК
1	30/09/96	5407	3544	8420	1090
2	01/10/96	5407	3556	8380	1058
3	02/10/96	5410	3554	8492	1055
4	03/10/96	5413	3542	8470	1035
5	04/10/96	5417	3544	8470	1054

В

Рисунок 6.1 – Примеры запоненных таблиц в пакете Statistica

6.2 Вычисление описательных статистик. Вычисление корреляций

К числу описательных статистик относятся: среднее, выборочное среднее (mean), выборочная дисперсия (variance), стандартное отклонение (Std.Dev.), медиана, мода, минимальное и максимальные значения (minimum, maximum), размах (range), квантиль (quartiles), выборочный коэффициент асимметрии (skewness), выборочный коэффициент эксцесса (kurtosis).

Формула для выборочного среднего имеет вид:

$$M(N)=(X(1)+X(2)+...+X(N))/N,$$

где N - число наблюдений.

Выборочное среднее является той точкой, сумма отклонений от которой всех рассматриваемых наблюдений равна 0. Среднее значение представляет собой характеристику положения.

Выборочная дисперсия (variance) и стандартное отклонение (Std.Dev.) определяется как:

$$((X(1)-M(N))^2+(X(2)-M(N))^2+...+(X(N)-M(N))^2)/(N-1).$$

Корень квадратный из выборочной дисперсии (variance) есть стандартное отклонение (Std.Dev.).

Мода - это наиболее часто встречающееся значение распределения.

Медиана - это серединное наблюдение в выборке.

Пусть имеется исходная выборка данных:

$$X(1), X(2), X(N).$$

Упорядочим их по возрастанию. Упорядоченные по возрастанию значения называют вариационным рядом:

$$X(1)<X(2)<...<X(N).$$

Серединное значение в этом ряду называется медианой. X(1) - минимальное значение выборки, X(N) - максимальное значение выборки.Разность между максимальным значением выборки и минимальным значением выборки называется размахом.

Асимметрия:

$$Sk = \frac{M[X - M[X]]^3}{\sqrt{D[X]^3}},$$

где M[X] - математическое ожидание, D[X] - дисперсия.

Эксцесс:

$$Ex = \frac{M[X - M[X]]^4}{D[X]^2}.$$

Иногда выборочную асимметрию и эксцесс используют для проверки гипотезы о том, что выборка нормальна. Для нормального распределения Sk=0; Ex=3.

Коэффициент ковариации:

$$Cov(X, Y)=M((X-M[X])(Y-M[Y])).$$

Коэффициент корреляции двух случайных величин X, Y определяется как

$$p(X, Y) = Cov(X, Y)/\sqrt{D(X) \cdot D(Y)}.$$

Корреляция есть нормированная ковариация. Коэффициент корреляции характеризует линейную зависимость между двумя случайными величинами. Коэффициент корреляции является мерой зависимости двух величин. Коэффициент корреляции - это безразмерная величина, значение которого лежит между -1 и +1. Если при возрастании одной величины наблюдается рост другой величины, то говорят о положительной корреляции, если при возрастании одной величины наблюдается тенденция уменьшения

другой величины, то говорят об отрицательной коррелированности величин.

Нулевая корреляция означает, что линейной зависимости между переменными нет. Если X, Y случайные величины, то из равенства 0 коэффициента корреляции следует независимость переменных.

В системе «Statistica» вычисляются выборочные коэффициенты корреляции и ковариации. Выборочные коэффициенты корреляции и ковариации получаются, если математические ожидания и дисперсии заменить их выборочными аналогами.

Ход работы

1. Откротйе «Statistica».
2. В ДО «Statistica Module Switcher» выберите модуль «Basic Statistica» и щелкните мышью по кнопке «Switcher», подтвердите.
3. Откройте файл данных «Cena_rek.sta».
4. Выберите в предлагаемом меню методов верхнюю строчку «Descriptive Statistic» - «Описательные статистики» и нажмите кнопку «ОК».
5. В ДО «Descriptive Statistic»:

- Нажмите на кнопку «Variables» в верхней части окна и выберите для анализа все переменные файла.
- Нажмите на кнопку «More statistics» для выбора расширенного набора описательных статистик, которые следует вычислить.
- Нажмите на кнопку «Detailed descriptive statistic» для просмотра описательных переменных выбранных переменных.
- Оцените близость распределения переменных к нормальному закону. Нажмите на кнопку «Histograms» группы «Distribution». На гистограмму можно наложить плотность нормального распределения.

6. Создайте отчет, в который поместите таблицу с описательными статистиками, гистограмму.
7. В модуле «Описательных статистик» вычислите корреляционную матрицу:

- В меню окна выберите строку «Correlation matrices», OK.
- Нажмите на кнопку «One variables» для определения переменных, OK.
- Нажмите на кнопку «Two list», определите два списка переменных: в первый список поместите переменные «ширина», «площадь», а во второй список - переменную «цена». Нажмите на кнопку «OK».
- Проанализируйте полученные коэффициенты.

8. Поместите в отчет таблицу корреляций.
9. Откройте файл «Olimp.sta»: File, Open data.
10. Выполните пп. 1-8 лабораторной работы.
11. Откройте файл «kyrs_val.sta»: File, Open data.
12. Выполните пп. 1-8 лабораторной работы.

Контрольное задание:

1. Вычислите на калькуляторе выборочное среднее, выборочную дисперсию, выборочное стандартное отклонение для переменных: ширина, площадь, цена. Сравните с вычисленными «Statistica». Сделайте вывод.

2. Проанализируйте, является ли распределение переменных «Площадь», «Цена», «Длина» нормальным по коэффициентам ass,eks, вычисленными «Statistica».

Лабораторная работа № 7
Методы регрессионного анализа в системе Statistica

Множественная линейная регрессия

7.1. Создать новый файл, в котором будут использованы исходные данные, приведенные ниже (рисунок 7.1):

Номер месяца	Объем продаж прохладительных напитков						
	1 S	2 S1	3 A	4 A1	5 T	6 W	7 P
1	8846,4	10132	288,5	200	1675	8,2	101,5
2	8774,7	8846,4	324,2	288,5	1405	9,4	103,8
3	9524,7	8774,7	332,2	324,2	723	15,1	109,2
4	11134,3	9524,7	439,2	332,2	2230	20,9	108,9
5	12239,8	11134,3	98,2	439,2	2615	25,7	112,4
6	8862,1	12239,8	212,8	98,2	3089	26,3	113,1
7	8646,9	8862,1	416	212,8	1741	22,8	114,1
8	11758,5	8646,9	327,7	416	2060	18,7	116
9	11867,2	11758,5	160,6	327,7	1777	15,5	116,6
10	9577,6	11867,2	403,1	160,6	1378,9	11,7	122,6
11	10898,4	9577,6	269,7	403,1	1253,3	5,6	119,5
12	9638,6	10898,4	280,5	269,7	794	6,5	130,6
13	9203,9	9638,6	335,1	280,5	1384,4	8,1	125
14	9231,1	9203,9	169,3	335,1	1392,5	10,3	124,2
15	7334,5	9231,1	206	169,3	2484,4	9,7	130,7
16	7647	7334,5	216,1	206	2777,5	22,4	131,6
17	7839,6	7647	322,2	216,1	3301,9	20,6	133,4
18	9787	7839,6	285,5	322,2	3635,9	26,8	139,1
19	9600	9787	79,2	285,5	3415,9	27,8	142,3
20	7199,9	9600	333,6	79,2	2606,8	18,3	139,9
21	9547,7	7199,9	293,1	333,6	2508	12	144,5
22	10187,5	9547,7	238,5	293,1	2834,1	9,7	143,9
23	9661,2	10187,5	255,4	238,5	2481,8	5,1	148
24	9189,2	9661,2	383,6	255,4	1474,4	2,8	149,3

Рисунок 7.1 – Скришот исходных данных

7.1.1. Командой «All Variable Specs» из меню «Data» ввести переменные: объем продаж за текущий месяц S, объем продаж за предыдущий месяц S1, расходы на рекламу в текущем месяце A, расходы на рекламу в предыдущем месяце A1, число туристов в текущем месяце T, средняя температура воздуха W, индекс розничных цен в текущем месяце P. Имена переменных, например S, вводить в столбец «Name», а комментарий, например «объем продаж за текущий месяц», в столбец «Long Name (label or formula)».

7.1.2. Ввести число наблюдений за 24 месяца, значения переменных, заголовки.

7.2. Запустить множественный регрессионный анализ:

7.2.1. Выполнить команду «Multiple Regression» из меню «Statistics», вкладка «Advanced», кнопка «Variables».

7.2.2. Выбрать зависимые (dependent) и независимые (independent) переменные (рисунок 7.2), «ОК», «ОК».

7.2.3. Нажать кнопку «Cancel» и изменить процедуру на пошаговую «Stepwise» (рисунок 7.3), «ОК».

7.2.4. Выбрать метод на вкладке «Advanced» (рисунок 7.4).

7.2.5. Выбрать параметры пошаговой процедуры (рисунок 7.5), «ОК».

7.2.6. Щелкнуть по кнопке «Summary: Regression Results» на вкладке «Advanced».

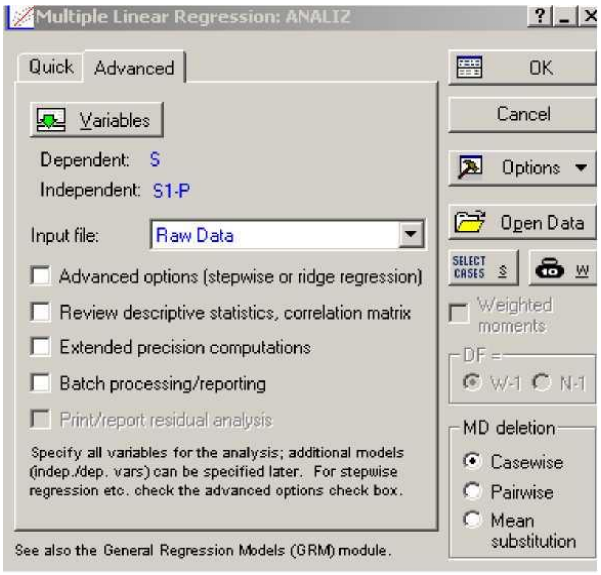


Рисунок 7.2 - Окно множественной регрессии

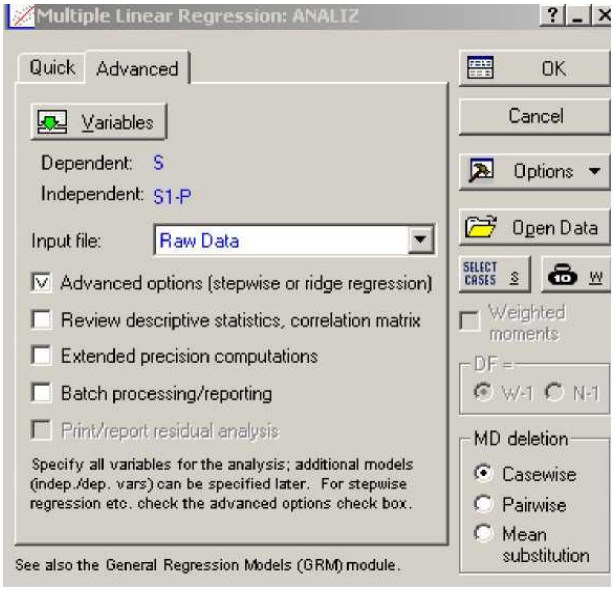


Рисунок 7.3 - Пошаговый анализ

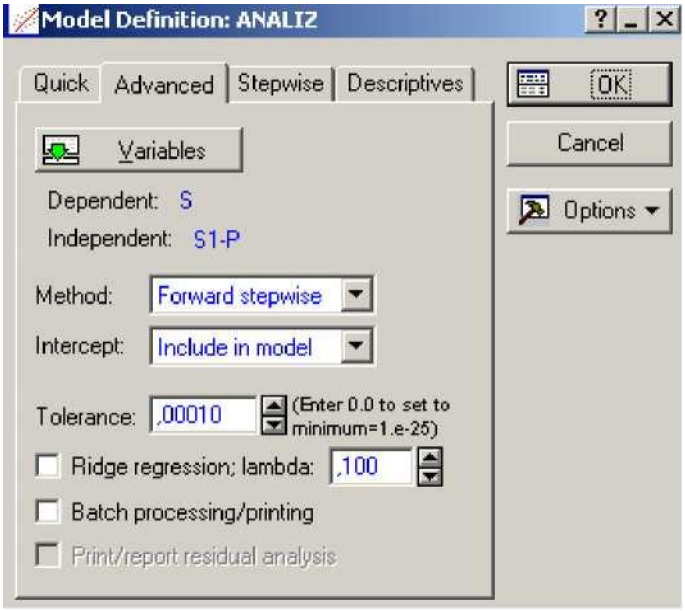


Рисунок 7.4 - Выбор метода

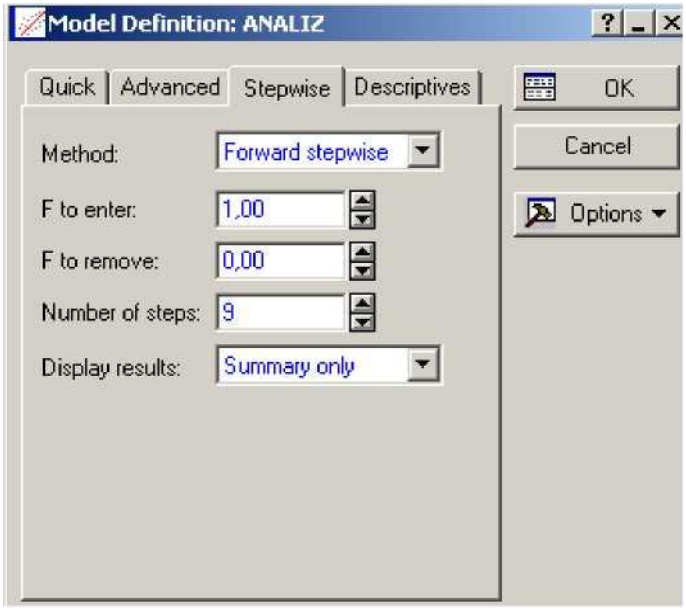


Рисунок 7.5 - Параметры пошаговой процедуры

7.2.7. Вернуться в окно анализа, щелкнув по кнопке «Multiple Regression...».

7.2.8. Произвести анализ остатков, щелкнув по кнопке «Perform residual analysis» на вкладке «Residuals/assumptions/prediction».

7.2.9. Исследовать зависимость между остатками и прогнозом, щелкнув по кнопке «Durbin-Watson statistic» на вкладке «Advanced».

7.2.10. Вернуться в окно анализа, щелкнув по кнопке «Residual Analysis:.».

7.2.11. Щелкнуть по кнопке «Predicted vs. Observed» на вкладке «Scatterplots».

7.2.12. Вернуться в окно анализа, щелкнув по кнопке «Residual Analysis:.».

7.2.13. Щелкнуть по кнопке «Cancel» для того, чтобы вернуться в окно «Model definition», далее «ОК».

7.2.14. Щелкнуть по кнопке «Predict dependent variable» на вкладке «Residuals/assumptions/prediction».

7.2.15. Ввести данные для прогноза на следующий месяц (рисунок 7.6), щелкнуть по кнопке «ОК».

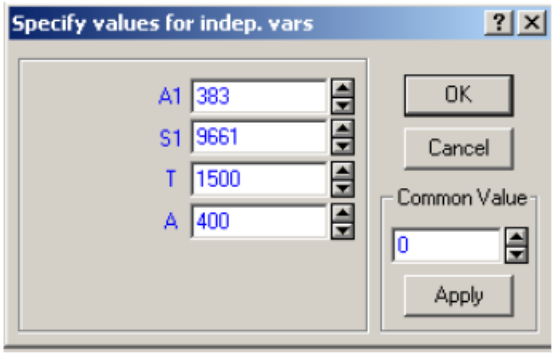


Рисунок 7.6 – Данные для прогноза

7.2.16. Результат представлен на рисунке 7.7. Ожидаемый объем продаж 11473,93.

7.2.17.

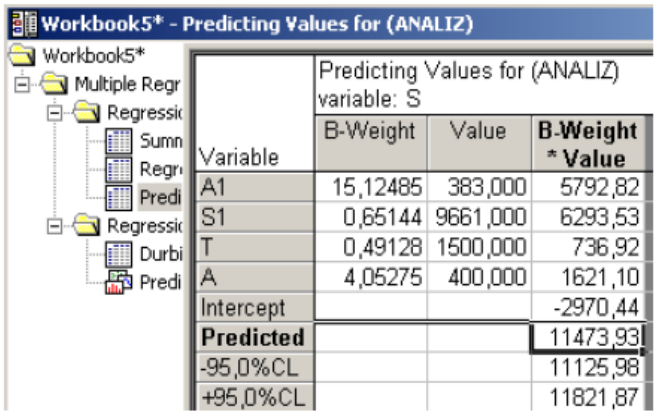


Рисунок 7.7 – Ожидаемый объем продаж

7.3. Сохранить и закрыть все файлы.

Лабораторная работа № 8
Решение практической задачи методами кластерного анализа

Рассмотрим процедуру решения практической задачи методом кластерного анализа в системе STATISTICA.

Задачей кластерного анализа является организация наблюдаемых данных в наглядные структуры. Для решения данной задачи в кластерном анализе используются следующие методы: «Joining (tree clustering)» (иерархические агломеративные методы или древовидная кластеризация), «K-means clustering» (метод К средних), «Two-way joining» (двувходовое объединение).

Разберем принцип проведения кластерного анализа на основе данных, содержащих данные по показателю уровня жизни населения и показателям-аргументам, участвующих в классификации.

Рассмотрим процесс формирования выборок в системе STATISTICA.

1. Из переключателя модулей STATISTICA откройте модуль «Cluster Analysis» (Кластерный Анализ). Высветите название модуля и далее нажмите кнопку «Switch to» (Переключиться в) либо просто дважды щелкните мышью по названию модуля «Cluster Analysis».

2. На экране появится стартовая панель модуля (рисунок 8.1) «Clustering Method» (методы кластерного анализа): «Joining (tree clustering)» (иерархические агломеративные методы или древовидная кластеризация),

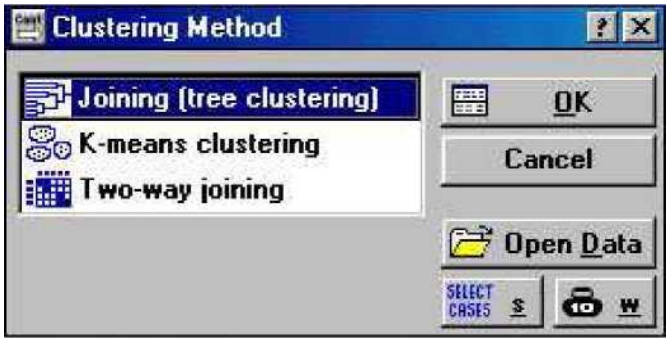


Рисунок 8.1 - Стартовая панель модуля «Clustering Method» (методы кластерного анализа) «Joining (tree clustering)» (иерархические агломеративные методы)

«K-means clustering» (метод К средних), «Two-way joining» (двувходовое объединение). Разберем каждый из этих методов.

Создайте файл date_1.sta, содержащий данные по показателю уровня жизни населения и показателям-аргументам, участвующих в классификации. Откройте файл (Open Data) date_1.sta. После выбора «Joining (tree clustering)» и нажатия «ОК» появляется окно «Cluster Analysis: Joing (Tree Clustering)» (окно ввода режимов работы для иерархических агломеративных методов) (рисунок 8.2), в котором кнопка «Variables» позволяет выбрать переменные, участвующие в классификации. Нажмите на кнопку «Variables» и выберите все переменные «Select All». После соответствующего выбора нажмите «ОК».

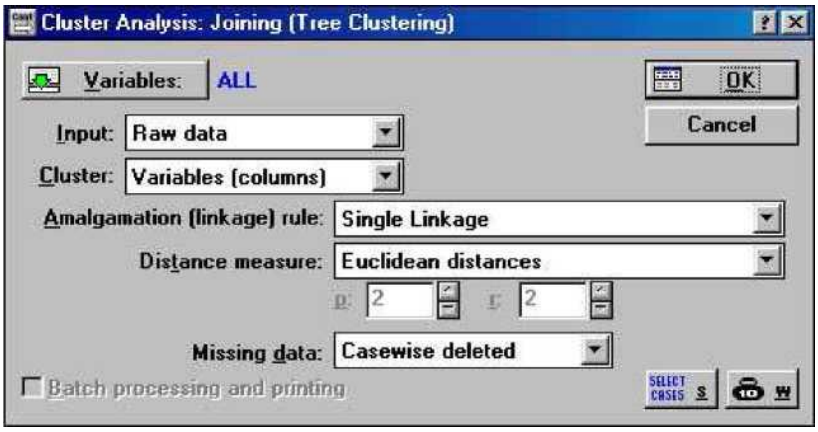


Рисунок 8.2 - «Cluster Analysis: Joing (Tree Clustering)» (окно ввода режимов работы для иерархических агломеративных методов)

Также можно задать «Input» (тип входной информации) и «Cluster» (режим классификации (по признакам или объектам)). Можно указать «Amalgamation (linkage) rule» (правило объединения) и «Distance measure» (метрика расстояний). «Codes for grouping variable» (коды для групп переменной) будут указывать количество анализируемых групп объектов.

«Missing data» (пропущенные переменные) позволяет выбрать либо построчное удаление переменных из списка, либо заменить их на средние значения. «Open Data» позволяет открыть файл с данными. Причем можно указать условия выбора наблюдений из базы данных «Select Cases». Можно задавать веса переменным, выбрав их из списка, нажав клавишей W.

Внесите значения, как показано на рисунке 8.2.

После задания всех необходимых параметров и нажатия кн. «ОК» будут произведены вычисления, а на экране появится окно, содержащее результаты кластерного анализа «Joining Results» рисунок 8.3.

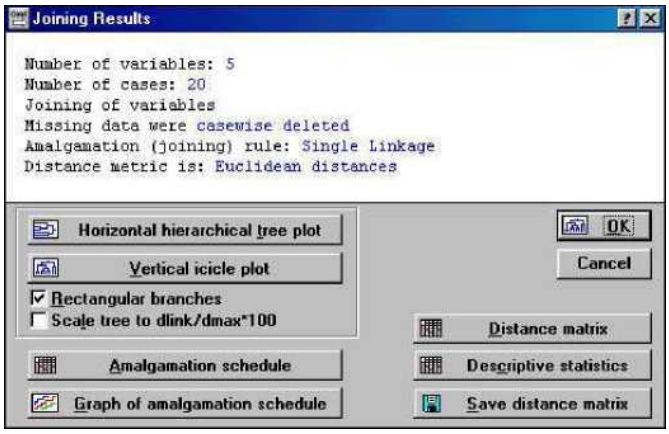


Рисунок 8.3 - Окно, содержащее результаты кластерного анализа «Joining Results»

Вывод результатов и их анализ

Информационная часть диалогового окна «Joining Results Discriminant Function Analisis Results» (результаты анализа кластерных функций) сообщает следующее:

- «Number of variables» - число переменных;
- «Number of cases» - число наблюдений;
- «Missing data were casewise deleted» - осуществлена классификация наблюдений или переменных (зависит от уровня параметра в строке «Cluster» в предыдущем окне настройки);
- «Amalgation (joing) rule» - правило объединения кластеров (название иерархического агломеративного метода, заданного в строке «Amalgation rules», а в предыдущем окне настройки);
- «Distanse metric is» - метрика расстояния (зависит от установки в строке «Distance measure» в предыдущем окне настройки). Пользователь может вызвать на экран горизонтальную и вертикальную диаграмму («Horizontal hierachical plot» или «Vertical icicle plot»). Наиболее традиционным является вертикальное представление (рисунок 8.4).

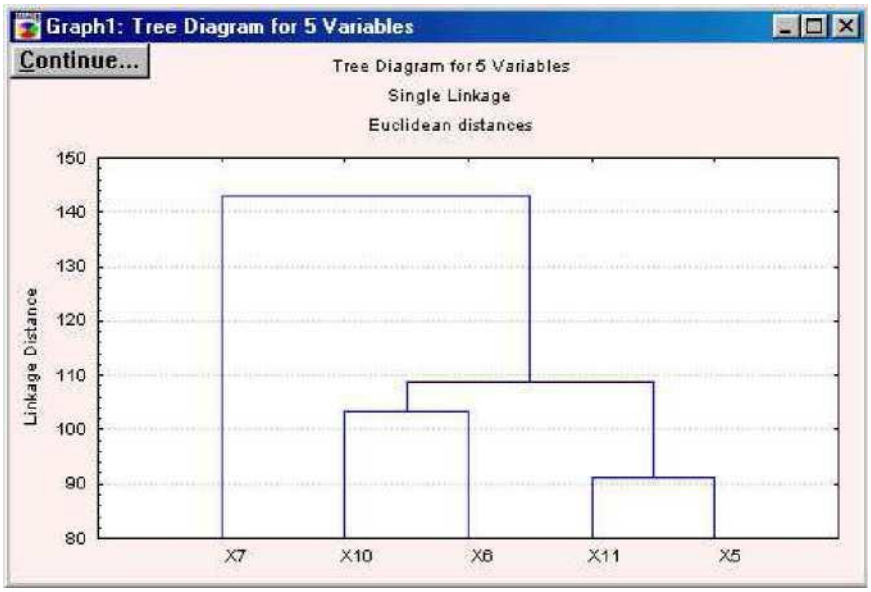


Рисунок 8.4 – Диаграмма «Vertical icicle plot»

Теперь представим себе, что постепенно (очень малыми шагами) вы «ослабляете» ваш критерий о том, какие объекты являются уникальными, а какие нет. Другими словами, вы понижаете порог, относящийся к решению об объединении двух или более объектов в один кластер. В результате, вы связываете вместе всё большее и большее число объектов и агрегируете (объединяете) все больше и больше кластеров, состоящих из все сильнее различающихся элементов. Окончательно, на последнем шаге все объекты объединяются вместе. Когда данные имеют ясную «структуру» в терминах кластеров объектов, сходных между собой, тогда эта структура, скорее всего, должна быть отражена в иерархическом дереве различными ветвями. В результате успешного анализа методом объединения появляется возможность обнаружить кластеры (ветви) и интерпретировать их.

Чтобы вернуться в окно, содержащее другие результаты кластерного анализа, необходимо щелкнуть по «Continue».

Щелчком мыши можно раскрыть строку «Amalgamation schedule», содержащую протокол объединения кластеров (рисунок 8.5).

В заголовке указан иерархический агломеративный метод и метрика расстояния. Таблица может занимать несколько окон.

Amalgamation Schedule (date_22.sta)

Continue... Single Linkage
Euclidean distances

linkage distance	Obj. No. 1	Obj. No. 2	Obj. No. 3	Obj. No. 4	Obj. No. 5
91,07861	X5	X11			
103,1825	X6	X10			
108,6525	X5	X11	X6	X10	
142,8405	X5	X11	X6	X10	X7

Рисунок 8.5 – Протокол объединения кластеров «Amalgamation schedule»

Следующей в окне результатов идет кнопка «Graph of amalgamation schedule». После щелчка, раскрывается окно, содержащее ступенчатое, графическое изображение изменений расстояний при объединении кластеров (рисунок 8.6).

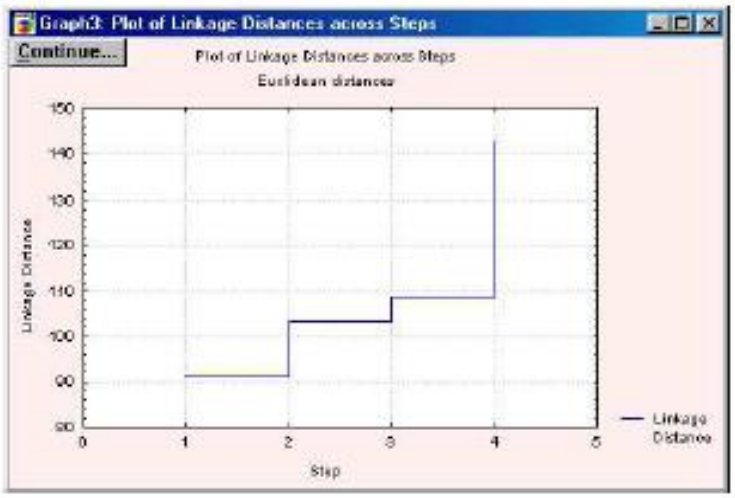


Рисунок 8.6 – Графическое изображение изменений расстояний при объединении кластеров «Graph of amalgamation schedule»

Для просмотра матрицы расстояний необходимо осуществить щелчок на строке «Distance matrix» (рисунок 8.7).

Euclidean distances (date_22.sta)

Continue...	X5	X11	X6	X7	X10
X5	.	91,	148,	143,	109,
X11	91,	.	197,	203,	156,
X6	148,	197,	.	210,	103,
X7	143,	203,	210,	.	149,
X10	109,	156,	103,	149,	.

Рисунок 8.7 - Матрица расстояний

В основном окне результатов классификации имеется строка «Save distance matrix as:» (Сохранить матрицу расстояний как:), позволяющая задать имя файла, в котором будет сохранена матрица расстояний, которая в дальнейшем будет подвергнута обработке.

Отрока «Discriptive statistics» содержит такие важнейшие описательные статистики, как среднее (means) и среднеквадратическое отклонение (standart deviations) для каждого наблюдения. При проведении классификации n объектов по k признакам, для пользователя представляют большой интерес значения этих показателей для каждого признака. Для того чтобы эти характеристики рассчитывались именно по признакам, необходимо вернуться в основное окно настройки параметров и задать в строке «Cluster» значение «variables (columns)».

K-means clustering (метод К средних)

Суть этого метода состоит в следующем: исследователь заранее определяет количество классов (k), на которые необходимо разбить имеющиеся наблюдения, и первые k наблюдений становятся центрами этих классов. Для каждого следующего наблюдения рассчитываются расстояния до центров кластеров, и данное наблюдение относится к тому кластеру, расстояние до которого было минимальным. После чего для этого кластера (в котором увеличилось количество наблюдений) рассчитывается новый центр тяжести (как среднее по каждому показателю) по всем включенным в кластер наблюдениям.

Предположим, имеются гипотезы относительно числа кластеров (по наблюдениям или по переменным). Вы можете указать системе образовать ровно три кластера так, чтобы они были настолько различны, насколько это возможно. Это именно тот тип задач, которые решает алгоритм метода К средних. В общем случае метод К средних строит ровно К различных кластеров, расположенных на возможно больших расстояниях друг от друга.

Ход работы

Создайте файл date_2.sta, содержащий данные по показателю больного населения и показателям-аргументам, участвующих в классификации.

Из стартовой панели модуля (рисунок 8.1) «Clustering Method» (методы кластерного анализа) выберите «K-means clustering» (метод К средних). Откройте файл (Open Data) date_2.sta.

После нажатия «ОК» появляется окно «Cluster Analysis: K-means clustering» (метод К средних) (рисунок 8.8), в котором кнопка «Variables» позволяет выбрать переменные, участвующие в классификации. Нажмите на кнопку «Variables» и выберите все переменные «Select All».

В строке «Cluster» указывается, как ведется классификация: при запуске установлен режим «Variables (colums)» - классифицируются переменные на основании их наблюдений, однако в подавляющем большинстве случаев используется режим «Cases (rows)» - классифицируются наблюдения. Для того чтобы включить режим «Cases (rows)» надо нажать на кнопку в конце строки, после чего в открывшемся окне подвести курсор на надпись «Cases (rows)» и нажать левую кнопку.

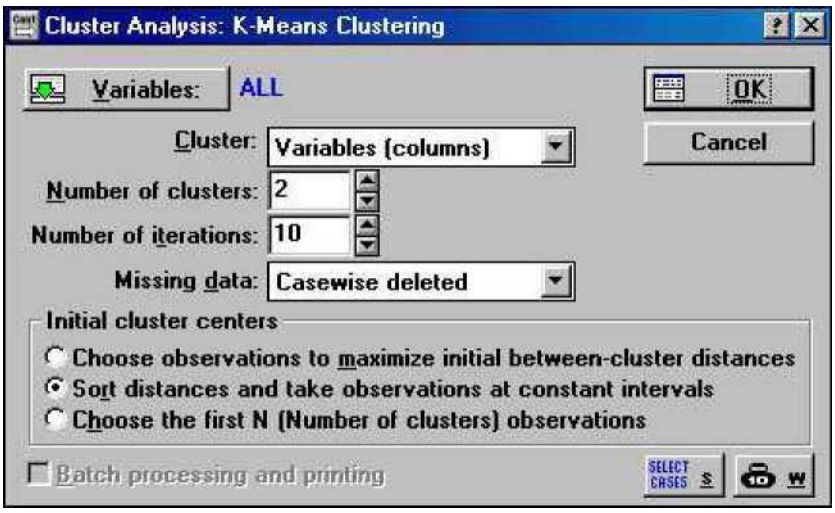


Рисунок 8.8 – Окно «Cluster Analysis: K - means clustering» (метод К средних)

В строке «Number of iterations» указывается количество итераций в расчетах кластеров. Как правило, установленных по умолчанию 10 итераций вполне достаточно. В строке «Missing data» устанавливается режим работы с теми наблюдениями (или переменными, если установлен режим «Variables (columns)» в строке «Cluster»), в которых пропущены данные. Если установить режим «Subsituted by means» (Заменять на среднее), то вместо

пропущенного числа будет использовано среднее по этой переменной (или наблюдению). Переключение в режим «Subsitituted by means» выполняется аналогично переключениям в строке «Cluster». После соответствующего выбора нажмите «OK». Будут произведены вычисления, и появится новое окно: «K - Means Clustering Results» (рисунок 8.9).

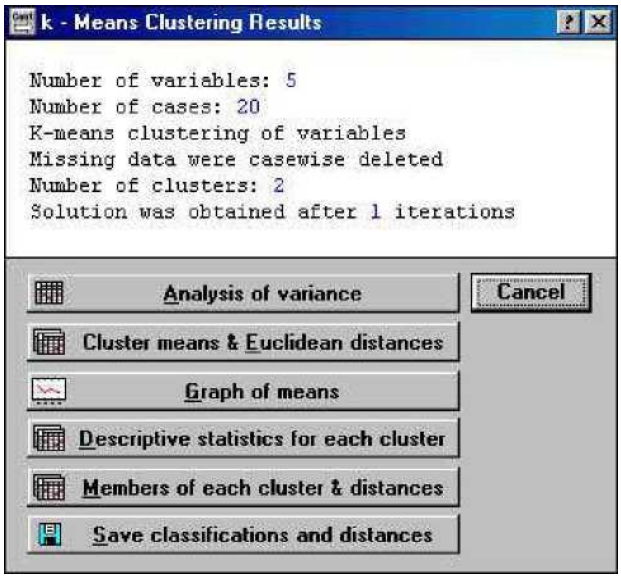


Рисунок 8.9 – Окно «K - Means Clustering Results»

Вывод результатов и их анализ

В верхней части окна (в том же порядке, как они идут на экране):

- Количество переменных;
- Количество наблюдений;
- Классификация наблюдений (или переменных, зависит от установки в предыдущем окне в строке «Cluster») методом K - средних;
- Наблюдения с пропущенными данными удаляются (или: изменяются средними значениями. Зависит от установки в предыдущем окне в строке «Missing data»).
- Количество кластеров;
- Решение достигнуто после «:» итераций.

В нижней части окна расположены кнопки для вывода различной информации по кластерам.

1. Analysis of Variance (анализ дисперсии). После нажатия появляется таблица (рисунок 8.10), в которой приведена

межгрупповая и внутригрупповая дисперсии. Где строки - переменные (наблюдения), столбцы - показатели для каждой переменной: дисперсия между кластерами, число степеней свободы для межклассовой дисперсии, дисперсия внутри кластеров, число степеней свободы для внутриклассовой дисперсии, F - критерий для проверки гипотезы о неравенстве дисперсий. Проверка данной гипотезы похожа на проверку гипотезы в дисперсионном анализе, когда делается предположение о том, что уровни фактора не влияют на результат.

2.



Рисунок 8.10 – Окно «Analysis of Variance» (анализ дисперсии)

3. «Cluster Means & Euclidean Distances» (средние значения в кластерах и евклидово расстояние). Выводятся две таблицы. В первой (рисунок 8.11) указаны средние величины класса по всем переменным (наблюдениям). По вертикали указаны номера классов, а по горизонтали переменные (наблюдения).

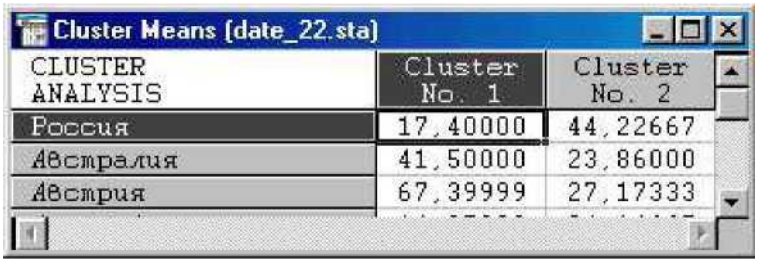


Рисунок 8.11 – Окно «Cluster means»

Во второй таблице (рисунок 8.12) приведены расстояния между классами. И по вертикали, и по горизонтали указаны номера кластеров. Таким образом, при пересечении строк и столбцов указаны расстояния между соответствующими классами. Причем выше диагонали (на которой стоят нули) указаны квадраты, а ниже - просто евклидово расстояние.

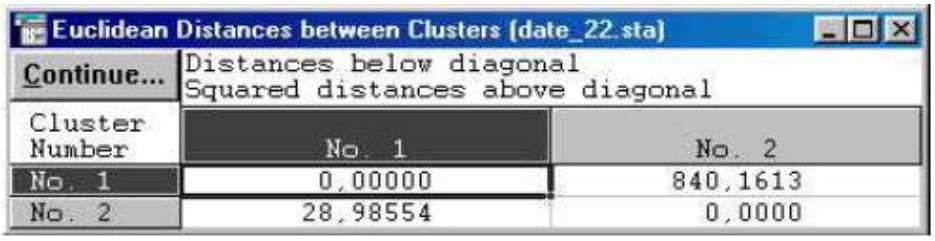


Рисунок 8.12 - Окно «Euclidian Distance between Clusters»

4. «Graph of means» представляет собой графическое изображение (рисунок 8.13) информации содержащейся в таблице, выводимой при нажатии кнопку «Analysis of Variance» (анализ дисперсии). На графике показаны средние значения переменных для каждого кластера.

5.

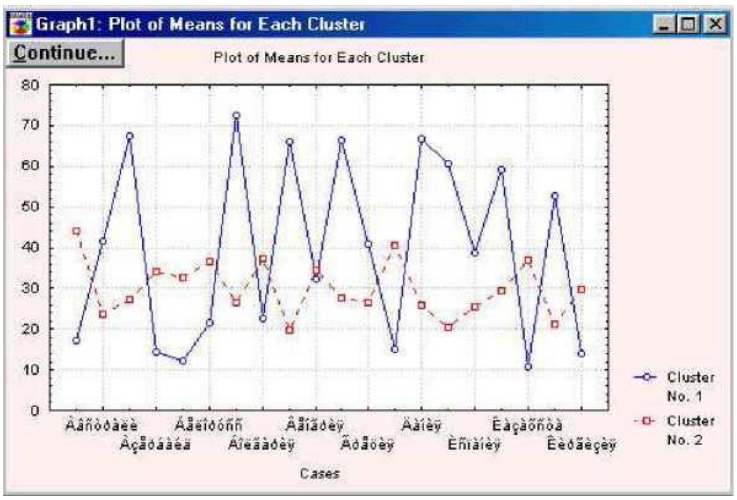


Рисунок 8.13 – Окно «Graph of means»

По горизонтали отложены участвующие в классификации переменные, а по вертикали - средние значения переменных в разрезе получаемых кластеров.

6. «Descriptive Statistics for each cluster» (описательная статистика для каждого кластера). После нажатия этой кнопки выводятся окна, количество которых равно количеству кластеров. В каждом таком окне в строках указаны переменные (наблюдения), а по горизонтали их характеристики, рассчитанные для данного класса: среднее, несмещенное среднеквадратическое отклонение, несмещенная дисперсия.

7. «Members for each cluster & distances». Выводится столько окон, сколько задано классов. В каждом окне указывается общее

число элементов, отнесенных к этому кластеру, в верхней строке указан номер наблюдения (переменной), отнесенной к данному классу и евклидово расстояние от центра класса до этого наблюдения (переменной). Центр класса - средние величины по всем переменным (наблюдениям) для этого класса.

8. «Save classifications and distances». Позволяет сохранить в формате программы STATISTICA таблицу, в которой содержатся значения всех переменных, их порядковые номера, номера кластеров к которым они отнесены, и евклидовы расстояния от центра кластера до наблюдения. Записанная таблица может быть вызвана любым блоком или подвергнута дальнейшей обработке.

Обычно, когда результаты кластерного анализа методом K средних получены, можно рассчитать средние для каждого кластера по каждому измерению, чтобы оценить, насколько кластеры различаются друг от друга. В идеале вы должны получить сильно различающиеся средние для большинства, если не для всех измерений, используемых в анализе (рисунок 8.13), значения переменных пересекаются, но все-таки наблюдаются достаточно четкие различия кластеров. Для более отчетливой группировки следует сократить число параметров. Значения F-статистики, полученные для каждого измерения, являются другим индикатором того, насколько хорошо соответствующее измерение дискриминирует кластеры. Так решение найдено после одной итерации (меньше чем было задано), то можно сделать вывод о том, что итоговая конфигурация является искомой.

В системе реализованы также и другие методы кластеризации, например «Two-way joining», в котором кластеризуются случаи и переменные одновременно. На рисунке 8.14 показан результат кластеризации для данных из файла date_2.sta.

Трудность с интерпретацией полученных результатов этим методом возникает вследствие того, что сходства между различными кластерами могут происходить из-за (или быть причиной) некоторого различия подмножеств переменных. Поэтому получающиеся кластеры являются по своей природе неоднородными. Возможно, это кажется вначале немного туманным; в самом деле, в сравнении с другими описанными методами кластерного анализа (см. Объединение (древовидная

кластеризация) и Метод K средних), двувходовое объединение является, вероятно, наименее часто используемым методом. Однако некоторые исследователи полагают, что он предлагает мощное средство разведочного анализа данных (за более подробной информацией можно обратиться к описанию этого метода у Хартигана (Hartigan, 1975)).



Рисунок 8.14 - Результат кластеризации «Two-way joining» методом

Лабораторная работа №9
Основные сведения о матричной лаборатории MatLab. Справочная система. Основные объекты

Система MatLab предназначена для автоматизации математических расчетов и моделирования процессов. Система располагает готовой коллекцией команд, с помощью которых можно искать корни алгебраических уравнений и экстремумы функций нескольких переменных, строить графики поверхностей, сопровождать текстовую и графическую информацию звуком и многое другое. Если имеющаяся коллекция команд не отвечает запросам исследователя, то возможно написание собственных алгоритмов решения задач. Система понимает программы на языках Паскаль, Си и Фортран.

Текущий документ, в котором пользователь вводит команды, а система выводит результат их исполнения, называется сессией.

Основное окно системы MatLab показано на рисунке 9.1.

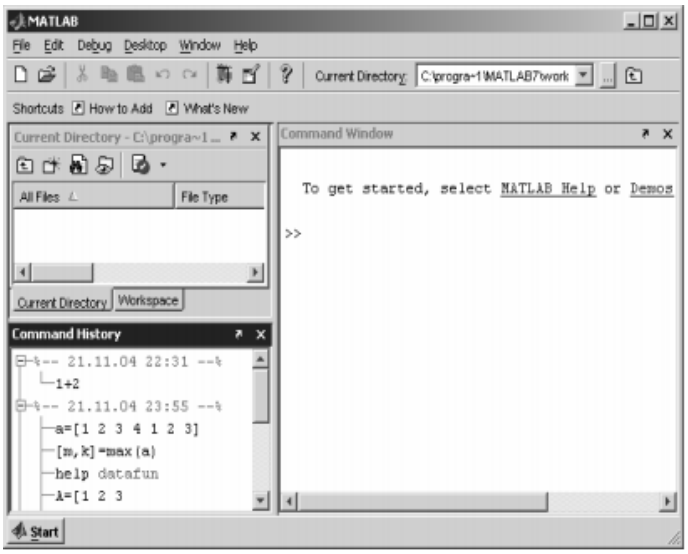


Рисунок 9.1 - Основное окно системы MatLab

Меню системы состоит из пунктов *File* (работа с файлами), *Edit* (редактирование сессии), *Debug* (отладка программ), *Desktop* (отображение панелей инструментов), *Window* (установка свойств окна), *Help* (справка).

Сессии MatLab сохраняются как файлы с расширением m и именуются m-файлами.

Под строкой меню располагается панель инструментов со стандартными кнопками *New m-file* (Создать новый файл), *Open file* (Открыть файл), *Cut* (Вырезать выделенный объект), *Copy* (Копировать), *Paste* (Вставить), *Undo* (Отменить), *Redo* (Повторить), *Help* (Справка).

Кнопка  Simulink запускает одноименное расширение системы

MatLab - средство моделирования систем, построенных из типовых блоков. Поле со списком *Current Directory* (Текущая директория) состоит из набора часто используемых путей к m-файлам пользователей. В поле *Current Directory* отображается текущая директория, в которую по умолчанию сохраняются m-файлы.

Сделайте текущей свою директорию, для этого воспользуйтесь кнопкой  (справа от поля *Current Directory).*

Обычно при запуске MatLa*b* автоматически выводятся плавающие панели инструментов - *Current Directory* (Текущая директория со списком имеющихся m-файлов), *WorkSpace* (Рабочая область для файлов с расширением *mat*, содержащих определения переменных, векторов, матриц и т.п.), *Launch Pad* (Простой доступ к специальным модулям *MatLab*), *Command History* (История ранее введенных команд), *Command Window* (Командное окно).

Если на Вашем ПК MatLab автоматически отображает не все панели, то Вы можете добавить их через пункт меню *Desktop* (Экран) или *View* (Вид).

В командном окне находится командная строка, начинающаяся символом «>>». В командной строке можно вводить команды (как при работе с операционной системой MS DOS).

В примере работы 1 (таблица 9.1) показано, как работать со справкой, и рассмотрены демонстрационные примеры по возможностям MatLab.

План примера 1

1-3. Вызов справки через командную строку. Очистка командного окна.

4-5. Работа с интерактивной справкой.

6. Вывод листинга m-файла.

7-8. Демонстрация настроек основного окна MatLab.

9-10. Демонстрация возможностей ядра MatLab.
11-13. Демонстрация возможностей расширений MatLab.

Таблица 9.1 - Пример работы 1: Подсказка

1. Выведите справку об имеющихся операциях	Наберите в командной строке *help ops* и нажмите
2. Проверьте, есть ли функция *tan* в MatLab	*>>help tan*
3. Очистите командное окно	*>>Clc*
4. Вызовите справочную систему MatLab	*Help* | *MatLab Help*
5. Найдите информацию о команде дифференцирования diff	*Help* | *Search* | *Search Type: Function Name, Search for:*
6. В командном окне введите команду вывода содержимого m-файла	*>>type sym/diff.m*
7.Выведите список демонстрационных примеров	*Help* | *MatLab Help* | *Demos*
8.Посмотрите демонстрационный ролик о рабочей области	*Desktop Environment: WorkSpase* | *Run this demo*
9. Посмотрите демонстрацию основных операций с матрицами	*Matrices: Basic Matrix Operations* | *Run this demo*
10.Вызовите демонстрационный пример обработки графика поверхности	*Graphics: 3D-Plots* | *Run this demo*
11.Посмотрите пример моделирования орбиты спутника с помощью расширения *Mapping Toolbox*	*Toolbox* | *Mapping* | *Animated Satellite Orbits* | *Run this demo*
12. Просмотрите в командном окне, как сделан демонстрационный пример	*>>type orbits*
13. Вызовите в командном окне список всех расширений системы MatLab	*>>ver*

Система MatLab поддержана интерактивной справкой в *html*-формате (таблица 9.1, пункт 4).

В левом подокне имеются вкладки *Contents* (Содержание), *Index* (Указатель), *Search* (Поиск), *Demos* (Примеры), *Favorites* (Избранные гиперссылки) (таблица 9.1, пункт 5).

Справка выводится в правом подокне. В конце справки о команде *diff* приводится список m-файлов, в названии которых упоминается запрашиваемая команда.

Научимся просматривать листинги m-файлов (таблица 9.1, пункт 6).

Система MatLab снабжена обширной коллекцией демонстраций (таблица 9.1, пункт 7).

Коллекция *Desktop Environment* (Установки экрана) состоит из наглядной демонстрации назначения команд из пункта меню *Desktop* (Экран) (таблица 9.1, пункт 8).

Одним из основных объектов MatLab являются матрицы (таблица 9.1, пункт 9).

Посмотрим, как система *MatLab* строит трехмерные графики (таблица 9.1, пункт 10).

Клик по кнопкам *Mesh, Surf, Surfl* выведет разные виды отображения графика поверхности. За кнопкой *Contour* прячутся линии уровня, а за кнопкой *Quiver* - поле градиентов.

Примеры графиков поверхностей можно найти в папке *Gallery*.

Система Matlab включает специализированные пакеты расширения - *Toolbox*. В области математических вычислений к таким пакетам относятся, например, *Statistics Toolbox* (пакет прикладных программ по статистике) и *Optimization Toolbox* (пакет прикладных программ для решения оптимизационных задач и систем нелинейных уравнении).

Примером расширения системы *Simulink* является подсистема *Real Time Windows Target* и *WorkShop* имитационного моделирования в масштабе реального времени, обрабатывающая сигналы датчиков, поступающих в режиме on-line.

Пакет *Fuzzy Logic Toolbox* позволяет моделировать системы с использованием результатов теории нечетких множеств. Пакет идентификации систем *System Identification Toolbox* предназначен

для создания математических моделей динамических систем на основе экспериментально полученных данных.

Matlab располагает пакетами для обработки сигналов и изображений и специфическими пакетами прикладных программ - *Data Analysis & Visualization*. К последним можно отнести, например, пакет по финансово-экономическим расчетам *Financial Toolbox* и пакет анализа географических данных, карт *Mapping Toolbox* (таблица 9.1, пункты 11-13).

Обратите внимание на название демонстрационного примера - *orbits.m.*

Перейдем от просмотра демонстрационных примеров к созданию собственных.

Познакомимся с режимом прямых вычислений, в котором команда набирается в командной строке и после нажатия на клавишу *Enter*, MatLab выполняет введенную команду.

План примера 2

1-2. Работа в режиме прямых вычислений.

3-8. Управление форматом вывода действительных чисел.

9-15. Работа с комплексными числами.

Таблица 9.2 – Пример работы 2: Подсказка

1. Наберите в командной Enter	*>> 1 + 2* *ans = 3*
2. Посчитайте квадрат числа 0.025	*>> 0.025^2* *ans = 6.2500e-004*
3. Установите формат short	*>> format short*
4. Посчитайте квадрат числа 0.5	*>> 0.5^2* *ans = 0.2500*
5. Установите формат long	*File | Preferences... Выберите long в списке Numeric Format.*
6. Разделите 100 на 3	*>> 100/3* *ans = 33.33333333333334*
7. Установите формат short	*>> format short*
8. Перейдите в режим вывода большего	*File | Preferences. Выберите compact в списке Numeric display*

Продолжение таблицы 9.2

9. Найдите квадратный корень из отрицательного числа	>> *sqrt(-25)* *ans = 0 + 5.0000i*
10. Выведите значения константы j	>> *j* *ans = 0 + 1.00001*
11. Задайте значение комплексной переменной z	>> *z=1 - i* *z = 1.0000 - 1.00001*
12. Найдите модуль z	>> *abs(z)* *ans = 1.4142*
13.Определите вещественную часть z	>> *real(z)* *ans = 1*
14. Определите мнимую часть z	>> *imag(z)* *ans = -1*
15. Найдите угол в тригонометрической форме	>> *angle(z)* *ans = -0.7854*

MatLab вычисляет сумму и помещает ее в системную переменную *ans*, определенную по умолчанию для вывода результата.

MatLab может посчитать значение математического выражения, состоящего из чисел, констант, переменных, функций, допустимых операций и специальных знаков.

Константы и переменные в MatLab отдельно не описываются, как, например, в языке Паскаль.

Числа в MatLab считаются константами, имена которых совпадают с их значениями.

MatLab работает с действительными и комплексными числами.

В памяти ПК действительное число представляется в виде трех структурных частей: знак, порядок числа и мантисса числа (набор значащих цифр).

Целая и дробная часть действительного числа разделяются точкой.

Для действительных чисел существуют два способа представления - формата: с фиксированной точкой и с плавающей точкой.

Формат с фиксированной точкой представляет собой десятичное представление вещественного числа с целой и дробной частью, разделенными точкой. Например, 462.7

Формат с плавающей точкой (экспоненциальный формат) представляет собой десятичное представление с одной цифрой в целой части и несколькими в дробной, к числу добавляется справа буква "e" и целое число, обозначающее порядок числа. Например, число 462.7 в формате с плавающей точкой выглядит так: 4.627E2, что означает 4.627*102.

По умолчанию MatLab выводит результаты вычислений в нормализованной форме (одна значащая цифра до десятичной точки и четыре цифры после нее, символ e, порядок числа из трех цифр) (таблица 9.2, пункт 2).

Научимся выводить результаты вычислений в других допустимых форматах.

Short - короткое представление в формате с фиксированной точкой (5 знаков мантиссы) (таблица 9.2, пункты 3, 4). Если результат вычислений не укладывается в формат Short, то он выводится в формате *Short e*.

Short e - короткое представление в формате с плавающей точкой (5 знаков мантиссы и 3 знака порядка).

Long - длинное представление в формате с фиксированной точкой (15 знаков мантиссы).

Long e - длинное представление в формате с плавающей точкой (15 знаков мантиссы и 3 знака порядка).

Hex - представление чисел в шестнадцатеричной системе счисления.

Bank - денежный формат.

Требуемый формат вывода результата вычислений можно установить в диалоговом окне MatLab (таблица 9.2, пунты 5, 6).

Этот формат вывода сохранится для всех последующих вычислений, если только не будет установлен другой формат.

Информацию о форматах можно получить, набрав в командной строке *help format* (таблица 9.2, пункты 7-9).

При вычислении квадратного корня из отрицательного числа никакой ошибки или предупреждения не возникает. MatLab автоматически переходит в область комплексных чисел.

Комплексные числа в алгебраической форме имеют вид $z = a + b \cdot i$, где a - действительная часть комплексного числа, а b - мнимая. Квадратный корень из -1 обозначается в MatLab константой i или j (таблица 9.2, пункты 10-15).

Аргументы функций берутся в круглые скобки!

Для выбора введенных ранее команд можно использовать клавиши <↑>, <↓> или окно *Command History* с историей команд. История команд содержит время и дату каждого сеанса работы.

В примере 2 мы познакомились с константами i и j. Константа - это значение определенного типа, имеющее собственное имя.

Примеры констант MatLab:

Pi - число π.

Inf - бесконечность.

NaN - неопределенность численного результата, например, $\frac{0}{0}$.

Realmax - самое большое число в экспоненциальном формате для Вашего ПК.

Realmin - самое маленькое число в экспоненциальном формате.

В примере 3 показано, как работать с константами и переменными (таблица 9.3).

План примера 3

1-3. Использование констант *MatLab*.

4-8. Определение переменных и вывод информации о них.

9-14. Работа с переменными в окне *WorkSpace*.

Таблица 9.3 - Пример работы 3: Подсказка

1. Узнайте диапазон изменения действительных чисел на Вашем ПК	Используйте константы *realmin* и *realmax*
2. Посмотрите, что получается при делении 1 и -1 на ноль	*>> 1 / 0* Warning: Divide by zero. *ans = Inf*

Продолжение таблицы 9.3

3. Попробуйте найти значение выражения $\frac{\sin(0)}{\ln(1)}$	*>> sin(0)/log(1)* Warning: Divide by zero. *ans = NaN*
4. Выведите список использованных имен переменных	*>> who* *Your variables are: z*
5. Присвойте значение переменной строкового типа	*>> a='abc'* *a = abc*
6. Присвойте значение переменной целого типа	*>> m=25* *m = 25*
7. Задайте координаты вектора	*>> k=[1 2 3]* *k = 1 2 3*
8. Выведите информацию об используемых переменных	*>> whos*
9. Измените координаты вектора k	*Двойной щелчок по имени переменной в окне Workspace*
10. Выведите окно редактирования информации о переменной a	*Клик по кнопке Open (Открыть)* 
11. Удалите из памяти переменную m	*Клик по кнопке Delete (Удалить)* 
12. Сохраните все переменные в свою папку в отдельном файле variables1.mat	*Клик по кнопке*  *Save Workspace As (Сохранить рабочую среду как...)*
13. Удалите из памяти все переменные	*>>clear*
14. Загрузите переменные в память из файла *vari- ablesl.mat*	Клик по кнопке *Load Workspace*  (Загрузить рабочую среду ...)

Сообщение «*Warning: Divide by zero*» означает «Предупреждение: деление на ноль»!

Как во всех средах программирования, в MatLab предусмотрена возможность работы с переменными. Переменная -

сплошной поименованный участок оперативной памяти, предназначенный для хранения информации определенного типа.

Программист в MatLab не должен заранее описывать тип переменной (множество значений, принимаемых величиной). Тип переменной в MatLab определяется присвоенным ей значением. Для того чтобы присвоить переменной z значение 10, достаточно написать z=10. Знак равенства используется в качестве оператора присваивания.

Именем переменной может быть любая последовательность букв и цифр без пробела, начинающаяся с буквы. Строчные и прописные буквы различаются, например *Mz* и *mz* являются двумя разными переменными. Количество воспринимаемых в MatLab символов в имени переменной составляет 31.

MatLab запоминает значения всех переменных, определенных во время сеанса работы. При работе с достаточно большим количеством переменных необходимо знать, какие переменные уже использованы, а какие нет. Для этой цели служит команда *who,* выводящая в командное окно MatLab список используемых переменных (таблица 9.3, пункты 4-7).

Команда *whos* позволяет получить более подробную информацию о переменных в виде таблицы (таблица 9.3, пункт 8).

Обратите внимание на то, что все переменные воспринимаются системой как матрицы разного размера.

Для просмотра переменных можно использовать окно *Workspace*, для перехода в которое следует активизировать одноименную закладку. Данное окно содержит таблицу, аналогичную той, которая выводится по команде *whos*.

Двойной щелчок по строке, соответствующей каждой переменной, приводит к отображению ее содержимого в отдельном окне. Панель инструментов окна *Workspace* позволяет удалить лишние переменные, сохранить и открыть рабочую среду (таблица 9.3, пункты 9-12).

Для освобождения памяти от всех переменных используется команда *clear*. Если после команды указан список переменных (через пробел), то только эти переменные будут удалены из памяти (таблица 9.3, пункт 13), например:

>> clear z

Обратите внимание на исчезновение содержимого окна *WorkSpace* (таблица 9.3, пункт 14).

Часто не очень удобно после каждого присваивания переменной ее значения сразу получать результат данной операции. Поэтому в MatLab можно подавлять вывод результата в командное окно, для этого достаточно завершить оператор присваивания точкой с запятой.

Векторы и матрицы. Основные операции и функции

Все данные MatLab представляет в виде массивов. Массив - упорядоченная, пронумерованная совокупность однородных данных.

Как любой объект MatLa*b*, массив должен иметь свое имя.

Количество размерностей (измерений, индексов, координат), необходимых для обращения к элементу массива, является ключевой характеристикой массива.

Одномерный массив соответствует понятию линейной таблицы (вектора). Элементы одномерного массива располагаются в памяти последовательно друг за другом.

Двумерный массив соответствует понятию прямоугольной таблицы (матрице, набору векторов). Элементы двумерного массива располагаются в памяти друг за другом так, что при переходе от младших адресов к старшим второй индекс изменяется быстрее, чем первый.

N-мерный массив соответствует понятию n-мерного параллелепипеда (тензора). Элементы n-мерного массива располагаются в памяти друг за другом так, что при переходе от младших адресов к старшим наиболее быстро изменяется крайний правый индекс.

Второй ключевой характеристикой массива является его *размер* - общее число элементов. Размер многомерного массива равен произведению размеров образующих его одномерных массивов.

Доступ к элементам массива осуществляется при помощи индексации. В MatLab нумерация элементов массива начинается с единицы.

Важно понять, что вектор, матрица и тензор являются математическими объектами, а одномерные, двумерные или многомерные массивы - способы хранения этих объектов в компьютере.

В примере 4 мы научимся создавать векторы и матрицы обычного и специального вида (таблица 9.4).

План примера 4

1-4. Создание матриц, векторов - строк и векторов - столбцов

5-8. Доступ к элементам матрицы.

9-10. Создание магической и единичной матрицы.

11-12. Создание матриц с единичными и нулевыми элементами.

13-18. Создание матриц с равноотстоящими и случайными элементами.

Таблица 9.4 - Пример работы 4: Подсказка

1. Очистите память	*>>clear*
2. Задайте вектор-столбец b размера 3x1	*>> b=[-1; 0; 1]*
3. Задайте матрицу a размера 2x3	*>> a=[1 2 3; 4 5 6]*
4. Введите вектора - строки k и n	*>> k=[1,2,3];* *>> n=[4,5,6];*
5. Выведите элемент матрицы a, стоящий во второй строке и втором столбце	*>> a(2,2)*
6. Измените значение второй координаты вектора b	*>> b(2,1)=2*
7. Выведите 6-й элемент матрицы a	*>> a(6)* *a(*
8. Замените нулями элементы массива k	*>> k(1:3)=0*

Продолжение таблицы 9.4

9. Создайте магический квадрат размера 3x3	>> *m=magic(3)*
10. Создайте единичную матрицу размера 2x3	>> *e=eye(2,3)*
11. Создайте матрицу, состоящую из одних единиц, размера 4x2	>> *o=ones(4,2)*
12. Сделайте матрицу с нулевыми элементами того же размера, что и матрица e	>> *zeros(size(e))*
13. Создайте вектор из 10 точек, равномерно распределенных на отрезке от 5 до 50	>> *linspace(5,50,10)*
14. Создайте матрицу размера 5x5 из случайных чисел	>> *rand(5)*
15. Создайте матрицу размера 3x4 из случайных чисел	>> *rand(3,4)*
16. Создайте матрицу размера 3x4 из случайных чисел, принадлежащих отрезку от 10 до 100	>> *rand(3,4)*90+10*
17. Выведите одно случайное число	*» rand*
18. Сохраните все переменные в свою папку в отдельном файле variables2.mat	*Клик по кнопке*  *Save Workspace As (Сохранить рабочую среду как...)*

При вводе вектора - столбца его элементы разделяются точкой с запятой (таблица 9.4, пункт 3).

При задании матрицы строки отделяются друг от друга точкой с запятой.

Матрицу a можно рассматривать как вектор-столбец из двух элементов, каждый из которых является вектором - строкой, в которой три элемента. Именно поэтому строки при наборе отделяются точкой с запятой.

При задании матрицы в конце каждой строки можно нажимать на клавишу *Enter*. Попробуйте!

Еще один способ ввода матриц состоит в том, что матрицу можно трактовать как вектор-строку, каждый элемент которой является вектором – столбцом (таблица 9.4, пункт 4). Например, матрицу a можно ввести при помощи команды:

>>a=[[1;4] [2;5] [3;6]]

При задании вектора - строки его элементы можно разделять запятыми или пробелами.

Доступ к элементам массива осуществляется при помощи индексов, заключенных в круглые скобки после имени массива. Доступ к элементам матриц осуществляется при помощи двух индексов – номеров (таблица 9.4, пункты 5, 6)

Расположение элементов матрицы в памяти компьютера определяет еще один способ обращения к ним. Матрица a размера 2 на 3 хранится в виде вектора длины *2*3*, в котором элементы матрицы расположены один за другим построчно: [1 2 3 4 5 6].

Для доступа к элементам матрицы можно использовать один индекс, задающий порядковый номер элемента матрицы в векторе (таблица 9.4, пункты 7, 8).

Система MatLab располагает командами для создания матриц специального вида. Познакомимся с некоторыми из них (таблица 9.4, пункт 9).

Напомним, что сумма элементов магического квадрата в строках, в столбцах и на диагоналях равна одному и тому же числу (таблица 9.4, пункты 10-13). Проверьте!

Часто в задачах и вычислительных экспериментах требуются массивы, состоящие из случайных чисел (таблица 9.4, пункты 14-18). Функция *rand()* создает массивы случайных чисел, равномерно распределенных на отрезке [0,1].

Функция *randn()* генерирует случайные числа, распределенные по нормальному закону с нулевым математическим ожиданием и единичным средним квадратичным отклонением.

В примере 5 показано, как получать информацию о матрицах и преобразовывать матрицы (таблица 9.5).

План примера 5.

1-3. Вывод размерности и размера матриц.

4-5. Определение максимальных и минимальных элементов матриц.

6-7. Вывод диагональных элементов и средних арифметических.

8-12. Нахождение сумм и произведений элементов матрицы.

13-15. Симметричные преобразования матрицы.

16-18. Сортировка и поворот элементов матрицы.

Таблица 9.5 - Пример работы 5: Подсказка

1. Узнайте размерность матрицы a	*>>ndims(a)*
2. Узнайте число строк и столбцов матрицы a	*>>size(a)*
3. Узнайте длину вектора n	*>>length(n)*
4. Найдите минимальный элемент вектора b	*>>min(b)*
5. Найдите максимальный элемент вектора b	*>>max(b)*
6. Выведите элементы, стоящие на главной диагонали матрицы a	*>>diag(a)*
7. Вычислите среднее арифметическое столбцов матрицы a	*>>mean(a)*
8. Найдите сумму элементов в столбцах матрицы a	*>>sum(a)*
9. Найдите сумму элементов в столбцах матрицы a	*>>sum(a,1)*
10. Найдите сумму элементов в строках матрицы a	*>>sum(a,2)*
11. Найдите произведение элементов в столбцах матрицы a	*>>prod(a)*
12. Найдите произведение элементов в строках матрицы a	*>> prod(a,2)*

Продолжение таблицы 9.5

13. Отразите симметрично элементы матрицы a относительно вертикальной оси. Результат присвойте переменной x	>> *x=fliplr(a)*
14. Отразите симметрично элементы матрицы a относительно горизонтальной оси	>> *flipud(a)*
15. Получите все возможные перестановки элементов вектора n	>> *perms(n)*
16. Проведите сортировку элементов по возрастанию в строках матрицы a	>> *sort(x,2)*
17. Поверните элементы матрицы a на 90° против часовой стрелки	>> *rot90(a)*
18. Поверните элементы матрицы a на 180° против часовой стрелки	>> *rot90(a,2)*

Часто необходимо знать не только значение минимального или максимального элемента, но и его индекс. Для этого функции *min* или *max* вызываются с двумя выходными аргументами, например, [s,i]=min(b). В s будет помещено значение минимального элемента, а в i - его индекс. Попробуйте!

Если в векторе или матрице есть несколько максимальных или минимальных элементов, то всегда возвращается номер первого из таких элементов.

Вызов функции *min* или *max* с двумя выходными параметрами для матрицы приведет к выводу самих значений и их номеров в столбцах матрицы (таблица 9.5, пункт 6).

Для нахождения максимума или минимума не по столбцам матрицы, а по строкам предусмотрена следующая форма вызова: max(a,[ ],2) или min(a,[ ],2).

Для того чтобы дополнительно получить номера максимальных или минимальных элементов в строках матрицы используется вызов с двумя выходными аргументами: [s,i]=max(a,[ ],2) или [s,i]=min(a,[ ],2).

Существуют специальные функции, служащие для выделения треугольных частей матрицы:

Tril(A) - результатом является нижняя треугольная часть матрицы A.

Tril(A,k) - результатом является нижняя треугольная часть матрицы A, начиная с k-й диагонали. Значение k=0 обозначает главную диагональ. Положительные значения k обозначают верхние диагонали, отрицательные - нижние.

Triu(A) - результатом является верхняя треугольная часть матрицы A.

Triu(A,k) - результатом является верхняя треугольная часть матрицы A, начиная с k-й диагонали.

Число, стоящее после имени матрицы в списке аргументов ряда функций, определяет номер размерности, для которой вычисляется характеристика. Так, значение 1 (установленное по умолчанию) задает столбцы, значение 2 – строки (таблица 9.5, пункты 11, 12).

Функции *cumsum()* и *cumprod()* возвращают матрицу того же размера с накоплением сумм и произведений элементов. Проверьте!

Результаты встроенных элементарных функций часто подстраиваются к виду аргументов. Если аргументом является массив, то результатом функции будет массив того же размера, но с элементами, равными значению функций от соответствующих элементов исходного массива.

Попробуем выполнить некоторые преобразования с матрицами: переставим, отсортируем, повернем элементы (таблица 9.5, пункты 13-15).

Результатом выполнения команды *perms(n)* является матрица, строки которой есть возможные перестановки элементов вектора n.

Вызов функции *sort()* с двумя выходными аргументами приводит к образованию массива индексов соответствия элементов упорядоченного и исходного массивов (таблица 9.5, пункт 16). Например, результатом команды [s,i]=sort(x,2) будет упорядоченная по строкам матрица x и матрица, в которой на месте элемента стоит его индекс в неупорядоченной матрице x.

Рассмотрим некоторые арифметические операции, используемые в математических выражениях. Напомним, что

числа, переменные, константы или арифметические выражения, над которыми проводится операция, называются операндами.

Операция называется унарной, если она совершается над одним операндом. Если операндов - два, то операция называется бинарной.

В примере 6 будут рассмотрены основные арифметические операции над матрицами и векторами (таблица 9.6).

План примера 6

1-4. Сложение и вычитание векторов.

5-9. Умножение матриц и возведение в степень.

10-13. Деление матриц. Нахождение обратной матрицы. Решение СЛУ.

14-17. Поэлементные операции над матрицами.

18. Транспонирование матрицы.

Таблица 9.6 - Пример работы 6: Подсказка

1. Определите вектора a и b	*>>a=[1 2 3]* *>>b=[4 5 6]*
2. Выполните сложение векторов a и b	*>>a+b*
3. Выполните вычитание векторов a и b	*>>a-b*
4. Увеличьте координаты вектора a на 5	*>>a+5*
5. Определите матрицу c	*>>c=[-1 0; 0 1; 1 1]*
6. Умножьте вектор a на матрицу c	*>>a*c*
7. Умножьте матрицу c на число 5	*>>c*5*
8. Определите квадратную матрицу d	*>>d=[1 2; 3 4]*
9. Возведите матрицу d в куб	*>>d^3*
10. Найдите обратную матрицу для матрицы d. Результат присвойте переменной z	*>>z=eye(size(d))/d*
11. Перемножьте матрицы d и z. Получится единичная матрица	*>>d*z*
12. Решите систему линейных уравнений $\begin{cases} x + y = 3 \\ x - y = -1 \end{cases}$	*>>[1 1; 1 -1]\[3; -1]*
13. Сделайте проверку	*>>[1 1; 1 -1]*[1; 2]*
14. Выполните почленное умножение массивов a и b	*>>a.*b*
15. Возведите координаты вектора b в куб	*>>b.^3*
16. Выполните почленное деление слева направо массивов a и b	*>>a./b*
17. Выполните обратное поэлементное деление справа налево массивов a и b	*>>a.\b*
18. Выполните транспонирование матрицы d	*>> d.'*

Напомним, что при сложении или вычитании - матрицы должны быть одного размера, а при умножении - количество столбцов первой матрицы должно совпадать с количеством строк второй матрицы (таблица 9.6, пункты 5-9).

В MatLab определены операции деления матриц. Деление матриц слева направо обозначается знаком /. Обратное деление матриц обозначается знаком \. Операции деления матриц используются для нахождения обратных матриц и решения систем линейных уравнений (таблица 9.6, пункты 10-13).

Знаки поэлементных операций над векторами и матрицами предваряются точкой (таблица 9.6, пункты 14, 15).

Показателем степени может быть вектор той же длины, что и возводимый в степень. При этом каждый элемент первого вектора возводится в степень, равную соответствующему элементу второго вектора:

Деление соответствующих элементов векторов одинаковой длины выполняется при помощи операции ./. Обратное поэлементное деление (деление элементов второго вектора на соответствующие элементы первого) осуществляется при помощи операции .\ (таблица 9.6, пункты 16-18)

В MatLab парные пустые квадратные скобки [ ] обозначают пустое множество, которое, в частности, позволяет удалять строки и столбцы матрицы. Чтобы удалить строки (столбцы) нужно вместо них указать пустое множество. Например, первую строку квадратной матрицы d можно удалить по команде d(1,:)=[ ], а второй и третий столбцы матрицы u - по команде - u(:,2:3)=[ ]. Попробуйте!

В MatLab есть коллекция матричных операций, используемых в линейной алгебре. Приведем некоторые матричные операции линейной алгебры:

Det(A) - возвращает определитель квадратной матрицы A.

Rank(A) - ранг матрицы A.

Inv(A) - возвращает матрицу, обратную квадратной матрице A.

Eig(A) - возвращает вектор собственных значений матрицы A.

Cross(A,B) - результатом является векторное произведение векторов A и B в трехмерном пространстве.

Решение задач

Решим в MatLab задачу из курса аналитической геометрии. Напомним, что при решении задачи, необходимо вначале продумать алгоритм ее решения, и только потом браться за кодировку.

Для записи на диск текста сессии предназначена команда *diary* <имя_файла>, которая организует запись на диск всех команд и результатов их выполнения.

Если Вы хотите, чтобы файл с текстом сессии оказался в Вашем каталоге, сделайте его текущим!

Команда *Diary off* приостанавливает запись в файл. Команда *Diary on* вновь начинает запись в файл. Команду *Diary off* следует выполнить пользавершению записи на диск текста сессии.

Просмотреть текст сессии в командном окне MatLab можно по команде *Type* <имя_файла>.

Задача 1. Даны координаты вершин пирамиды. A1 (0,2,7), A2(4,2,5), A3(0,7,1), A4(1,5,0). Найти площадь грани $A_1A_2A_3$ и объем пирамиды.

Математическая модель решаемой задачи. Площадь грани $A_1A_2A_3$ можно найти как половину модуля векторного произведения векторов A_1A_2 и A_1A_3.

Объем пирамиды равен $\frac{1}{6}$ части модуля смешанного произведения векторов A_1A_2, A_1A_3, A_1A_4.

$$S = \frac{1}{2}\left|\left[A_1A_2, A_1A_3\right]\right|$$

$$V = \frac{1}{6}\left|A_1A_2, A_1A_3, A_1A_4\right|$$

Дневник сессии в MatLab к задаче 1 представлен на рисунке 9.2.

```
>> % Ведение дневника сессии в файле familia_1.m
>>  diary familia_1.m
>> % Координаты вершин пирамиды
>> a1=[0 2 7];
>> a2=[4 2 5];
>> a3=[0 7 1];
>> a4=[1 5 0];
>> % Координаты векторов
>> % A1A2
>> a1a2=a2-a1
a1a2 =     4     0    -2
>> % A1A3
>> a1a3=a3-a1
a1a3 =     0     5    -6
>> % A1A4
>> a1a4=a4-a1
a1a4 =     1     3    -7
>> % Векторное произведение векторов A1A2 и A1A3
>> vect_pr=Cross(a1a2,a1a3)
vect_pr =    10    24    20
>> % Квадрат модуля векторного произведения векторов
A1A2 и A1A3
>> Kv=sum(vect_pr.^2)
Kv =        1076
>> % Площадь треугольника A1A2A3
>> S=sqrt(Kv)/2
S =   16.4012
>> % Объединение векторов A1A2, A1A3, A1A4 в матрицу
A
>> a=cat(1,a1a2,a1a3,a1a4)
a =     4     0    -2
        0     5    -6
        1     3    -7
>> % Смешанное произведение векторов
>> V_par=det(a)
V_par =   -58
>> % Объем пирамиды

   >> V=abs(V_par)/6
   V =    9.6667
   >> % Сохранение рабочей среды
   >> save familia_1.mat
   >> % Конец записи дневника сессии в файл
   >> Diary off
```

Рисунок 9.2 – Скришот листинга дневника сессии в MatLab

Проверьте решение задачи в среде MatLab!

В рассмотренной задаче нам потребовалось создать квадратную матрицу, строки которой являются координатами векторов A_1A_2, A_1A_3, A_1A_4.

Функция *cat()* организует конкатенацию (объединение) массивов. Первый аргумент функции определяет способ конкатенации. Если значение этого параметра равно 1, то массивы сцепляются как строки (см. задачу 1). Если значение параметра равно 2, то массивы соединяются как столбцы. Например, результатом команды a=cat (2,a1a2,a1a3,a1a4) будет a = 4 0 -2 0 5 -6 1 3 -7

Приведем ряд элементарных математических функций, имеющийся в системе MatLab:

Abs(A) - модуль каждого числового элемента матрицы A.

Exp(A) - возвращает экспоненту для каждого элемента матрицы A.

Log(A), Log2(A), Log10(A) - натуральный, двоичный и десятичный логарифм элементов матрицы A.

Mod(A,B) - возвращает остаток от деления A на B.

Sqrt(A) - результатом является квадратный корень каждого элемента A. Cos(A) - возвращает косинус элементов матрицы A.

Sin(A) - возвращает синус элементов матрицы A.

Tan(A) - возвращает тангенс элементов матрицы A.

Cot(A) - возвращает котангенс элементов матрицы A.

Asin(A) - возвращает арксинус элементов матрицы A.

Round(A) - округляет до ближайшего целого числа элементы матрицы A.

Задача 2. Даны координаты вершин треугольника A(-5,9), B(7,0), C(5,14). Найти длину высоты CD.

Математическая модель. Длину высоты CD можно найти как расстояние от точки C до прямой AB. Расстояние от точки до прямой можно найти, подставив координаты точки в нормальное уравнение прямой. Нормальное уравнение прямой получается из общего уравнения прямой делением коэффициентов на длину нормали - вектора, координаты которого равны коэффициентам при x и y .

Дневник сессии в MatLab к задаче 2 представлен на рисунке 9.3.

```
>> % Ведение дневника сессии в файле familia_2.m
>> diary familia_2.m
>> % Координаты вершин треугольника

>> A=[-5 9];
>> B=[7 0];
>> C=[5 14];
>> % Координаты вектора AB
>> AB=B-A
AB =    12    -9
>> % Общее уравнение прямой AB вида Mx+Ny+K=0
>> M=AB(2)
M =    -9
>> N=-AB(1)
N =   -12
>> K=M*(-A(1))+N*(-A(2))
K =    63
>> % Квадрат нормирующего множителя
>> Kv=M^2+N^2
Kv =   225
>> % Нормирующий множитель
>> Norm=sqrt(Kv)
Norm =    15
>> % Расстояние от точки C до прямой AB
>> abs(M*C(1)+N*C(2)+K)/Norm
ans =    10
>> % Сохранение рабочей среды
>> save familia_2.mat
>> % Конец записи дневника сессии в файл
>> diary off
>> % Просмотр текста файла familia_2.m
>> type familia_2.m
```

Рисунок 9.3 – Скришот листинга дневника сессии в MatLab

Проверьте решение задачи в среде MatLab!

В примере 7 показано, как строить графики (таблица 9.7). Перед построением графика нужно определить множество точек, принадлежащих графику. При построении графика MatLab соединяет данные точки линиями.

План примера 7.

1-3. Построение графика функции одной переменной.

4. Построение двух графиков на одном листе.

5-7. Построение графиков в полярной системе координат.

8-10. Построение трехмерного графика.

11-14. Построение разных видов трехмерных графиков.

Таблица 9.7 - Пример работы 7: Подсказка

1. Задайте значения аргумента x: значения меняются от -1 до 1 с шагом 0.02	*>>x=-1:0.02:1;*
2. Посчитайте значения функции	*>> y=asin(x);*
3. Постройте график	*>> plot(x,y)*
4. Постройте на одном листе графики двух функций. Результат выполнения команд представлен на рисунке 9.4.	*>> x1=-pi/2:0.02:pi/2;* *>> y1=sin(x1);* *>> plot(x,y, 'R', x1,y1,'B')*
5. Задайте значения полярного угла t: значения меняются от 0 до 2π с шагом $\pi/100$	*>> t=0:pi/100:2*pi;*
6. Постройте график кардиоды (рисунок 9.5). Цвет графика - красный, тип линии - двойной пунктир, тип точки - окружность	*>> polar(t,3*(1 +cos(t)),'R:o')*
7. Постройте синюю трехлепестковую розу из звездочек (рисунок 9.6)	*>> polar(t,3*sin(3*t), 'B*')*
8. Создайте квадратную сетку, узлы которой отстоят друг от друга на 0.2, а x и y меняются от -4 до 4	*>> [x,y]=meshgrid([-4:0.2:4]);*
9. Посчитайте значения функции	*>> z=x.^2-y.^2;*
10. Постройте трехмерный график (рисунок 9.7)	*>> plot3(x,y,z)*
11. Постройте график из синих колец (рисунок 9.8)	*>> plot3(x,y,z, 'oB')*
12. Создайте цветной каркас поверхности (рисунок 9.9)	*>> mesh(x,y,z)*
13. Постройте цветную поверхность (рисунок 9.10)	*>> surf(x,y,z)*
14. Постройте цветную поверхность с проекциями (рисунок 9.11)	*>> surfc(x,y,z)*

Параметры *'R'* и *'B'* задают цвета линий: *R (red)* - красный, *-B (blue)* - синий. Допустимыми являются также следующие обозначения цветов:

Y - желтый, *M* - фиолетовый, *C* - голубой, *G* - зеленый, *W* - белый, *K* - черный.

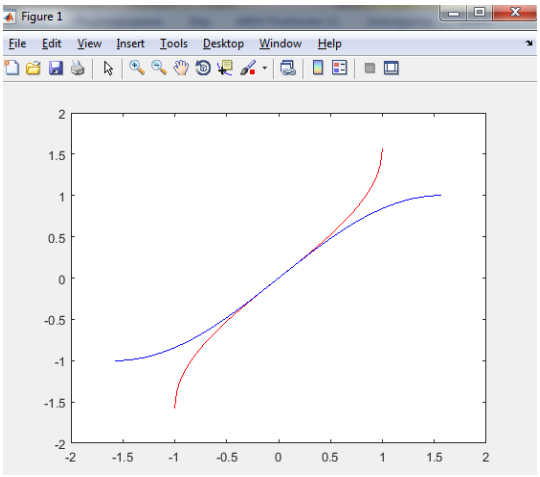


Рисунок 9.4 - Графики функций y=sinx и y=arcsinx

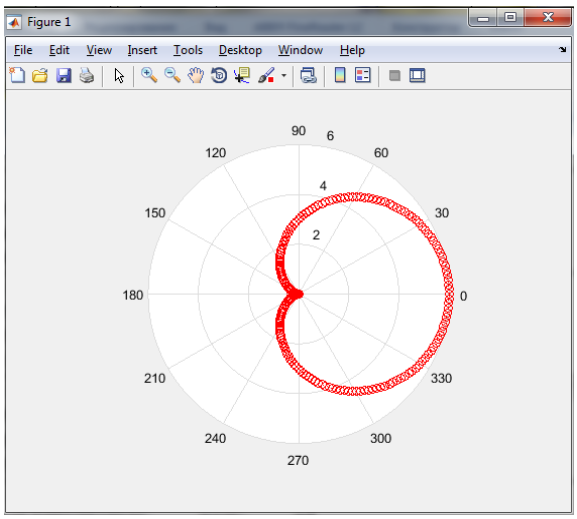


Рисунок 9.5 - Кардиоида, построенная в полярной системе координат

Кроме цвета линии, можно задать тип точки и тип линии. Приняты следующие обозначения:

Тип точки (. точка, *o* окружность, *x* крест, + плюс, * звездочка, *S* квадрат, *D* ромб, *V* треугольник вниз, ^ треугольник вверх, < треугольник влево, > треугольник вправо, *p* пятиугольник, *H* шестиугольник);

Тип линии (- сплошная, : двойной пунктир, -. штрих-пунктир, -- штриховая).

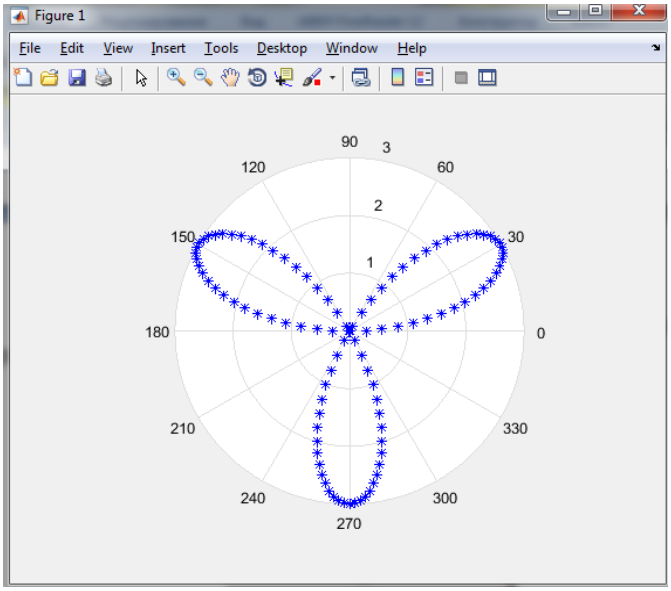


Рисунок 9.6 - Трехлепестковая роза

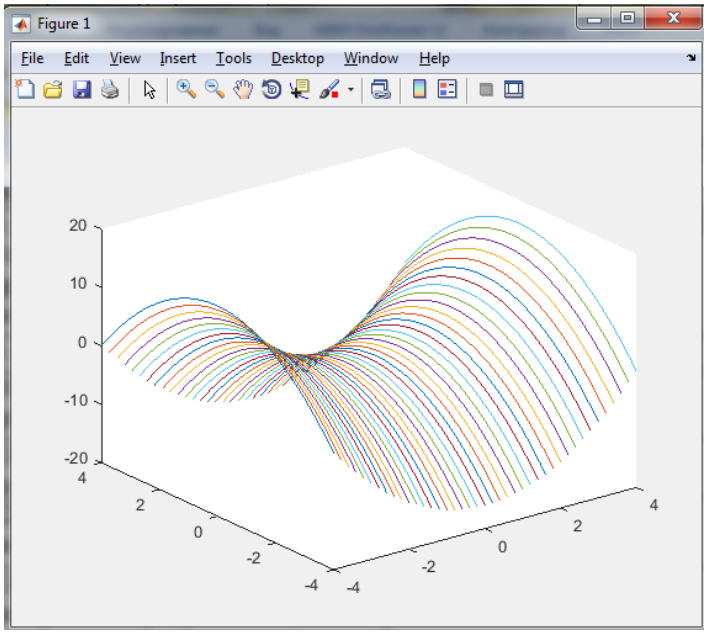


Рисунок 9.7 - Гиперболический параболоид

Для построения трехмерных поверхностей необходимо указать точки из области определения функции двух переменных. Функция *meshgrid()*создает матрицу точек, координаты которых служат аргументами функции (таблица 9.7, пункты 8-10).

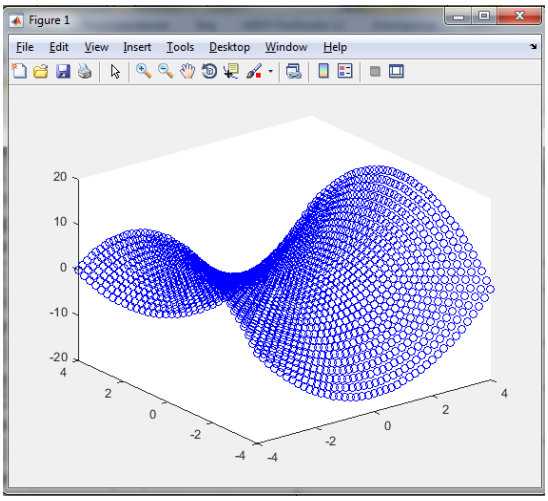


Рисунок 9.8 – Гиперболический параболоид, построенный их синих колец

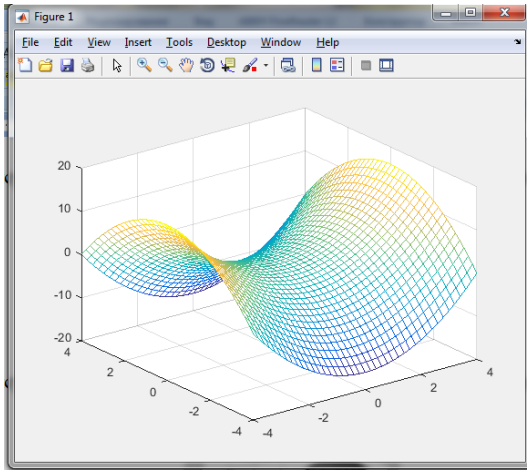


Рисунок 9.9 - Гиперболический параболоид (цветной каркас поверхности)

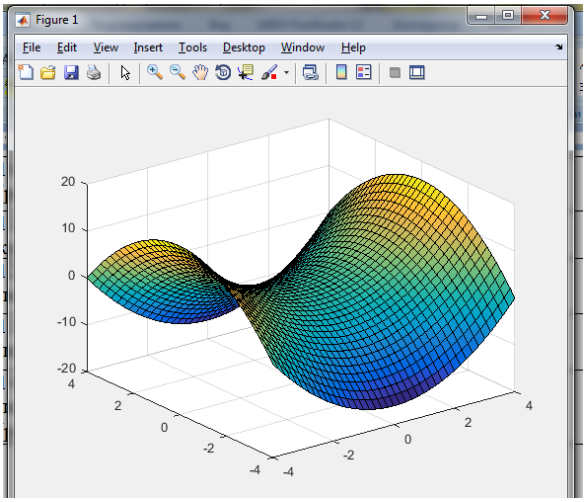


Рисунок 9.10 - Гиперболический параболоид (цветная поверхность)

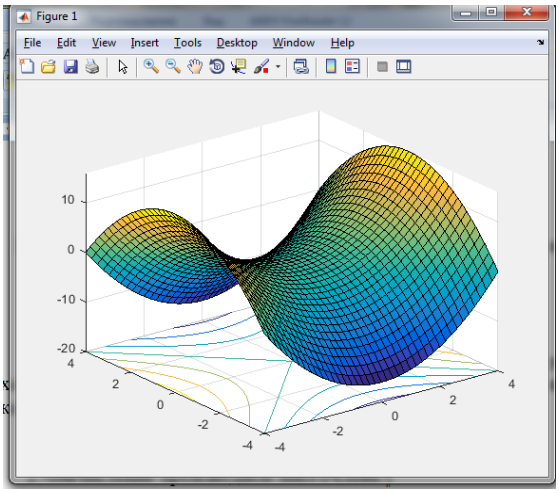


Рисунок 9.11 - Гиперболический параболоид (цветная поверхность с проекциями)

В примере 8 показано, как искать корни алгебраических уравнений, находить производные от многочленов и строить графики производных некоторых функций (таблица 9.8).

План примера 8

1. Нахождение корней уравнения.
2. Вычисление производной многочлена.
3. Построение графика производной функции $y = \frac{x^3}{3}$.

Таблица 9.8 - Пример работы 8: Подсказка

1. Найдите корни уравнения $x^3 - 6x^2 + 11x - 6 = 0$	*>> y=[1,-6,11,-6];* *>> roots(y)* *ans =* *3.0000 2.0000 1.0000*
2. Определите коэффициенты многочлена, являющегося производной функции $y = x^3 - 6x^2 + 11x - 6$	*>> polyder(y)* *ans =* *3 -12 11*
3. Постройте график производной функции $y = \frac{x^3}{3}$ (рисунок 9.12)	*>> x=-2:0.1:2;* *>> y=x.^3/3;* *>> d=diff(y);* *>> plot(d/0.5)*

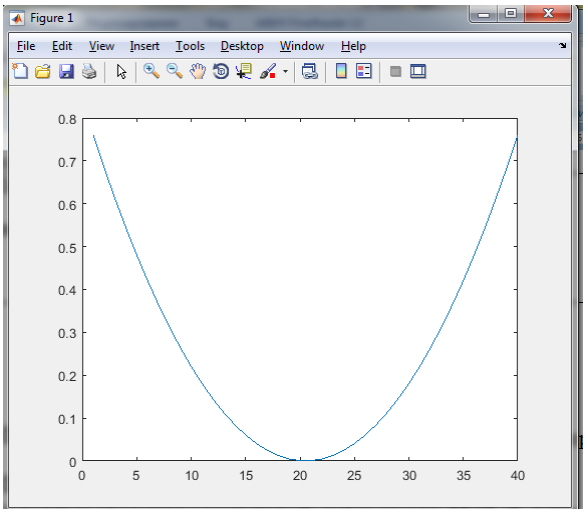


Рисунок 9.12 – График производной функции $y = \frac{x^3}{3}$

Подготовьтесь к ответам на все(!) контрольные вопросы и выполните все(!) контрольные задания.

Контрольные вопросы и задания

1 Основные сведения о матричной лаборатории MatLab. Справочная система. Основные объекты

1. Опишите возможности системы MatLab. Приведите примеры.
2. Что такое сессия?
3. Охарактеризуйте назначение пунктов меню системы MatLab.
4. Что такое m-файлы?
5. Охарактеризуйте назначение пиктограмм на панели инструментов.
6. Что такое текущая директория? Сделайте свой каталог текущей директорией?
7. Как вывести (убрать) плавающие панели инструментов?
8. Объясните назначение панелей *WorkSpace*, *Launch Pad*, *Command History*, *Command Window*?
9. Почему некоторые файлы, создаваемые MatLab, имеют расширение *mat*?
10. Выведите список допустимых операций.

11. Вызовите справку о функции, названной преподавателем.

12. Очистите командное окно.

13. Найдите информацию о команде, названной преподавателем, используя интерактивную справочную систему.

14. Как просмотреть листинг m-файла? Продемонстрируйте.

15. Вызовите демонстрационный ролик о назначении команд из пункта меню *Desktop*.

16. Объясните работу демонстрационного ролика *3D-Plots*.

17. Что такое *Toolbox*?

18. Приведите примеры расширений MatLab в области математических вычислений.

19. Для чего предназначена система *Simulink* и ее расширение *Real Time Windows Target* и *WorkShop*? Покажите демонстрационный пример.

20. Охарактеризуйте назначение нескольких расширений системы и покажите демонстрационные примеры.

21. Как вызвать список всех расширений системы?

22. Как просмотреть текст демонстрационного примера?

23. Что такое режим прямых вычислений?

24. Что такое системная переменная?

25. Каково назначение системной переменной *ans*?

26. Из чего состоит математическое выражение в MatLab?

27. Нужно ли описывать константы и переменные в MatLab?

28. Верно ли, что числа в MatLab считаются безымянными константами?

29. Как в памяти ПК представляется вещественное число? Что такое мантисса? Порядок?

30. Верно ли, что целая часть числа отделяется от дробной запятой?

31. Что представляет собой формат с фиксированной точкой?

32. Что представляет собой формат с плавающей точкой?

33. Что представляет собой нормализованная форма вывода результата вычислений?

34. Как выглядят числа в формате *Short*? *Short e*?

35. Как выглядят числа в формате *Long*? *Long e*?

36. Что представляют собой форматы *Hex* и *Bank*?

37. Приведите два способа изменения формата вывода результатов вычислений.

38. Как вывести информацию обо всех имеющихся форматах?

39. Как перейти в режим более плотного отображения строк на экране?

40. Можно ли в MatLab извлекать корни из отрицательных чисел? Что при этом происходит?

41. Как в MatLab обозначается квадратный корень из -1?

42. Как задать значение комплексной переменной?

43. Найдите модуль комплексного числа, вещественную и мнимую части.

44. Определите угол в тригонометрической форме комплексного числа.

45. Как скопировать в командную строку введенную ранее команду? Приведите два способа.

46. Что такое константа? Каково значение констант *Inf*, *NaN*, *Realmax* и *Realmin*?

47. Что означает сообщение: «*Warning: Divide by zero*»?

48. Что такое переменная? Тип переменной?

49. Как задается тип переменной в MatLab?

50. Как выглядит оператор присваивания в MatLab?

51. Что служит именем переменной в MatLa*b*?

52. Верно ли, что в именах переменных большие и маленькие буквы не различаются?

53. Каково максимальное количество символов в имени переменной, воспринимаемых системой?

54. Выведите список используемых переменных.

55. Выведите всю информацию об используемых переменных.

56. Измените значение переменной, используя окно *Workspace*.

57. Как удалить переменную из памяти? Приведите два способа.

58. Сохраните используемые переменные и их значения в отдельный файл.

59. Удалите из памяти значения всех переменных и считайте их из файла.

60. Как подавить вывод результата операции присваивания?

2 Векторы и матрицы. Основные операции и функции

61. Что такое массив?

62. Что такое размерность и размер массива?

63. Что такое одномерный, двумерный и n-мерный массивы? Как располагаются в памяти элементы таких массивов?

64. Как организуется доступ к элементам массива?

65. Определите в MatLab матрицу, вектор-строку и вектор-столбец, указанные преподавателем.

66. Охарактеризуйте три способа ввода матриц.

67. Чем отделяются при вводе элементы строк? Столбцов?

68. Распечатайте, а затем измените значение указанного элемента матрицы.

69. Выведите элемент матрицы по его порядковому номеру.

70. Замените нулями все элементы матрицы.

71. Создайте магическую матрицу. Что это такое?

72. Создайте единичную матрицу указанного размера.

73. Создайте матрицу, состоящую из единиц и нулей.

74. Создайте вектор из 20 точек, равномерно распределенных на указанном отрезке.

75. Создайте матрицу указанного размера из случайных чисел, принадлежащих заданному диапазону.

76. Выведите одно случайное число.

77. Для чего используется функция *randn()*?

78. Распечатайте размерность матрицы.

79. Выведите число строк и столбцов матрицы.

80. Выведите длину вектора.

81. Найдите минимальный и максимальный элемент вектора.

82. Распечатайте номер строки и столбца, в которых находится минимальный (максимальный) элемент матрицы.

83. Выведите максимальные и минимальные элементы по строкам и по столбцам матрицы.

84. Выведите элементы, стоящие на главной диагонали. Выведите элементы, стоящие выше и ниже главной диагонали.

85. Найдите среднее арифметическое столбцов матрицы.

86. Вычислите сумму элементов по строкам и по столбцам матрицы.

87. Вычислите произведение элементов по строкам и по столбцам матрицы.

88. Что возвращают функции *cumsum()* и *cumprod()*?

89. Проведите симметрию матрицы относительно вертикальной оси.

90. Проведите симметрию матрицы относительно горизонтальной оси.

91. Получите все возможные перестановки элементов {1, 2, 3}.

92. Проведите сортировку по возрастанию в строках матрицы.

93. Поверните элементы матрицы на 90° и 180° против часовой стрелки.

94. Что такое операнды?

95. Какая операция называется унарной, а какая - бинарной?

96. Выполните сложение и вычитание векторов.

97. Прибавьте к элементам вектора число. Умножьте матрицу на число.

98. Умножьте вектор на матрицу. Возведите матрицу в степень.

99. В чем смысл операций деления матриц? Приведите примеры.

100. Создайте обратную матрицу к заданной.

101. Решите систему линейных уравнений $\begin{cases} 2x - y - z = 4 \\ 3x + 4y - 2z = 11. \\ 3x - 2y + 4z = 11 \end{cases}$

102. Выполните почленное умножение указанных массивов.

103. Возведите все элементы матрицы в заданную степень.

104. Возведите первый элемент вектора в квадрат, второй - в куб и т.д.

105. Выполните поэлементное деление двух массивов справа налево и слева направо.

106. Выполните транспонирование матрицы.

107. Удалите указанные строки и столбцы матрицы.

108. Посчитайте определитель и ранг матрицы.

109. Найдите собственные значения матрицы.

110. Найдите векторное произведение векторов.

3 Решение задач

111. Как записать на диск текст сессии?

112. Как приостановить запись в файл? Как возобновить запись в файл?

113. Как просмотреть текст дневника сессии?

114. Как сохранить рабочую среду с помощью команды?

115. Что такое конкатенация матриц? Как ее провести?

116. Каков смысл первого аргумента функции *cat()*?

117. Найдите модуль и экспоненту числа.

118. Вычислите логарифм с указанным основанием от указанного числа.

119. Найдите остаток от целочисленного деления двух чисел.

120. Найдите значение тригонометрических функций при указанном аргументе.

121. Найдите значение обратных тригонометрических функций при указанном аргументе.

122. Вычислите квадратный корень числа.

123. Округлите элементы вектора до ближайших целых чисел.

124. Постройте график функции $y = \cos x$.

125. Постройте графики функций $y = \cos x$ и $y = \arccos x$.

126. Как изменить цвет графика? Назовите константы цветов.

127. Как изменить тип линии графика? Назовите условные обозначения.

128. Как изменить тип точки графика? Назовите условные обозначения.

129. Постройте графики функций $\rho = \dfrac{1}{2 - \sin\varphi}$ и $\rho = \dfrac{2}{1 - \sqrt{2}\cos\varphi}$ в полярной системе координат.

130. Постройте эллиптический параболоид $z = 2x^2 + 4y^2$. Сделайте цветной каркас, каркас из черной кольчуги, цветную поверхность, цветную поверхность с проекциями.

131. Найдите корни уравнения $x^2 - 5xx + 6 = 0$.

132. Найдите производную функции $y = x^2 - 5xx + 6$.

133. Постройте график производной функции $y = \cos x$.

Упражнения для собеседования

- Даны координаты вершин пирамиды. $A_1(0,2,7)$, $A_2(4,2,5)$, $A_3(0,7,1)$, $A_4(1,5,0)$.

1. Найти площадь грани $A_1A_2A_3$ и объем пирамиды.
2. Найти угол между ребрами A_1A_2 и A_1A_4.
3. Найти уравнение ребра A_1A_4.
4. Найти уравнение плоскости $A_1A_2A_3$.
5. Найти уравнение высоты, опущенной из вершины A_4 на грань $A_1A_2A_3$.
6. Найти длину высоты, опущенной из вершины A_4 на грань $A_1A_2A_3$.
7. Показать, что векторы A_1A_2, A_1A_3 и A_1A_4 образуют базис и найти координаты вектора A_2A_3 в этом базисе.

- Даны координаты вершин треугольника A(- 5,9), B(7,0), C (5,14).

1. Найти уравнения сторон AB и BC их угловые коэффициенты.
2. Найти угол B в радианах и градусах.
3. Найти уравнение биссектрисы внутреннего угла B.
4. Найти уравнение высоты CD .
5. Найти длину высоты CD.
6. Найти уравнение медианы AE .
7. Найти координаты точки пересечения медианы AE с высотой CD.

- Дана система уравнений $\begin{cases} 5x + 8y + 3z = -18 \\ 3x - 2y + 6z = 7 \\ 2x + y - z = -5 \end{cases}$.

1. Решить систему линейных уравнений методом Крамера.
2. Решить систему линейных уравнений, используя обратную матрицу.

Индивидуальные задания

Опишите постановку задачи, создайте математическую модель ее решения, решите задачу в MatLab, проверьте правильность полученных результатов.

1. Вычислить значение определенного интеграла методом прямоугольников:

$$z=\int_a^b \frac{\ln^2 x}{x}dx \approx \frac{b-a}{n}(y_1+y_2+\dots y_n), y_i=\frac{\ln^2 x_i}{x_i}, i=1,2,\dots,n,$$

$$x_{i+1}=x_i+\Delta x, a=1, b=4, n=50, \Delta x=\frac{b-a}{n}.$$

2. Вычислить значение определенного интеграла методом трапеций:

$$z=\int_a^b \frac{\sqrt{x^2-0{,}16}}{x}dx \approx \frac{b-a}{n}\left(\frac{y_0+y_n}{2}+y_1+y_2+\dots+y_{n-1}\right), y_i=\frac{\sqrt{x_i^2-0{,}16}}{x_i},$$

$$i=0,1,\dots,n, x_{i+1}=x_i+\Delta x, a=1, b=2, n=100, \Delta x=\frac{b-a}{n}.$$

3. Вычислить значение определенного интеграла по формуле Симпсона:

$$z=\int_b^b \frac{dx}{x} \approx \frac{b-a}{6m}(y_0+y_{2m}+2(y_2+y_4+\dots+y_{2m-2})+4(y_1+y_3+\dots+y_{2m-1})),$$

$$y_i=\frac{1}{x_i}, i=0,1,\dots,2m, x_{i+1}=x_i+\Delta x, a=1, b=2, m=20, \Delta x=\frac{b-a}{2m}.$$

4. Вычислить спектр фаз на выходе цепи:

$$\varphi(t) = \omega_H t_i + m_f \sin \Omega t_i - \operatorname{arctg} \frac{2Q\Delta\omega_H}{\omega_0} \cdot \cos \Omega t_i,$$

$$t_{i+1} = t_i + 2, i = 1,2,\ldots,10, \Delta\omega_H = m_f \cdot \Omega, \Omega = \begin{cases} 10{,}5 \cdot 10^3, i < 5 \\ 15{,}5 \cdot 10^3, i \geq 5 \end{cases},$$

$$t_1 = 3, \omega_H = 10^4, m_f = 5, Q = 55, \omega_0 = 10^4.$$

5. Вычислить изменения частоты на выходе цепи:

$$\omega = \omega_H + m_f \Omega \cos \Omega t_i + \frac{b\Omega \sin \Omega t_i}{1 + b^2 \Omega^2 \cos^2 \Omega t_i},$$

$$b = \frac{2Q\Delta\omega_H}{\omega_H}, \Delta\omega_H = m_f \cdot \Omega, t_{i+1} = t_i + 3, i = 1,2,\ldots,8, \omega_H = \begin{cases} 1{,}1 \cdot 10^6, i < 4 \\ 2{,}5 \cdot 10^5, i \geq 4 \end{cases},$$

$$t_1 = 3, \omega_H = 10, m_f = 4, Q = 50, \Omega = 10^3.$$

6. Вычислить спектральные составляющие:

$$C_K = \frac{2Et_u}{T} \cdot \frac{\sin\left(\frac{2\pi}{T} \cdot \frac{t_u}{2} \cdot K\right)}{\left(\frac{2\pi}{T} \cdot \frac{t_u}{2} \cdot K\right)}; K = 1,2,3,4,5; E = \begin{cases} 1, K < 2 \\ 2{,}2 \leq K < 4; T = 5; t_u = 1 \\ 3, K \geq 4 \end{cases}$$

7. Вычислить значение функции:

$$Z = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{x_i^j}{j!}; x_{i+1} = x_i + 1; N = 5; M = 6; x_1 = 3.$$

8. Вычислить значения функции:

$$S=\begin{cases}\dfrac{a+b}{e^x\cos x}, x<2{,}8\\ \dfrac{a+b}{x+1}, 2{,}8\le x<6\\ e^x+\sin x, x\ge 6\end{cases}; a=2{,}6; b=-0{,}39; x\in[0;7]; \Delta x=0{,}5.$$

9. Вычислить значение определенного интеграла методом прямоугольников:

$$z=\int_a^b \frac{1}{x^2}\sin\frac{1}{x}dx\approx\frac{b-a}{n}(y_1+y_2+\dots y_n); y_i=\frac{1}{x_i^2}\sin\frac{1}{x_i}; n=50; a=1; b=2{,}5$$

.

10. Вычислить значение определенного интеграла методом трапеций:

$$z=\int_a^b \frac{1}{\sqrt{9+x^2}}dx\approx\frac{b-a}{n}\left(\frac{y_0+y_n}{2}+y_1+y_2+\dots+y_{n-1}\right); y_i=\frac{1}{\sqrt{9+x_i^2}};$$
$$n=100; a=0; b=2.$$

11. Определить максимальный и минимальный элементы массива A(25) и их порядковые номера. Элементы массива - случайные числа из диапазона от 0 до 100.

12. Вычислить сумму элементов матрицы A(4x5). Элементы матрицы - случайные числа из диапазона от 0 до 100.

13. Найти наименьшие элементы и номера строк и столбцов, в которых они расположены, целочисленной матрицы A(6x5). Элементы матрицы A - случайные целые числа из диапазона от 0 до 100.

14. Вычислить количество отрицательных элементов каждого столбца для матрицы B(6x5). Элементы матрицы - случайные целые числа из диапазона от -100 до 100.

15. Вычислить сумму и число положительных элементов каждого столбца матрицы A(8x8), находящихся над главной

диагональю. Элементы матрицы - случайные целые числа из диапазона от -100 до 100.

16. Вычислить сумму и число положительных элементов каждого столбца для матрицы B(10x15). Элементы матрицы - случайные целые числа из диапазона от -100 до 100.

17. Вычислить суммы и произведения каждой строки матрицы A(6,5). Элементы матрицы - случайные целые числа из диапазона от -100 до 100.

18. Вычислить сумму и число элементов матрицы B(8x8), находящихся под главной диагональю и на ней. Элементы матрицы - случайные целые числа из диапазона от -100 до 100.

19. Вычислить сумму и число положительных элементов матрицы B(8x8), находящихся над главной диагональю. Элементы матрицы - случайные целые числа из диапазона от -100 до 100.

20. Вычислить значения функции:

$$y_i = \begin{cases} \sqrt{ax_i^2}, x_i < 10 \\ \sqrt{bx_i^2}, 2,10 \le x < 20; a = 3,5; b = 2,6; c = 4,1; x_{i+1} = x_i + 3; x_1 = 1 \\ \sqrt{cx_i^2}, x \ge 20, i = 1,2,\ldots,50 \end{cases}$$

Библиографический список

1 Боровиков, В.П. Прогнозирование в системе STATISTICA в среде Windows: Основы теории и интенсивная практика на компьютере: учебное пособие для вузов / В.П. Боровиков, Г.И. Ивченко. - М.: Финансы и статистика, 2000. - 384 с.

2 Боровиков, В.П. Программа STATISTICA для студентов и инженеров / В. П. Боровиков. - М.: КомпьютерПресс, 2001. - 304 с.

3 Вуколов, Э.А. Основы статистического анализа. Практикум по статистическим методам и исследованию операций с использованием пакетов Statistica и Excel / Вуколов Э.А. - М.: Высшее образование, 2010. - 464 с.

4 Гультяев, А. Визуальное программирование в среде MATLAB : учеб. курс / А. Гультяев. - Спб.:Питер, 2001. – 432 с.

5 Дьяконов, В. MATLAB:учеб. курс / В. Дьяконов. - Спб.:Питер, 2012. – 738 с.

6 Лавренов, С.М. Excel: Сборник примеров и задач / С.М. Лавренов. - М.: Финансы и статистика, 2001. - 336 с.

7 Мартынов, Н.Н. MATLAB 5x. Вычисления, визуализация, программирование / Н.Н. Мартынов, А.П. Иванов. - М. : КУДИЦ-ОБРАЗ, 2000. – 336 с.

8 Микшина, В.С. Лабораторный практикум по информатике: учебное пособие для вузов / В.С. Микшина, Г.А. Еремеева, Н.Б. Назина и др.; Под ред. В.А. Острейковского. - М.: Высш.шк., 2003. - 376 с.

9 Халафян, А.А. Statistica 6. Статистический анализ данных / А.А. Халафян. - М.: Бином-Пресс, 2010. - 528 с.

10 Халафян, А.А. Statistica 6. Математическая статистика с элементами теории вероятностей / А.А. Халафян. - М.: Бином, 2010. - 496 с.

7 семестр

Лабораторная работа №1
Интерполяция функций

Цель работы: изучение методов решения задачи интерполяции; приобретение навыков программирования методов интерполяции; приобретение навыков использования стандартных средств системы MATLAB для проведения интерполирования.

Теоретические сведения

Постановка задачи интерполяции

Пусть на отрезке $a \le x \le b$ задана сетка $\varpi = \{x_0 = a < x_1 < x_2 < ... < x_n = b\}$ и в ее узлах x_i заданы значения функции y(x), равные

$$y(x_0) = y_0,..., y(x_i) = y_i,..., y(x_n) = y_n.$$

Требуется построить интерполянту – функцию f(x,c), совпадающую с функцией y(x) в узлах сетки

$$f(x_i, c) = y_i, i = 0,1,..., n, \qquad (1.1)$$

где $c = \{c_0,..., c_n\}$ - некоторые неизвестные параметры. Основная цель решения этой задачи состоит в том, чтобы иметь возможность вычисления значений функции y(x) для значений x, не содержащихся в таблице данных. Основным вопросом интерполяции является выбор интерполянты f(x,c) и оценка погрешности интерполяции, т.е. величины y(x)-f(x,c). Математически он заключается в определении неизвестных параметров c при выборе определенного вида функциональной зависимости.

Рассмотрим линейную зависимость функции f(x,c) от параметров c, т.е. будем считать, что она представима в виде обобщенного многочлена

$$f(x, c) = \sum_{k=0}^{n} c_k \phi_k(x).$$

Для определения неизвестных коэффициентов c_k необходимо решить систему линейных алгебраических уравнений:

$$\sum_{k=0}^{n} c_k \phi_k(x_i) = y_i, \ i = 0,1,...,n.$$

Данная задача имеет единственное решение только в том случае, когда определитель системы уравнений $\det[\phi_k(x_i)]$ отличен от нуля. Система функций, удовлетворяющая такому требованию, называется чебышевской. В качестве $\phi_k(x)$ чаще всего выбирают: степенные $\phi_k(x) = x^k$ (тогда $f(x) = P_n(x)$ - полином степени n), тригонометрические $\{\phi_k(x)\} = \{\cos kx, \sin kx\}$ (в этом случае f(x) – тригонометрический полином), экспоненциальные и другие системы линейно-независимых функций.

Любая непрерывная на отрезке [a, b] функция f(x) может быть хорошо приближена некоторым полиномом $P_n(x)$:

$$P_n(x) = \sum_{k=0}^{n} c_k x^k, \qquad (1.2)$$

где $c = \{c_0,..., c_n\}$ - неопределенные коэффициенты. Согласно основному условию интерполяции (2.1) для нахождения неизвестных коэффициентов c имеем систему линейных алгебраических уравнений:

$$c_0 + c_1 x_0 + c_2 x_0^2 + \ldots + c_n x_0^n = y_0,$$
$$c_0 + c_1 x_1 + c_2 x_1^2 + \ldots + c_n x_1^n = y_1,$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$c_0 + c_1 x_n + c_2 x_n^2 + \ldots + c_n x_n^n = y_n.$$

Данная система имеет единственное решение, так как ее определителем является отличный от нуля определитель Вандермонда. Отсюда следует, что интерполяционный полином (1.2) существует и единственен.

Интерполяционный многочлен Лагранжа

Будем строить многочлен n-й степени, который исторически обозначается $L_n(x)$ и называется многочленом Лагранжа, в виде линейной комбинации специальных (базисных) многочленов n-й степени $l_i(x)$ при i=0,1,…,n:

$$L_n(x) = \sum_{i=0}^{n} c_i l_i(x). \qquad (1.3)$$

Для того чтобы такой многочлен был интерполяционным для функции y(x), потребуем выполнения условий интерполяции $L_n(x) = y(x_i) = y_i$. Эти условия будут выполнены, если $c_i = y_i$, а базисные многочлены $l_i(x)$ удовлетворяют условиям

$$l_i(x_j) = \begin{cases} 0, \text{ если } j \neq i \\ 1, \text{ если } j = i \end{cases} \forall i, j \in \{0,1,\ldots,n\}. \qquad (1.4)$$

Положим

$$l_i(x) = \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}, \; i = 0,1,2,\ldots,n. \qquad (1.5)$$

Нетрудно проверить, что многочлены $l_i(x)$ удовлетворяют условиям (1.4). подставляя (1.5) в (1.3), получаем

$$L_n(x) = \sum_{i=0}^{n} \frac{(x - x_0)...(x - x_{i-1})(x - x_{i+1})...(x - x_n)}{(x_i - x_0)...(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_n)} y_i \qquad (1.6)$$

или в более компактной форме

$$L_n(x) = \sum_{i=0}^{n} y_i \prod_{j \neq 0}^{n} \frac{(x - x_j)}{(x_i - x_j)}.$$

Погрешность интерполяции степенными функциями

Рассмотрим в качестве погрешности величину остаточного члена интерполяции в произвольной точке $x \in [a, b]$:

$$R_n(x) = y(x) - L_n(x).$$

Оценку величины $R_n(x)$ будем проводить в предложении, что функция f(x) имеет (n+1) непрерывную производную на отрезке [a, b].

Введем многочлен (n+1)-й степени $\Pi_{n+1}(x)$, определенный через узлы $x_0, x_1, ..., x_n$:

$$\Pi_{n+1}(x) = \prod_{i=0}^{n} (x - x_i) = (x - x_0)(x - x_1)...(x - x_n).$$

Тогда абсолютную погрешность интерполяционной формулы Лагранжа (1.6) в произвольной точке $x' \in [a, b]$ можно оценить с помощью неравенства

$$|R_n(x)| \leq \frac{M}{(n+1)!} |\Pi_{n+1}(x)|, \qquad (1.7)$$

где $M = \max_{x \in [a,b]} |f^{(n+1)}(x)|$.

Рассмотрим как ведет себя погрешность интерполяции при увеличении числа узлов. Выражение для погрешности (1.7) состоит из трех различных частей; факториал и произведение разностей с увеличением n уменьшают ошибку, а порядок производной при этом растет. Для многих функций величина M_{n+1} увеличивается быстрее, чем (n+1)!. В результате полиномиальные интерполянты редко сходятся к обычной непрерывной функции при $n \to \infty$. Наблюдаемый на практике эффект проявляется в том, что интерполяционный полином высокой степени может вести себя «плохо» в точках, отличных от узлов интерполяции (x_i, y_i), $i = \overline{0,n}$. Поэтому на практике часто используют интерполянты степени не выше 5-6.

Примером может служить функция Рунге вида $r(x) = 1/(1 + 25x^2)$. С увеличением порядка интерполирующего полинома при равномерном распределении узлов интерполяции на интервале [-1, 1] происходит ухудшение качества приближения на краях интервала. Это объясняется тем, что производные r(x), которые фигурируют в выражении для погрешности интерполяции, быстро растут с увеличением числа n. Таким образом, точность приближения зависит не только от числа узлов интерполяции (т.е. порядка интерполирующего полинома), но и от их расположения на интервале [a, b].

Задача о наилучшем выборе узлов интерполирования была решена Чебышевым. Наилучшие узлы интерполирования выбираются равными корнями «полинома, наименее отклоняющегося от нуля» на отрезке интерполирования. Полином, наименее отклоняющийся от нуля на отрезке [-1, 1], был найден Чебышевым и назван его именем. Полином Чебышева определяется следующим выражением:

$$T_n(t) = \cos(n \arccos t), \qquad (1.8)$$

где n=0, 1, 2, … Эта тригонометрическая функция является многочленом при любом n. При n=0 и n=1 непосредственно из (1.8)

получаем $T_0(t) = 1, T_1(t) = t.$ Далее, обозначая $\alpha = \cos t$, имеем $T_1(t) = \cos\alpha$, $T_n(t) = \cos n\alpha$. Так как по правилу сложения косинусов $\cos(n+1)\alpha + \cos(n-1)\alpha = 2\cos\alpha\cos n\alpha$, справедливо

$$T_{n+1}(t) + T_{n-1}(t) = 2T_1(t)T_n(t). \qquad (1.9)$$

Из (1.9) следует, что последовательность функций $T_n(t)$, определяемая рекуррентно, представляет собой многочлен степени n: $T_2(t) = 2t^2 - 1, T_3(t) = 4t^3 - 3t, T_4(t) = 8t^4 - 8t^2 + 1$ и т.д.

Из всех многочленов степени n со старшим коэффициентом 1нормированный многочлен Чебышева $T_n'(t) = 2^{-n+1}T_n(t)$ наименее уклоняется от нуля на отрезке [-1, 1]. Последнее фактически означает, что среди всех многочленов степени n вида

$$P_n(t) = t^n + a_1t^{n-1} + a_2t^{n-2} + ... + a_n$$

именно нормированный многочлен $T_n'(t)$ минимизирует максимальное расстояние от графика многочлена до оси абсцисс при $t \in [-1,1]$. Таким образом, максимальная погрешность интерполирования будет минимальной, если в качестве многочлена $\Pi_{n+1}(t)$ взять нормированный многочлен Чебышева $T_n(t)$, так как именно он наименее уклоняется от нуля на отрезке [-1, 1]. В этом случае точки $t_0, t_1, ..., t_n$ будут корнями многочлена Чебышева $T_{n+1}(t)$ или $T_{n+1}'(t)$, которые имеют вид

$$t_k = \cos\frac{2k+1}{2n}\pi, \text{ где } k = 0, 1, ..., n-1.$$

Обычно эти узлы называются чебышевскими узлами интерполяции.

На интервале [a, b] эти корни можно представить по формуле Чебышева

$$x_{i+1} = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{(2i+1)\pi}{2n+2}\right), i = \overline{0, n.}$$

Интерполяционная функция с использованием именно этих точек в качестве узлов дает наилучшее приближение к интерполируемой функции.

Средства MATLAB для проведения интерполяции

Для интерполирования функций в MATLAB можно использовать операторы polyfit и polyval.

Оператор polyfit(x,y,n) находит массив коэффициентов a длиной (n+1) полинома степени n по массивам длиной m узлов x и значений функции y в узлах $x, m \geq n+1$. Этот полином аппроксимирует функцию y(x) по методу наименьших квадратов. Предполагается, что полином задается в виде

$$y(x) = a_1 x^n + a_2 x^{n-1} + \ldots + a_n x + 1.$$

Если m=n+1, то программа возвращает коэффициенты интерполяционного полинома.

Оператор polyval(a,x) возвращает значение полинома в точке x, коэффициенты которого определены в векторе a.

Кроме этого, существуют операторы, позволяющие проводить интерполяцию одно- и многомерных данных:

- interpft(y,n,dim) - аппроксимация периодической функции на основе быстрого преобразования Фурье (y - одномерный массив значений функции; n - число узлов в массиве значений);
- spline(x,y,z) - интерполяция y=y(x) кубическим сплайном и вывод

соответствующих значений в точках z. Для получения большей информации используется конструкция pp=spline(x,y), здесь командой v=ppval(pp,z) можно найти значения в точках z, а командой [xs,Coef,m,L]=unmkpp(pp) получить данные о векторе разбиений аргумента xs, коэффициентах Coef, m=length(xs), L=length(Coef)/m;

- interp1(x,y,z), interp1(x,y,z,'method') - одномерная табличная интерполяция (если y - двумерный массив, интерполяция ведется по каждому столбцу; значения z должны входить в диапазон значений x). Можно указать метод интерполяции -

кусочно-линейной (linear по умолчанию), ступенчатой (nearest), кубической (cubic), кубическими сплайнами (spline). Функция interp1q(x,y,z) реализует быструю линейную интерполяцию на неравномерной сетке;

- interp2(x1,x2,y,z1,z2), interp1(x1,x2,y,z1,z2,'method') - двумерная табличная интерполяция y=y(x1,x2), аргументы должны меняться монотонно и быть заданы в формате функции meshgrid;
- interp3(x1,x2,x3,y,z1,z2,z3), interp3(...,'method') - трехмерная табличная интерполяция y=y(x1,x2,x3);
- interpn(x1 ,x2,...,y,z1,z2,...), interp3(...,'method') - многомерная табличная интерполяция y=y(x1,x2,...);
- griddata(x1,x2,y,z1,z2), griddata(x1,x2,y,z1,z2,'method') - двумерная табличная интерполяция на неравномерной сетке.

Для реализации программы полезными являются следующие операторы:

- sum(x) и prod(x) - суммирование и произведение элементов вектора (для двумерного массива выполняется поиск сумм и произведений по столбцам);
- diff(x), diff(x,n), diff(x,n,dim) - вычисление конечных разностей (первых, n-го порядка или по указанному измерению); если x - массив, берутся разности между столбцами.

Порядок выполнения работы

Написать соответствующие m-файлы сценарии и m-файлы функции для реализации следующих задач.

1. Провести интерполяцию функции Рунге на отрезке [-1, 1] по формуле Лагранжа для n=11 при равномерном распределении узлов интерполяции.
2. Провести интерполяцию функции Рунге на отрезке [-1, 1] по формуле Лагранжа для n=11 для чебышевских узлов.
3. Построить графики функции Рунге и ее интерполянт не менее чем в 100 узлах. Сравнить результаты.
4. Выбрать функцию согласно номеру варианта и провести ее интерполяцию по формуле Лагранжа при равномерном распределении узлов на заданном интервале для n=11 и n=6.

5. Провести интерполяцию по тем же узлам, используя стандартные функции MATLAB.

6. Построить графики исходной функции и интерполянт не менее чем в 100 узлах. Сравнить результаты.

7. Вычислить и построить графики функций ошибок интерполяции.

Содержание отчета

1. Цель работы, задание;
2. Описание метода решения, краткие сведения из теории (формулы, алгоритм и т.п.);
3. Программу (распечатку), ее описание;
4. Сравнение результатов расчета;
5. Краткие выводы.

Задание

Номер варианта	y(x)	a	b
1	$\sqrt{(x-2)^3(5-x)}$	2	5
2	$(1+x)(x^2+3)^{-1}$	-9	9
3	$10e^{-x}(x^3-2x+1)$	-2	2
4	$(1+x^2)\sin 2x$	-1	5
5	$\exp(-0{,}5x\cos x)$	-5	3
6	$\exp(2{,}5\sin x)\cos x$	1	3
7	$x^2\sin(2x-3)$	0	4
8	$x(x^2+3)^{-1}$	-5	5
9	$(1+x^2)(1+x^3)^{-1}$	0	4
10	$(x-3)\cos^2 x$	-3	3
11	$(1-x)\cos x$	0	4
12	$\sqrt{x}-\cos(1{,}5x)$	0	8
13	$\sin 2x-0{,}2x^2$	0	6
14	$(x^2+1)^{-1}\sin(x+1)$	-3	2
15	$0{,}5x^2+8x^{-1}+8$	-4	-1

Контрольные вопросы

1. Сформулируйте постановку задачи интерполяции.
2. Сформулируйте постановку задачи полиномиальной интерполяции со степенным базисом.
3. Выведите интерполяционную формулу Лагранжа. Приведите примеры для интерполяции по 2 и 3 точкам.
4. Приведите оценку погрешности интерполяционной формулы Лагранжа.
5. Дайте определение полиномов Чебышева. Выведите рекуррентное соотношение для их вычисления.
6. Как осуществляется интерполяция по чебышевским узлам?
7. Дайте определение интерполяционного полинома Ньютона. Укажите условия практического применения многочленов Ньютона и Лагранжа.
8. Дайте определение разделенных и конечных разностей. Укажите их свойства.
9. Постройте таблицу разделенных разностей для пяти точек.
10. Поставьте задачу кусочно-полиномиальной интерполяции.

Лабораторная работа №2
Аппроксимация данных методом наименьших квадратов

Цель работы: изучение метода наименьших квадратов для аппроксимации данных; приобретение навыков программирования метода наименьших квадратов для аппроксимации экспериментальных данных; приобретение навыков использования стандартных средств системы MATLAB для задачи аппроксимации.

Теоретические сведения

Пусть имеются результаты измерений двух величин x и y и предполагается, что они связаны линейной зависимостью

$$y = b_0 + b_1 x + \varepsilon. \qquad (2.1)$$

При этом ошибка измерения является некоторой случайной величиной, а ее среднее значение равно нулю. Так как результаты обычно не ложатся на прямую, то необходимо построить наилучшую линейную функцию, проходящую наиболее близко к каждому результату измерения, но возможно не совпадающую с ними. Это типичная задача обработки результатов эксперимента.

Для нахождения неизвестных коэффициентов модели (2.1) используют метод наименьших квадратов, а именно минимизируют функцию двух переменных вида

$$Q(b_0, b_1) = \sum_{i=1}^{n} [y_i - (b_0 + b_1 x_i)]^2.$$

Запишем необходимое условие минимума дифференцируемой функции:

$$\frac{\partial Q}{\partial b_0} = 0, \ \frac{\partial Q}{\partial b_1} = 0. \qquad (2.2)$$

В результате решения системы (2.2) из двух линейных уравнений получим

$$b_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}, \; b_0 = \bar{y} - b_1\bar{x}, \qquad (2.3)$$

где средние значения x и y определяются по формулам

$$\bar{x} = \frac{1}{n}\sum x_i, \; \bar{y} = \frac{1}{n}\sum y_i.$$

Формулы (2.3) позволяют построить график прямой, называемой прямой регрессии e на x, которая дает приближенное представление о зависимости между переменными y и x. В дополнение к ней обычно строят и регрессию x на y, модель которой аналогична (2.1), только в качестве независимой переменной используют y. Таким образом, получают две пересекающиеся в точке $(\bar{x}, \bar{y})$ прямые, при этом большинство результатов измерений лежит именно между ними. Во многих случаях удается линеаризовать исходную зависимость и рассматривать ее относительно некоторых новых переменных, которые образуют линейную связь. Например, для функции $y = ax^b + c$ берут новые переменные Y=lg(y-c) X=lgx и получают зависимость вида Y=bX+lga.

В общем случае, если регрессия y на x отличается от линейной, рассматривают линейную по параметрам регрессионную модель вида

$$y = b_0 + b_1 a_1(x) + \ldots + b_{k-1} a_{k-1}(x), \qquad (2.4)$$

где $a_1(x), \ldots, a_{k-1}(x)$ - известные функции; $b_0, b_1, \ldots, b_{k-1}$ - неизвестные параметры.

Пусть имеется n наблюдений (x_i, y_i), которые являются результатом реализации случайного вектора (X, Y). Подставляя в (2.4) выборочные точки x_i, получаем систему:

$$y_i = b_0 + b_1 a_1(x_i) + \ldots + b_{k-1} a_{k-1}(x_i) + \varepsilon_i, \; i = 1,2,\ldots,n. \qquad (2.5)$$

Здесь ε_i - случайные независимые друг относительно друга и распределенные по нормальному закону ошибки наблюдений. Необходимо отметить, что система уравнений (2.5) относительно неизвестных b_j является типичной пере- или недоопределенной задачей (в зависимости от соотношения n и k) и должна решаться методом наименьших квадратов.

По методу наименьших квадратов в качестве оценок b_j принимают значения $\tilde{b}_j$, дающие минимум функции

$$Q(b_0, b_1,\ldots, b_{k-1}) = \sum_{i=1}^{n} (y_i - (b_0 + b_1 a_1(x_i) + \ldots + b_{k-1} a_{k-1}(x_i)))^2.$$

Из необходимых условий минимума следует, что оценки $\tilde{b}_j$ являются решениями алгебраической системы k уравнений с k неизвестными. Данную систему уравнений часто называют нормальной системой. В матричных обозначениях эту систему уравнений можно записать в виде

$$(A^T A)\tilde{b} = A^T y, \qquad (2.6)$$

где $y^T = y_1, y_2,\ldots, y_n)$ - n-вектор наблюдений; $\tilde{b}^T = (b_0, b_1,\ldots, b_{k-1})$ - k-вектор оценок параметров:

$$A = \begin{bmatrix} 1 & a_1(x_1) & . & a_{k-1}(x_1) \\ 1 & a_1(x_2) & . & a_{k-1}(x_2) \\ \dot{1} & \dot{a}_1(x_n) & . & \dot{a}_{k-1}(x_n) \end{bmatrix} \text{- регрессион ная } n \times k \text{ - матрица}$$

При условии, что матрица $(A^T A)$ невыраженная, решение (2.6) можно записать в виде

$$\tilde{b} = (A^T A)^{-1} A^T y.$$

Часто в качестве функций $a_k(x)$ принимают степенные функции, т.е. $a_k(x) = x^k$. В этом случае регрессионная матрица имеет вид

$$A = \begin{bmatrix} 1 & x_1 & . & x_1^{k-1} \\ 1 & x_2 & . & x_2^{k-1} \\ . & . & . & . \\ 1 & x_n & . & x_3^{k-1} \end{bmatrix}.$$

Уравнение для нахождения неизвестного вектора параметров b будет иметь вид

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & . & \sum x_i^{k-1} \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & . & \sum x_i^k \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & . & \sum x_i^{k+1} \\ \sum . x_i^{k-1} & \sum . x_i^k & \sum . x_i^{k+1} & . & \sum . x_i^{2k-2} \end{bmatrix} \cdot \begin{bmatrix} \tilde{b}_0 \\ \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_{k-1} \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \\ \sum . y_i x_i^{k-1} \end{bmatrix} \quad (2.7)$$

При больших значениях k система (2.7) плохо обусловлена и для ее решения используются специальные методы. В обычной практике статистической обработке данных k невелико ($k \le 5$) и тогда эту систему можно решить либо методом Гаусса, либо подходящим итерационным методом.

Средства MATLAB для аппроксимации функций

Для предварительной статистической обработки данных используются операторы:

- max(A), min(A) - поиск экстремальных элементов по столбцам массива A;
- max(A,B), min(A,B) - формирование массива с элементами, равными экстремальным из соответствующих элементов массивов;

- max(A,[],dim), min(A,[],dim) - вектор экстремальных элементов по измерению dim;
- [C,I]=max(...), [C,I]=min(...) - дополнительно выводится строка индексов экстремальных элементов;
- median(x), median(x,dim) - медианы массива;
- mean(x), mean(x,dim) - средние значения;
- std(x), std(x,flag), std(x,flag,dim) - стандартное отклонение (flag=0 - несмещенная оценка для s ; flag=1 - смещенная оценка);
- cov(x,y), cov(x,y,flag) - ковариация для массивов x и y (каждый столбец - переменная, строка - наблюдение);
- cov(x), cov(x,flag) - ковариация для столбцов массива x;
- corrcoef(x), corrcoef(x,y) - коэффициенты корреляции.

Для построения полиномиальной регрессионной кривой используется оператор polyfit(x,y,n), возвращающий вектор коэффициентов полинома p(x) степени n, который с наименьшей среднеквадратичной погрешностью аппроксимирует функцию y(x). Результатом является вектор-строка длиной (n+1), содержащая коэффициенты полинома в порядке уменьшения степеней x. Если length(x)=n+1, то реализуется обычная полиномиальная аппроксимация, при которой график полинома точно проходит через узловые точки с координатами (x, y), хранящиеся в векторах x и y. В противном случае точного совпадения графика с узловыми точками не наблюдается. Полином строится в виде

$$p(1) * x^{n} + p(2) * x^{(n-1)} + \ldots + p(n) * x + p(n+1).$$

Порядок выполнения работы

1. Написать m-файлы для реализации метода наименьших квадратов для построения линейной регрессии по x и y. Взять 10 первых точек из своего варианта. Вычислить медиану, средние значения, смещенную и несмещенную оценку среднеквадратичного отклонения, коэффициент корреляции.
2. Написать m-файлы функции для реализации метода наименьших квадратов для построения нелинейной

полиномиальной регрессии со степенным базисом (n=3, 5, 7) и построить аппроксимирующие кривые.

3. Построить график и отобразить на нем следующие данные: исходные данные в виде точек и графики аппроксимирующих функций.

4. Использовать стандартные средства MATLAB для построения соответствующих кривых. Сравнить результаты.

5. Построить графики ошибок аппроксимации для всех n.

Содержание отчета

1. Цель работы, задание;
2. Описание метода решения, краткие сведения из теории (формулы, алгоритм и т.п.);
3. Программу (распечатку), ее описание;
4. Сравнение результатов расчета;
5. Краткие выводы.

Задание

Вариант 1. x=[0:0.1:5]

2,2097	-6,8439	-1,1201	-5,6987	-4,8524	1,7439	1,0118	-0,5526	6,1079	2,3993
1,6262	8,4938	-4,8348	10,1190	7,7189	11,5615	-3,3621	9,6958	17,9124	1,1666
-2,0931	4,5551	9,6773	18,1837	19,0898	7,0177	9,5207	23,9392	16,3089	22,8691
29,6142	35,6152	30,7599	48,6692	50,3154	28,7633	47,1599	32,2506	56,5752	3,4583
57,5376	70,1709	77,7399	69,4833	91,2748	85,4722	100,9104	98,6184	104,6567	1258,1061
121,2690									

Вариант 2. x=[1:0.1:6]

0,9770	0,7950	0,6755	0,6424	0,5276	0,4223	0,4701	0,3811	0,36	0,2269
0,1635	0,1913	0,2440	0,1776	0,1848	0,2027	0,1306	0,1317	0,1842	0,1531
0,1250	0,0713	0,1601	0,1217	0,1106	0,0325	-0,0109	0,0017	0,0237	0,0494
0,0590	0,1345	0,0776	0,0551	0,0402	0,0998	0,0805	0,0174	0,1028	0,0804
0,0264	-0,0383	0,0896	0,0043	0,0742	0,0487	0,0620	-0,0322	-0,0132	0,1340
0,0458									

Вариант 3. x=[0:0.1:5]

-0,0627	0,2727	0,4647	0,5818	0,7122	0,8727	0,9211	0,9423	0,9318	0,9329
0,9794	1,1605	1,0324	1,1018	1,0523	1,2707	1,2655	1,3511	1,2770	1,2805
1,3692	1,1997	1,4977	1,4779	1,7262	1,6303	1,7824	1,6529	1,7918	1,7635
1,8133	1,7746	1,7036	1,6771	1,8053	1,9213	1,8812	1,9930	2,0343	2,1015
2,1716	2,0307	1,9889	2,0551	2,0983	2,1195	2,1950	1,9982	2,2107	2,3409
2,2633									

Вариант 4. x=[0.1:0.1:5.1]

-2,3660 -1,7701 -1,3201 -1,2720 -0,5055 -0,3275 -0,4319 -0,4051 -0,1362 0,0405
-0,2024 0,3066 0,1005 -0,0493 0,3263 0,6423 0,0442 0,7559 0,5858 0,5291
0,4964 0,8012 0,7039 1,2297 0,9044 1,1075 1,3314 0,8089 0,7722 1,0514
1,3509 0,6801 1,2743 1,0409 1,2800 1,0181 1,2439 1,4303 1,3457 1,4073
1,1276 1,2935 1,3850 1,6022 1,6745 1,3950 1,2535 1,7307 1,8445 1,2650
1,6089

Вариант 5. x=[-2.5:0.1:2.5]

0,4032 0,1809 -0,2587 -0,0399 -0,6520 0,1575 0,1644 0,3564 0,4596 -0,2643
0,8811 0,3974 1,5723 -0,0888 0,0314 -0,1035 0,9185 0,4832 0,4557 1,4323
1,4972 1,0005 0,0651 -0,2495 0,8214 1,8526 0,9669 1,0241 1,3991 1,4036
2,5676 2,5733 1,6042 2,3428 3,2754 2,8773 3,4023 2,9747 3,6903 3,8932
4,2285 5,4673 5,4242 6,1078 6,3413 6,4474 8,0035 9,0726 9,9586 11,3301
13,0481

Вариант 6. x=[-0.5:0.04:1.5]

1,6557 1,5859 1,9347 1,8121 1,9226 1,9367 1,9364 1,8522 1,8733 1,9431
1,8715 1,8908 2,1468 2,0052 1,8747 1,9859 1,8677 1,8331 1,9265 2,0296
1,9262 1,9555 1,7472 1,7900 1,7799 1,6532 1,6095 1,8002 1,6729 1,5712
1,6797 1,5855 1,4265 1,6228 1,4816 1,5650 1,4963 1,2806 1,1876 1,2788
1,2239 1,1453 1,2541 1,3299 1,1053 1,0565 1,1002 1,1773 1,0142 0,9708
1,0728

Вариант 7. x=[-0.5:0.04:1.5]

0,8381 0,6434 0,8021 0,5503 0,7475 0,8743 0,7827 0,9828 0,7537 0,5455
0,6848 0,9725 0,8272 0,8888 0,9585 0,5737 0,7556 0,7446 0,7009 0,8001
1,0066 0,8044 0,9543 0,7044 0,7818 0,6713 0,7297 0,7109 0,7870 0,7939
0,6826 0,7843 0,7412 0,6961 0,6057 0,6965 0,5632 0,3903 0,5605 0,4690
0,4037 0,5651 0,6047 0,2716 0,3140 0,1634 0,3118 -0,1071 0,2734 0,1377
0,1034

Вариант 8. x=[-2.5:0.1:2.5]

5,5555 5,2521 5,8400 4,5695 1,0690 3,5246 3,2401 2,9342 2,7411 2,4548
1,8499 1,9965 2,0240 1,8077 1,4790 1,5543 1,6709 1,6499 1,1657 0,8922
0,7358 0,8107 1,2648 0,7741 1,3061 1,0375 1,4110 0,6209 0,9102 0,8028
1,1657 1,0943 0,5789 1,7289 1,7427 1,0020 1,5585 1,9062 2,1220 2,4689
1,9622 2,2834 2,6925 2,9737 3,0287 3,4221 3,7058 4,8680 5,2288 5,1957
5,8044

Вариант 9. x=[-2.5:0.1:2.5]

-60,7872 -44,0478 -40,2793 -22,4243 -28,9959 -11,1719 -8,5942 -10,2967 -10,5551 -19,5120
-11,5548 -5,2881 -9,5449 -4,5954 -0,1912 -0,7586 -1,3469 -4,5631 -2,8370 -3,6565
-4,1916 9,9903 -5,5046 0,1045 -2,0025 -5,8137 -2,4241 0,4929 -9,0445 -3,7295
3,1438 -1,2373 2,2046 -12,3192 6,0708 -5,6375 6,6117 7,4719 7,5173 11,0604
5,6081 11,2105 9,446 9,4092 25,8665 28,3543 18,1601 32,7853 41,4828 46,8581
53,9812

Вариант 10. x=[-2.5:0.1:2.5]

-16,1665 -15,0660 -11,0476 -9,1973 -11,0186 -7,0022 -4,1416 -6,0624 -3,7241 -3,8446
-2,7844 -2,7243 -3,3814 -2,7923 -1,2677 -0,3478 -3,3987 -1,3731 0,6179 -1,2917
-0,6539 -0,1824 -0,8659 0,3861 0,0828 1,6138 -0,0362 0,3524 2,9339 2,9847
-0,1690 1,4802 1,0469 2,2307 1,4094 2,5439 0,3566 1,4424 4,7233 1,5527
3,6870 3,8141 3,7949 5,1861 5,3505 8,3778 9,8362 9,7080 11,5358 14,2881
15,3980

Вариант 11. x=[-2:0.1:3]

-0,7095 -1,5838 -1,6252 -2,2546 -2,4277 -1,8296 -1,1479 -1,3409 -1,5277 -0,1438
-0,8892 -0,4225 -0,3261 -0,3788 0,0011 -0,2830 -0,9448 -0,5947 0,0700 -0,6535
-0,1736 -0,4455 -0,5358 -0,4303 -0,0880 0,2379 0,1391 1,7861 0,2685 -0,3419
0,5380 0,7159 1,0192 0,3202 1,9274 1,2039 1,7263 1,6128 1,2685 2,5146
1,1874 0,4853 0,9640 0,1491 0,3526 1,1647 1,4467 0,3341 0,8319 0,6880
0,1797

Вариант 12. x=[-2:0.1:3]

-1,6072 -1,4757 -0,7322 -0,7821 -0,3665 0,3615 0,5739 1,5260 0,8965 2,0787
1,9542 0,7020 0,9112 1,433 1,6451 1,7587 2,1061 1,4802 1,4615 1,6351
1,6585 1,7164 1,9547 1,6961 0,6543 1,5326 1,4709 1,2881 1,4408 0,9872
0,4062 0,4934 -0,5482 0,5004 0,2399 -0,1578 -0,2936 -0,5693 -0,2558 -0,1781
-0,5483 -0,6949 -0,5803 -0,6985 -1,0657 -02113 -0,5887 -0,2807 -1,0114 -1,2703
-0,5019

Вариант 13. x=[0:0.8:4]

0,0059 -0,0274 0,1511 -0,0370 0,4939 0,2898 0,4232 0,8032 0,5692 0,7222
0,7837 1,1532 0,9884 0,8656 0,8900 1,3163 1,3501 1,3000 0,8693 0,9910
0,9580 0,9890 0,8174 0,7106 0,7361 0,9040 1,0387 0,6846 0,5683 0,6640
0,4547 0,7545 0,6140 0,2515 0,3056 0,5706 0,4154 0,4001 0,0869 0,3531
0,4111 0,2174 0,1631 0,2805 0,1600 0,3068 0,3666 0,5361 0,3825 0,3006
0,6375

Вариант 14. x=[0:0.8:4]

1,8205 0,7979 0,9409 0,8066 2,0887 20372 2,1345 1,8844 2,336 2,1583
1,7639 2,1481 2,2576 2,1162 2,1942 2,1289 1,6940 1,3912 1,9982 1,6866
1,5450 1,3948 1,1302 1,3512 1,0199 1,0516 1,1084 1,2309 1,1400 0,8365
1,3023 1,0661 1,3884 0,9011 1,0076 1,0544 1,2169 1,1322 1,2020 1,0408
1,3792 1,6504 1,6726 1,7785 1,8982 2,0655 2,0827 2,1873 2,5384 2,5431
2,8220

Вариант 15. x=[2:0.06:5]

2,0000 2,0600 2,1200 2,1800 2,2400 2,3000 2,3600 2,4200 2,4800 2,5400
2,6000 2,6600 2,7200 2,7800 2,8400 2,9000 2,9600 3,0200 3,0800 3,1400
3,2000 3,2600 3,3200 3,3800 3,4400 3,5000 3,5600 3,6200 3,6800 3,7400
3,8000 3,8600 3,9200 3,9800 4,0400 4,1000 4,1600 4,2200 4,2800 4,3400
4,4000 4,4600 4,5200 4,5800 4,6400 4,7000 4,7600 4,8200 4,8800 4,9400
5,5000

Контрольные вопросы

1. Какие бывают виды аппроксимации экспериментальных данных?
2. В чем разница между аппроксимацией по методу наименьших квадратов и интерполяцией?
3. Что такое линейная регрессия? Как можно линеаризовать данные? Приведите примеры.
4. Что такое линейная по параметрам регрессия? Какие требования предъявляются к базисным функциям?
5. Построить алгоритм вычисления линейной по параметрам регрессионной модели со степенным базисом.
6. Что такое регрессионная матрица?

Лабораторная работа №3
Регрессионный анализ

Цель работы: овладеть способами реализации регрессии различного вида, наиболее часто применяемый в задачах обработки данных.

Теоретические сведения

Под регрессионным анализом (или просто регрессией) обычно подразумевают нахождение некоторой формальной аналитической зависимости, которая приближенно (по критерию минимума среднеквадратической ошибки) аппроксимирует исходную зависимость. Последняя чаще всего бывает представлена некоторым набором точек (например, полученных в результате эксперимента).

Реализация линейной регрессии общего вида

При этом виде регрессии заданная совокупность точек приближается к функции вида

$$F(x, K_1, K_2, ..., K_n) = K_1 * F_1(x) + K_2 F_2(x) + ... + K_n F_n(x).$$

Таким образом, функция регрессии является линейной комбинацией функций $F_1(x), F_2(x), ..., F_n(x)$, причем сами эти функции могут быть нелинейными, что резко расширяет возможности такой аппроксимации и распространяет ее на многие нелинейные функции. Для реализации линейной регрессии общего вида (ЛРОВ) используется функция Linfit (VX, VY, F), возвращающая вектор K коэффициентов ЛРОВ, при котором среднеквадратичная погрешность приближения «облака» исходных точек, координаты которых хранятся в векторах VX и VY, оказывается минимальной. Вектор F должен содержать функции $F_1(x), F_2(x), ..., F_n(x)$, записанные в символическом виде (рисунок 3.1).

$$VX := \begin{pmatrix} 1.1 \\ 2 \\ 3 \\ 4 \\ 4.8 \end{pmatrix} \quad VY := \begin{pmatrix} 25 \\ 12 \\ 9.4 \\ 16.2 \\ 26 \end{pmatrix} \quad F(x) := \begin{pmatrix} \frac{1}{x} \\ x^2 \\ \exp(x) \end{pmatrix}$$

$$K := \mathrm{linfit}(VX, VY, F) \quad g(t) := F(t) \cdot K$$

$$i := 0...4 \quad r := 1, 1.25...5$$

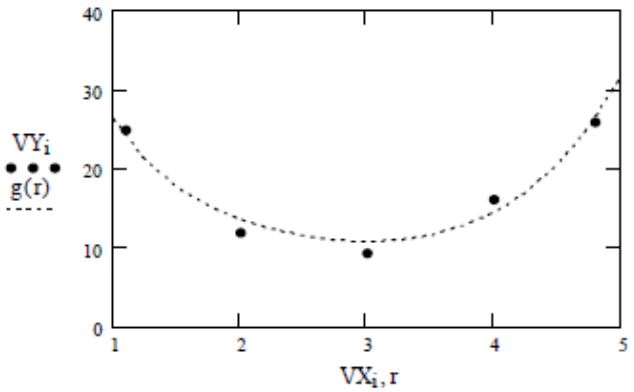


Рисунок 3.1 – Пример проведения линейной регрессии общего вида

Коэффициент функции регрессии

$$K = \begin{pmatrix} 26.064 \\ -0.284 \\ 0.228 \end{pmatrix}$$

Реализация одномерной полиноминальной регрессии

Для обеспечения полиноминальной регрессии при произвольной степени полинома вводится функция regress (VX, VY, n), которая возвращает вектор VS, запрашиваемый функцией interp (VS, VX, VY, x) и содержащий коэффициенты многочлена n-й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в векторах VX и VY. Для вычисления коэффициентов полинома регрессии в данном случае используется функция submatrix (рисунок 3.2).

$$\text{data} := \begin{pmatrix} 1 & 0.8 \\ 2 & 3.5 \\ 3 & 8 \\ 4 & 15 \\ 5 & 19 \\ 6 & 15 \end{pmatrix} \quad \text{Степень полинома } k := 3$$

Создание векторов X и Y исходных данных

$$X := \text{data}^{\langle 0 \rangle} \quad Y := \text{data}^{\langle 1 \rangle} \quad n := \text{rows}(\text{data})$$

$$z := \text{regress}(X, Y, k) \quad \text{fit}(x) := \text{interp}(z, X, Y, x)$$

$$i := 0..n-1 \quad \text{coeffs} := \text{submatrix}(z, 3, \text{lengh}(z) - 1, 0, 0)$$

$$j := 0...49 \qquad (\text{coeffs})^{T} = (7.133 \ -11.231 \ 5.734 \ -0.606)$$

$$tx_j := \min(X) + j \cdot \frac{\max(X) - \min(X)}{50}$$

$$R_2 := \frac{\sum \overrightarrow{(\text{fit}(X) - \text{mean}(Y))^2}}{\sum \overrightarrow{(Y - \text{mean}(Y))^2}} = 0.995$$

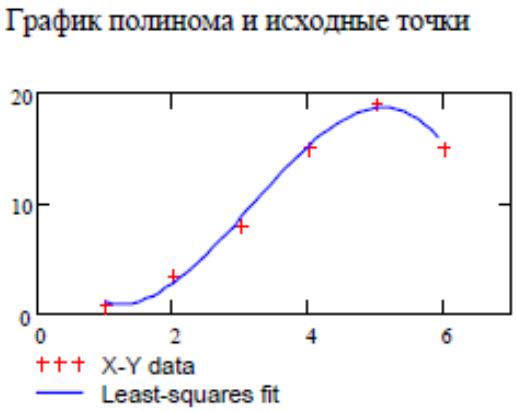


Рисунок 3.2 – Полиноминальная регрессия

На практике не рекомендуется делать степень аппроксимирующего полинома выше 4-6, поскольку погрешности реализации регрессии сильно возрастают. Функция regress создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек.

Регрессия отрезками полинома второй степени

Иногда, например, для фильтрации сигналов, полезна другая функция полиноминальной регрессии, дающая локальные приближения отрезками полиномов второй степени loes (VX, VY, span), которая возвращает вектор VS, используемый функцией interp (VS, VX, VY, x) для наилучшего приближения данных векторов VX и VY отрезками полиномов второй степени. Аргумент span>0 указывает размер локальной области приближаемых данных (рекомендуемое начальное значение -0,75). Чем больше span, тем сильнее сказывается сглаживание данных. При больших значениях span эта функция приближается к функции regress (VX, VY, 2). Пример приближения сложной функции со случайным разбросом ее значений с помощью совокупности отрезков полиномов второй степени для двух значений параметра span показан на рисунке 3.3.

$$i \coloneqq 1...199 \quad VX_i \coloneqq i$$

$$VY_i \coloneqq a\tan\left(\frac{i}{5}\right)\cdot\left(1-\exp\left(\frac{-i}{10}\right)+0.5\cdot md(1)\right)$$

$$span_1 \coloneqq 0.05 \quad span_2 \coloneqq 0.5$$

$$VS_1 \coloneqq loess(VX, VY, span_1) \quad VS_2 \coloneqq loess(VX, VY, span_2)$$

$$F_1(x) \coloneqq interp(VS_1, VX, VY, x)$$

$$F_{21}(x) \coloneqq interp(VS_2, VX, VY, x)$$

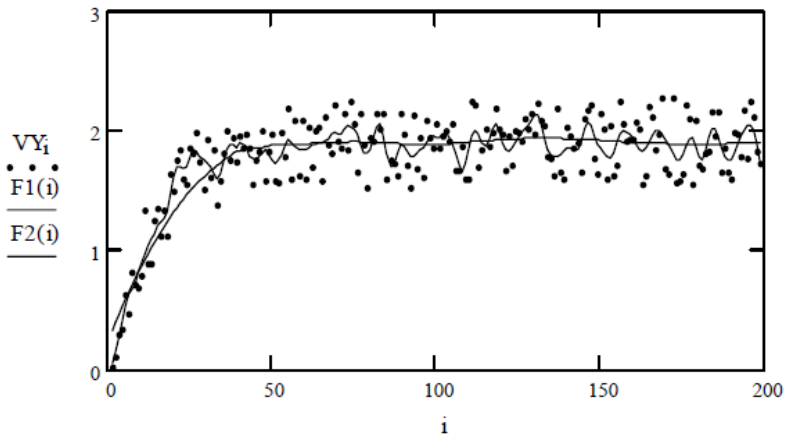


Рисунок 3.3 – Пример регрессии отрезками полиномов второй степени

Из рисунка 3.3 видно, что при значении span=0.05 отслеживаются характерные случайные колебания значений функции, тогда как уже при span=0.5 кривая регрессии становится практически гладкой. Недостатком данного вида регрессии является отсутствие простого описания аппроксимирующей функции в виде отрезков полиномов.

Проведение нелинейной регрессии общего вида

Под нелинейной регрессией общего вида (НРОВ) подразумевается нахождение вектора K параметров произвольной функции $F(x, K_1, K_2, ..., K_n)$, при котором обеспечивается минимальная среднеквадратичная погрешность приближения «облака» исходных точек.

Для проведения НРОВ используются функции genfit (VX, VY, VS, F), которая возвращает вектор K параметров функции F, дающий минимальную квадратичную погрешность приближения функцией $F(x, K_1, K_2, ..., K_n)$ исходных данных. Вектор F должен быть вектором с символьными элементами, содержащими аналитические выражения для исходной функции и ее производных, по всем параметрам. Вектор VS должен содержать начальные значения элементов вектора K, необходимые для решения системы нелинейных уравнений регрессии интерационным методом (рисунок 3.4).

$$\text{ORIGIN} := 1 \quad F(x, a, b) := a \cdot \exp(-b \cdot x) + a \cdot b$$

$$\frac{d}{da} F(x, a, b) \rightarrow \exp(-b \cdot x) + b \quad \frac{d}{db} F(x, a, b) \rightarrow a \cdot \exp(-b \cdot x) + a$$

$$F_1(x, k) := \begin{pmatrix} k_1 \cdot \exp(-k_2 \cdot x) + k_1 \cdot k_2 \\ \exp(-k_2 \cdot x) + k_2 \\ -k_1 \cdot x \cdot \exp(-k_2 \cdot x) + k_1 \end{pmatrix} \quad VX := \begin{pmatrix} 0.1 \\ 0.5 \\ 1 \\ 1.5 \\ 2 \\ 2.9 \end{pmatrix} \quad VY := \begin{pmatrix} 1.9 \\ 1.6065 \\ 1.34 \\ 1.22 \\ 1.1353 \\ 1.05 \end{pmatrix}$$

$$VS := \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad P := \text{genfit}(VX, VY, VS, F_1) \quad G(x) := F_1(x, P)_1$$

$$i := 1..6 \qquad x := 0, 0.1..3$$

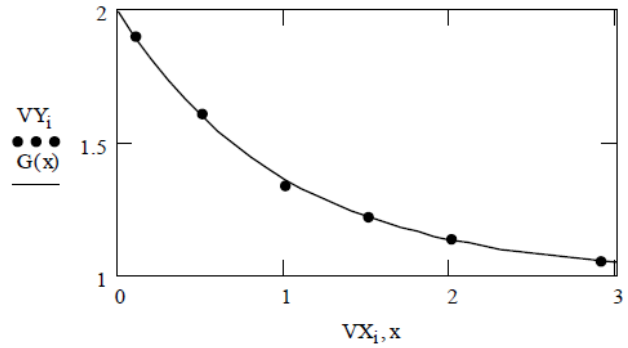


Рисунок 3.4 – Пример выполнения линейной регрессии общего вида с помощью функции F(x,a,b)

Вектор P возвращает значения $a = k_1$ и $b = k_2$ для наилучшего среднеквадратического приближения F(x,a,b).вычисление значений произвольных по переменным a и b проведены средствами символьных операций (первая строка документа рисунок 3.4). замены параметра a на k_1, а параметра b на k_2 связаны с необходимостью применения функции genfit в ее стандартном виде.

Порядок выполнения работы

1. Ознакомиться с особенностями проведения линейной регрессии, ЛРОВ, полиноминальной регрессии, регрессии отрезками полинома второй степени, НРОВ и регрессиями частного вида (sinfit, eapfit, logsfit, lgfit, medfit, pwfit).
2. Для данных рисунка 3.2 провести линейную регрессию.
3. Выполнить синусоидальную регрессию для вектора данных Y(x)=sin(x)-N/10+md(n/5), где N-номер варианта, а x=0…50.
4. Ознакомиться с возможностями многомерной регрессии (привести пример).

Содержание отчета

1. Цель работы, задание;
2. Описание метода решения, краткие сведения из теории (формулы, алгоритм и т.п.);
3. Программу (распечатку), ее описание;

4. Сравнение результатов расчета;
5. Краткие выводы.

Задание

1. Для данных примера 3 провести регрессии следующих видов:

- полиноминальную (2-й и 3-й степени);
- регрессию отрезками полиномов;
- регрессию логарифмической функцией logfit(x,y,q);
- регрессию логистической функцией lgsfit(x,y,q);

2. Дополнить пример 4 регрессией специального вида с помощью функции expfit(x.y.q);

3. Построить графики исходных и аппроксимирующих данных и провести сравнительный анализ с помощью коэффициента корреляции Пирсона.

4. Сделать выводы об эффективности различных видов регрессии.

Контрольные вопросы

1. В чем суть метода наименьших квадратов?
2. По каким критериям можно оценить точность аппроксимации?
3. Укажите последовательность проведения и область применения линейной регрессии для аппроксимации экспериментальных данных.
4. Укажите последовательность проведения и область применения линейной регрессии общего вида.
5. Укажите последовательность проведения и область применения аппроксимации экспериментальных данных полиномами.
6. Укажите последовательность проведения и область применения регрессии отрезками полинома второй степени.
7. Каково значение функции intercept(VX,VY)?
8. Каково значение функции slope(VX,VY)?
9. Каково значение функции linfit(VX,VY,F)?

10. Каково значение функции genfit(VX,VY,VS,F)?

Лабораторная работа №4
Статистическая обработка результатов эксперимента в системе MATLAB

Цель работы: получение навыков статистической обработки данных различными методами в системе MATLAB.

Теоретические сведения

Дискриминантный анализ, как раздел многомерного статистического анализа, включает в себя статистические методы классификации многомерных наблюдений в ситуации, когда исследователь обладает так называемыми обучающими выборками ("классификация с учителем"). Например, для оценки финансового состояния своих клиентов при выдаче им кредита банк классифицирует их по надежности на несколько категорий по ряду признаков. В случае, когда следует отнести клиента к той или иной категории используют процедуры дискриминантного анализа. Очень удобно использовать дискриминантный анализ при обработке результатов тестирования. Так при выборе кандидатов на определенную должность можно всех опрошенных претендентов разделить на две группы - удовлетворяющих и неудовлетворяющих предъявляемым требованиям.

Все процедуры дискриминантного анализа можно разбить на две группы и рассматривать их как совершенно самостоятельные методы. Первая группа процедур позволяет интерпретировать различия между существующими классами, вторая - производить классификацию новых объектов в тех случаях, когда неизвестно заранее, к какому из существующих классов они принадлежат.

Пусть имеется множество единиц наблюдения - генеральная совокупность. Каждая единица наблюдения характеризуется несколькими признаками: x_{ij} - значение j-й переменной i-го объекта (i=1,..., n; j=1,..., p). Предположим, что все множество объектов разбито на несколько подмножеств (два и более). Из каждого подмножества взята выборка объемом n_k, где k - номер подмножества (класса), k = 1,..., q.

Признаки, которые используются для того, чтобы отличать один класс (подмножество) от другого, называются дискриминантными переменными. Число объектов наблюдения должно превышать число дискриминантных переменных: p<n. Дискриминантные переменные должны быть линейно независимыми. Основной предпосылкой дискриминантного анализа является нормальность закона распределения многомерной величины. Это означает, что каждая из дискриминантных переменных внутри каждого из рассматриваемых классов должна быть подчинена нормальному закону распределения.

Основная идея дискриминантного анализа заключается в том, чтобы определить, отличаются ли совокупности по среднему какой-либо переменной (или линейной комбинации переменных), и затем использовать эту переменную, чтобы предсказать для новых членов их принадлежность к той или иной группе. Канонической дискриминантной функцией называется линейная функция:

$$d_{km} = \beta_0 + \beta_1 \cdot x_{1km} + ... + \beta_p \cdot x_{pkm},$$

где: d_{km} - значение канонической дискриминантной функции для m-го объекта в группе k (m=1, ..., n, k=1, ..., g); x_{pkm} - значение дискриминантной переменной X_i для m-го объекта в группе k; $\beta_0, \beta_1, ..., \beta_p$ - коэффициенты дискриминантной функции.

С геометрической точки зрения дискриминантные функции определяют гиперповерхности в p-мерном пространстве. В частном случае при p=2 она является прямой, а при p=3 — плоскостью.

Коэффициенты β_i первой канонической дискриминантной функции выбираются таким образом, чтобы центроиды (средние значения) различных групп как можно больше отличались друг от друга. Коэффициенты второй группы выбираются также, но при этом налагается дополнительное условие, чтобы значения второй функции были некоррелированы со значениями первой. Аналогично определяются и другие функции. Отсюда следует, что любая каноническая дискриминантная функция d имеет нулевую внутригрупповую корреляцию с $d_1, d_2, ..., d_{g-1}$. Если число групп

равно g, то число канонических дискриминантных функций будет на единицу меньше числа групп. Однако по многим причинам практического характера полезно иметь одну, две или же три дискриминантных функций. Тогда графическое изображение объектов будет представлено в одно–, двух– и трехмерных пространствах. Такое представление особенно полезно в случае, когда число дискриминантных переменных p велико по сравнению с числом групп g.

Рассмотрим простой пример. Пусть имеются следующие наблюдения:

x_1	x_2
1	2
2	6
3	1
4	3
5	5

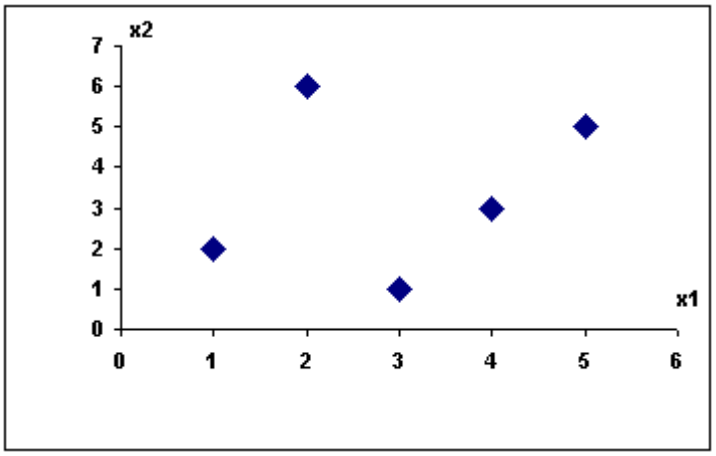


Рисунок 4.1 – Пример получения двух множеств

Из рисунка 4.1 видно, что наблюдения можно разделить на два множества (наблюдения 1,2 и наблюдения 3,4,5). Что бы наилучшим образом разделить множества, нужно построить соответствующую линейную комбинацию переменных x_1 и x_2 - дискриминантную функцию:

$$d = \beta_1 \cdot x_1 + \beta_2 \cdot x_2.$$

Обозначим $\bar{x}_{ij}$ среднее значение j-го признака у объектов i-го множества. Тогда для первого множества среднее значение функции будет равно:

$$\bar{d}_1 = \beta_1 \cdot \bar{x}_{11} + \beta_2 \cdot \bar{x}_{12}$$

для второго множества:

$$\bar{d}_2 = \beta_1 \cdot \bar{x}_{21} + \beta_2 \cdot \bar{x}_{22}.$$

Коэффициенты дискриминантной функции определяются таким образом, чтобы $\bar{d}_1$ и $\bar{d}_2$ как можно больше различались между собой. Требуется максимизировать разность $\bar{d}_1 - \bar{d}_2$. Вектор коэффициентов $B = (\beta_1, \beta_2)^T$ определяется из следующего выражения:

$$B = C^{-1}(\bar{X}_1 - \bar{X}_2),$$

где: C - объединенная ковариационная матрица наблюдений; $\bar{X}_1, \bar{X}_2$ - центроиды первого и второго множеств.

Объединенная ковариационная матрица в общем виде определяется из выражения:

$$C = (X_1^T X_1 + X_2^T X_2) / (n_1 + n_2 - 2),$$

где: X_1, X_2 - матрицы центрированных значений наблюдений двух групп; n_1, n_2 - количество наблюдений в каждой группе.

Для нашего примера матрица C имеет вид:

0,83	2
2	5,3

обратная матрица C^{-1} имеет вид:

13,3	-5
-5	2,1

Вектор B имеет вид:

-3,46
1,3

Таким образом:

$$\bar{d}_1 = \beta_1 \cdot x_{11} + \beta_2 \cdot x_{12} = 3{,}46 \cdot 1{,}5 + 1{,}3 \cdot 4 = 0{,}01;$$
$$\bar{d}_2 = \beta_1 \cdot x_{21} + \beta_2 \cdot x_{22} = 3{,}46 \cdot 4 + 1{,}3 \cdot 3 = -9{,}94,$$

и разность $\bar{d}_1 - \bar{d}_2$ составляет:

$$\bar{d}_1 - \bar{d}_2 = 0{,}01 + 9{,}94 = 9{,}95.$$

Для классификации новых данных нужно определить границу, разделяющую в частном случае две рассматриваемые группы. Такой величиной может быть значение функции, равноудаленное от $\bar{d}_1$ и $\bar{d}_2$:

$$A = (\bar{d}_1 + \bar{d}_2)/2 = (0{,}01 - 9{,}94)/2 = -4{,}965.$$

Величина A называется константой дискриминации. Объекты расположенные выше прямой:

$$-3{,}46 \cdot x_1 + 1{,}3 \cdot x_2 = -4{,}965$$

расположены ближе к центру первой группы и могут быть отнесены к этой группе. Представленная функция называется функцией классификации.

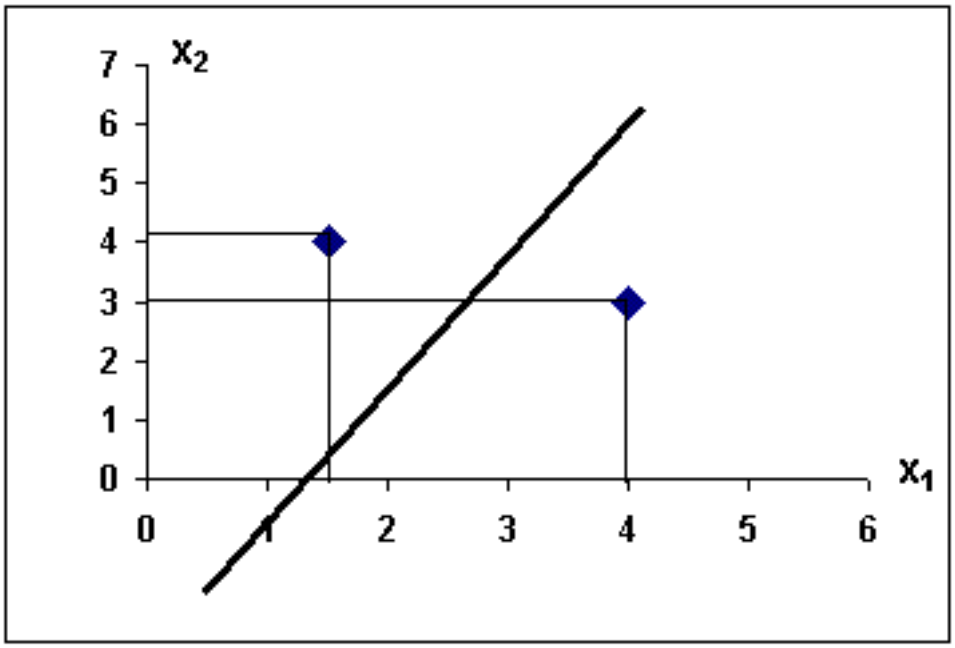


Рисунок 4.2 – Функция классификации

Пусть новое наблюдение имеет значения: $x_1 = 1, x_2 = 3$. Тогда:

$$-3{,}46 \cdot x_1 + 1{,}3 \cdot x_2 = 3{,}46 \cdot 1 + 1{,}3 \cdot 3 = 0{,}44 > -4{,}965$$

и следовательно наблюдение принадлежит первой группе.

Пусть следующее наблюдение имеет значения: $x_1 = 3, x_2 = 1$. Тогда:

$$-3{,}46 \cdot x_1 + 1{,}3 \cdot x_2 = 3{,}46 \cdot 3 + 1{,}3 \cdot 1 = -6{,}48 < -4{,}965$$

и, следовательно, наблюдение принадлежит второй группе.

Дискриминантный анализ. Классификация при наличии k обучающих выборок

При необходимости возможно разбиение множества объектов (наблюдений) на k классов более двух. В этом случае необходимо построение k дискриминантных функций аналогично тому как это представлено в предыдущем параграфе.

Рассмотрим пример. Пусть имеются следующие наблюдения:

x_1	x_2
1	8
2	6
3	10
4	4
5	1
6	4
7	10
8	6
9	8

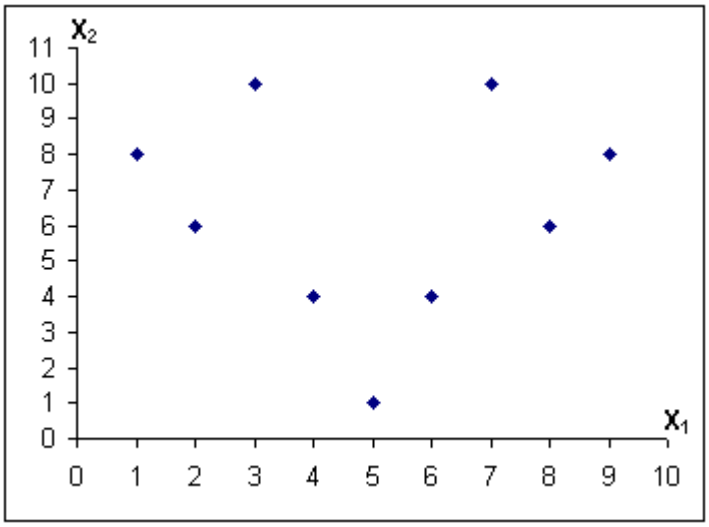


Рисунок 4.3 – Пример получения трех множеств

Наблюдения разделены на три множества (наблюдения 1,2,3, наблюдения 4,5,6 и наблюдения 7,8,9). Что бы наилучшим образом разделить множества, нужно построить соответствующую линейную комбинацию переменных x_1 и x_2 - дискриминантную функцию:

$$d = \beta_1 \cdot x_1 + \beta_2 \cdot x_2.$$

Обозначим x_{ij} среднее значение j-го признака у объектов i-го множества. Тогда для первого множества среднее значение функции будет равно:

$$\bar{d}_1 = \beta_1 \cdot \bar{x}_{11} + \beta_2 \cdot \bar{x}_{12}.$$

для второго множества:

$$\bar{d}_2 = \beta_1 \cdot \bar{x}_{21} + \beta_2 \cdot \bar{x}_{22}.$$

для третьего множества:

$$\bar{d}_3 = \beta_1 \cdot \bar{x}_{31} + \beta_2 \cdot \bar{x}_{32}.$$

Произведя вычисления по рассмотренной ранее схеме получим следующие функции классификации:

для классификации первого и второго множеств:

$$-4 \cdot x_1 + 2 \cdot x_2 = 3;$$

для классификации второго третьего множеств:

$$-6{,}2 \cdot x_1 - 3{,}2 \cdot x_2 = -57{,}8;$$

для классификации первого и третьего множеств:

$$-6{,}4 \cdot x_1 - 0{,}8 \cdot x_2 = -38{,}4.$$

Графики функций представлены на рисунке 4.4.

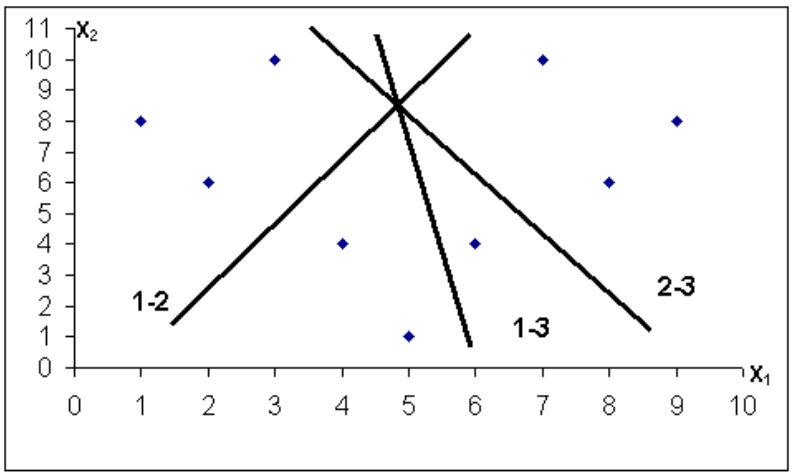


Рисунок 4.4 – График функций

Пусть новое наблюдение имеет значения: $x_1 = 3, x_2 = 7$. Тогда значения дискриминантных функций будут следующими:

$$f_{12} = -4 \cdot x_1 + 2 \cdot x_2 = -4 \cdot 3 + 2 \cdot 7 = -2 > -3;$$
$$f_{23} = -6{,}2 \cdot x_1 - 3{,}2 \cdot x_2 = -6{,}2 \cdot 3 - 3{,}2 \cdot 7 = -41 > -57{,}8;$$
$$f_{13} = -6{,}4 \cdot x_1 - 0{,}8 \cdot x_2 = -6{,}4 \cdot 3 - 0{,}8 \cdot 7 = -24{,}8 > -38{,}4.$$

Если $x_1 = 7, x_2 = 3$, то:

$$f_{12} = -4 \cdot x_1 + 2 \cdot x_2 = -4 \cdot 7 + 2 \cdot 3 = -22 < -3;$$
$$f_{23} = -6{,}2 \cdot x_1 - 3{,}2 \cdot x_2 = -6{,}2 \cdot 7 - 3{,}2 \cdot 3 = -53 > -57{,}8;$$
$$f_{13} = -6{,}4 \cdot x_1 - 0{,}8 \cdot x_2 = -6{,}4 \cdot 7 - 0{,}8 \cdot 3 = -47{,}2 < -38{,}4.$$

Необходимо отметить, что изменение числа переменных существенно влияет на результат дискриминантного анализа. Поэтому для определения необходимости включения (исключения) дискриминантной переменной обычно используют специальные критерии.

Для оценки вклада отдельной переменной в значение дискриминантной функции целесообразно пользоваться стандартизованными коэффициентами дискриминантной функции. Вычислить стандартизованные коэффициенты можно по следующей формуле:

$$a_j = \beta_j \cdot (c_{ij} / (p - m))^{1/2},$$

где: p - общее число исходных переменных; m - число групп; c_{ij} - элементы ковариационной матрицы.

Стандартизованные коэффициенты не зависят от масштабов измерений переменных. Именно поэтому по абсолютным величинам этих коэффициентов можно определить какая дискриминантная переменная вносит наибольший или наименьший вклад в значение дискриминантной функции.

Порядок выполнения работы

Порядок выполнения лабораторной работы будет рассмотрен на примере двух кластеров A01 и A02, которые состоят из ста объектов, характеризующихся двумя признаками x1 и x2.

Объекты каждого кластера можно представить в виде множества точек на плоскости. По оси абсцисс откладывается значение первого признака, по оси ординат – второго.

```
figure
hold on
plot(A01(:,1),A01(:,2),['b','.'])
plot(A02(:,1),A02(:,2),['r','.'])
hold off
set(gca,'XLim',[0 1],'YLim',[0 1])
```

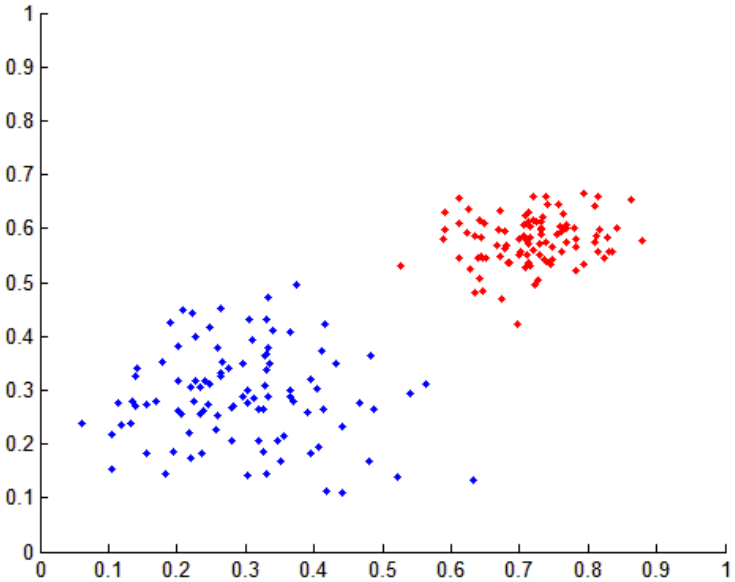


Рисунок 4.5 - Два кластера A01 и A02, которые состоят из ста объектов, характеризующихся двумя признаками x1 и x2

1 Линейные дискриминантные функции

1.1 Вектор проекций.

Основной целью дискриминантного анализа является нахождение такой линейной комбинации переменных (координат вектора проекций), при которой оптимально разделяются

исследуемые кластеры, т.е. при которой расстояние между проекциями кластеров на вектор максимально. В каноническом дискриминантном анализе расстояние между проекциями оценивается через разность математических ожиданий, в дискриминантном анализе Фишера – через дискриминант Фишера (критерий J).

Программная реализация описанных методов в рамках решаемой задачи одинакова. При этом следует помнить, что поиск координат вектора проекций и последующие расчеты осуществляются на обучающей выборке или на части контрольной выборки (в рассматриваемом примере A1 и A2).

```
N1=length(A1(:,1));%Количество объектов в кластере A1
N2=length(A2(:,1));%Количество объектов в кластере A2
M1=mean(A1);%Мат. ожидание кластера A1
M2=mean(A2);%Мат. ожидание кластера A2
      X1=zeros(N1,2);
for i=1:N1
X1(i,:)=A1(i,:)-M1;
end
S1=X1'*X1;%Ковариационная матрица кластера A1
X2=zeros(N2,2);
for i=1:N2
      X2(i,:)=A2(i,:)-M2;
end
S2=X2'*X2;%Ковариационная матрица кластера A2
S=S1+S2;%Объединенная ковариационная матрица
W=abs(S*(M1-M2)');%Вектор проекций
figure
plot(A1(:,1),A1(:,2),'.')
hold on
plot(A2(:,1),A2(:,2),['.','r'])
plot([0 W(1)],[0,W(2)],'r')
```

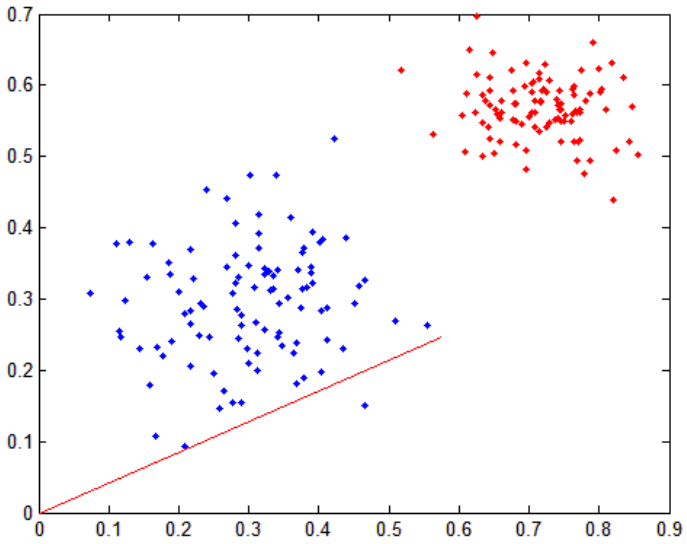


Рисунок 4.6 – Вектор проекций

1.2 Гистограмма проекций.

Чтобы определить оптимальное положение разделяющей прямой, необходимо спроецировать объекты кластеров на полученный вектор проекций.

```
Xp1=zeros(1,N1);
for i=1:N1
        Xp1(i)=A1(i,:)*W;%Массив проекций кластера A1
end
Xp2=zeros(1,N2);
for i=1:N2
        Xp2(i)=A2(i,:)*W;%Массив проекций кластера A2
end
[y1,x1]=hist(Xp1,10);
[y2,x2]=hist(Xp2,10);
figure
hold on
bar(x1,y1,'b')
bar(x2,y2,'r')
hold off
```

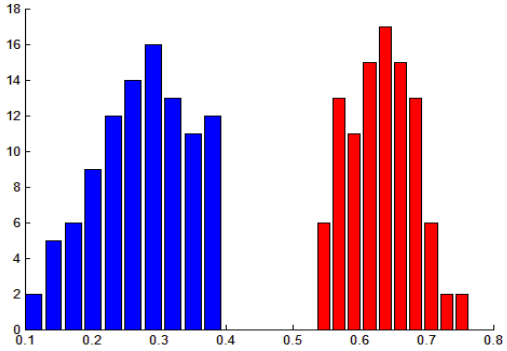


Рисунок 4.7 – Гистограмма проекций

1.3 Положение разделяющей прямой.

Положение разделяющей прямой можно найти двумя способами: методом минимизации ошибки (1) и методом минимизации стоимости ошибки (2).

При решении задачи методом 2 (для случая нормального распределения данных, равновероятного появления объектов каждого кластера и одинаковой стоимости ошибочной верификации объектов каждого кластера) положение разделяющей прямой определяется как среднее арифметическое математических ожиданий проекций.

```
m1=mean(Xp1); %мат. ожидание проекций кластера A1
m2=mean(Xp2); %мат. ожидание проекций кластера A2
a=(m1+m2)/2; %разделяющая прямая
```

При решении задачи методом 1 последовательно перебираются возможные значения разделяющей прямой, для них вычисляются ошибки верификации. Оптимальным считается такое положение разделяющей прямой, при котором ошибка минимальна.

```
a0=min(min(Xp1),min(Xp2)):0.01:max(max(Xp1),max(Xp2));
Na=length(a0);
e=zeros(1,Na);
e1=zeros(1,Na);
e2=zeros(1,Na);
for i=1:Na
        for j=1:N1
                if Xp1(j)>=a0(i)
```

```
                    e1(i)=e1(i)+1;%ошибка первого рода
                        end
                    end
            for k=1:N2
                if Xp2(k)<a0(i)
                    e2(i)=e2(i)+1;%ошибка второго рода
                        end
                    end
e(i)=e1(i)+e2(i);%суммарная ошибка
end
[em,im]=min(e);%минимум ошибки
plot(e)
a=a0(im);%разделяющая прямая
figure
hold on
bar(x1,y1,'b')
bar(x2,y2,'r')
stem(a,20,['g','.'])
hold off
```

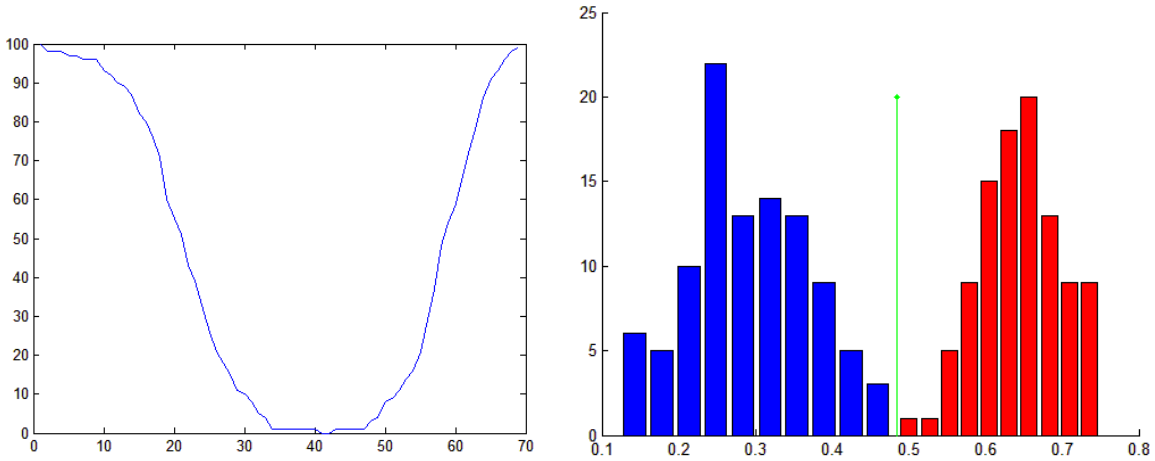


Рисунок 4.8 – Положения разделяющей прямой

1.4 Разделяющая прямая в пространстве признаков.

В пространстве признаков разделяющая прямая перпендикулярна вектору проекций и пересекает его на расстоянии, найденном в предыдущем пункте, от начала координат.

```
%Поиск координат ортогонального вектора осуществляется по уравнению:
%ax*W(1)+ay*W(2)=0
ax=1/10;%абсцисса вектора, ортогонального вектору проекций
ay=-ax*W(1)/W(2);%ордината вектора, ортогонального вектору проекций
aa=a*pinv(W);%смещение ортогонального вектора вдоль вектора проекций на расст-е a
figure
hold on
plot(A1(:,1),A1(:,2),'.')
plot(A2(:,1),A2(:,2),['.','r'])
plot([0 W(1)],[0,W(2)],'k')
plot([aa(1) ax+aa(1)],[aa(2),ay+aa(2)],'g')
hold off
```

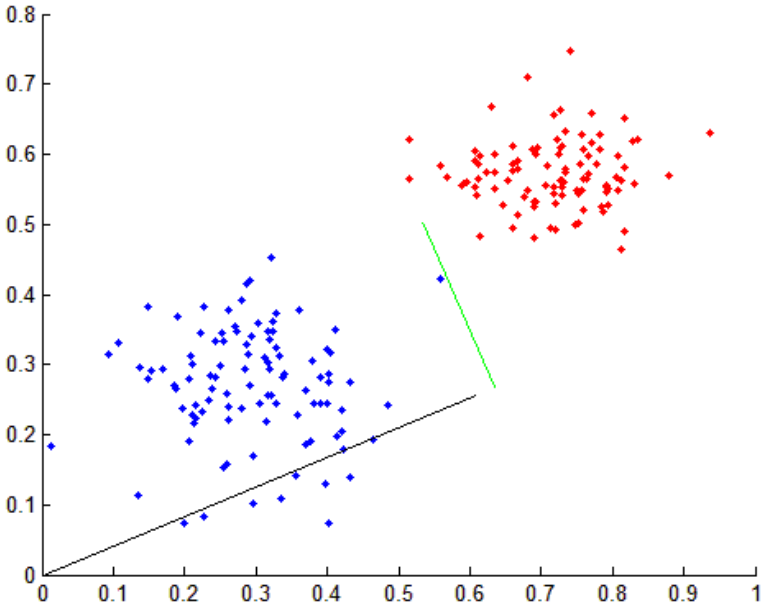


Рисунок 4.9 – Разделяющая прямая в пространстве признаков

1.5 Ошибки кластеризации.

Вычисление ошибок производится для контрольной выборки или, в случае если в качестве обучающей выборки была взята часть исследуемых данных, для всей выборки.

```
E1=0;
E2=0;
```

```
for i=1:N1
        if A01(i,:)*W1>=a
                E1=E1+1;%ошибка первого рода
                        end
end
for i=1:N2
        if A02(i,:)*W1<a
                E2=E2+1;%ошибка второго рода
                        end
end
E=E1+E2;%суммарная ошибка
```

2 Непараметрические методы

2.1 Вектор проекций.

Целью непараметрических методов является нахождение таких координат вектора проекций, при которых проекции одного кластера будут положительными, а другого – отрицательными. Таким образом, кластеризации проводится при сравнении значений проекций с нулем. Поиск координат вектора осуществляется путем последовательного пересчета их значений относительно координат текущего объекта и заданного коэффициента. Коэффициент может быть постоянным (метод непараметрического обучения с постоянным коэффициентом) или переменным (методы непараметрического обучения с абсолютной или частичной коррекцией).

При программной реализации методов можно столкнуться с ситуацией, когда невозможно разделить кластеры с абсолютной точностью, что влечет за собой бесконечную цикличность процесса. Чтобы избежать такой ситуации можно задать допустимый уровень ошибки (п. 2.1.2) или максимальное число итераций (п. 2.1.3).

```
clear X1 X2 W1
X1(:,1:2)=A1;
X1(:,3)=1;%Добавление столбца, заполненного единицами в кластер A1
```

```
X2(:,1:2)=A2;
X2(:,3)=1;%Добавление столбца, заполненного единицами в кластер A2
X2=X2*(-1);%Умножение второго кластера на -1
X=[X1; X2];%Объединение матриц
N=length(X(:,1));
```

2.1.1 С постоянным коэффициентом.

```
C1=1
W1=[1,1,1];%Начальное значение вектора проекций
i=0;
e1=1;
while e1>0
    i=i+1;
    if i>N
    i=1;
    end
        e1=0;
        if X(i,:)*W1'<0
        W1=W1+C1*X(i,:);  %Пересчет координат вектора проекций
        end
    for j=1:N
    if X(j,:)*W1'<0
        e1=e1+1; %Уровень ошибки
            end
                end
end
```

2.1.2 С абсолютной коррекцией.

```
W2=[1,1,1];%Начальное значение вектора проекций
i=0;
e2=1;
while e2>=10%Допустимый уровень ошибки
    i=i+1;
    if i>N
```

```
        i=1;
        end
        e2=0;
        if X(i,:)*W2'<0
        C2=abs(W2*X(i,:)')/(X(i,:)*X(i,:)');
        W2=W2+C2*X(i,:);    %Пересчет    координат    вектора
проекций
        end
              for j=1:N
                    if X(j,:)*W2'<0
                          e2=e2+1; %Уровень ошибки
                                end
                          end
end
```

2.1.3 С частичной коррекцией.

```
W3=[1,1,1];%Начальное значение вектора проекций
i=0;
e3=1;
M=0;
while e3>0&&M<1000%Максимальноечисло итераций
        i=i+1;
        if i>N
              i=1;
        end
        e3=0;
        if X(i,:)*W3'<0
              C3=2*abs(W3*X(i,:)')/(X(i,:)*X(i,:)');
              W3=W3+C3*X(i,:);  %Пересчет  координат  вектора
        проекций
        end
        for j=1:N
              if X(j,:)*W3'<0
                    e3=e3+1; %Уровень ошибки
                          end
        end
        M=M+1; %Счетчик итераций
```

end

2.2 Гистограмма проекций.

Аналогично п. 1.2.

2.3 Положение разделяющей прямой.

В случае абсолютной разделимости кластеров проекция разделяющей прямой проходит через ноль. Если же кластеры невозможно разделить с нулевой ошибкой (если приходится задавать допустимый уровень ошибки или максимальное число итераций), положение разделяющей прямой приходится уточнять с помощью метода минимизации ошибки или метода минимизации стоимости ошибки (п.1.3).

2.4 Разделяющая прямая в пространстве признаков.

Аналогично п. 1.4.

2.5 Ошибки кластеризации.

Аналогично п. 1.5.

Содержание отчета

1. Цель работы, задание;
2. Описание метода решения, краткие сведения из теории (формулы, алгоритм и т.п.);
3. Программу (распечатку), ее описание;
4. Сравнение результатов расчета;
5. Краткие выводы.

Задание

1. Используя генератор случайных чисел пакета MathCad создайте как обучающие, так и контрольные выборки с объемом и размерности пространства информативных признаков, заданные преподавателем. Пример листа MathCad, в котором создается выборка из 28 объектов с нормально распределенным законом распределения признаков (признаковое пространство четырехмерное) показан на рисунке 4.10.

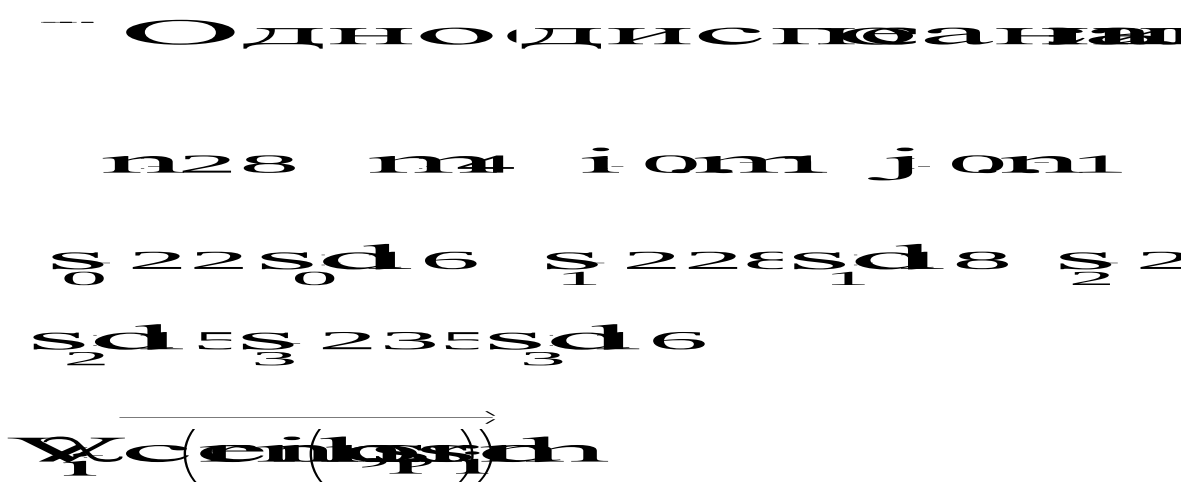


Рисунок 4.10. - Лист MathCad с генератором случайных чисел

Каждый информативный признак в выборке представлен вектором-столбцом Vλ из 28 элементов. Случайные числа, распределенные по нормальному закону распределения, генерируются функцией

rnorm(n, s, sd),

где n-число элементов в выборке, s – математическое ожидание информативного признака, sd -дисперсия информативного признака.

Варьируя эти параметры (s и sd) вы можете менять структуру распределения классов в признаковом пространстве и исследовать эффективность классификации при различных классовых структурах.

Функция ceil округляет случайное число до ближайшего целого.

Значок $\longrightarrow$ обозначает операцию векторизации, то есть одновременное выполнение скалярной операции над всеми элементами вектора.

2. В соответствии с порядком выполнения работы проведите статистическую обработку результатов эксперимента.
3. Объясните полученные результаты.
4. Сделайте выводы.

Контрольные вопросы

1. Что такое дискриминантный анализ?
2. Какова основная идея дискриминантного анализа?
3. Укажите последовательность проведения дискриминантного анализа.
4. Укажите последовательность проведения классификации при наличии k обучающих выборок.
5. Как в системе MATLAB программно реализуется вектор проекций?
6. Как в системе MATLAB программно реализуется гистограмма проекций?
7. Как в системе MATLAB программно реализуется разделяющая прямая в пространстве признаков?
8. Как создать обучающие и контрольные выборки, используя генератор случайных чисел пакета MathCad?

Лабораторная работа №5
Основы программирования в MATLAB

Цель работы: изучение методики проектирования программ цифровой обработки сигналов (ЦОС): моделирования сигналов, нормально распределенного и белого шума, программирования операций математической обработки, операций графического вывода данных, проектирования средств графического пользовательского интерфейса.

Теоретические сведения

Программные средства систем ЦОС обычно создаются на языке C/C++, так как в этом языке предусмотрены средства, обычно необходимые при аппаратной реализации на базе программируемых логических интегральных схем (ПЛИС) и сигнальных процессоров: удобные средства работы с битами, логические операции, средства работы с аппаратными прерываниями и др. Кроме того, компилятор С обеспечивает формирование исполняемой программы, близкой по скорости исполнения к программе на ассемблере.

При разработке систем жесткого реального времени на базе ПЛИС и быстродействующих сигнальных процессоров необходимо производить отработку алгоритмов ЦОС на универсальных средствах (с помощью компьютерных программ) используя низкоуровневое программирование без библиотечных функций (на C/C++) с тем, чтобы на следующем этапе можно было реализовать разработанные алгоритмы ЦОС на ПЛИС и сигнальных процессорах. Кроме того, как известно, ЦОС должна быть выполнена в течение критического срока обслуживания и это так же важно, как и корректность алгоритма ЦОС. Невыполнение задачи реального времени в течение критического срока обслуживания равносильно невыполнению задачи в целом. Поэтому очень важно, чтобы на втором этапе разработки отработка алгоритмов ЦОС производилась с использованием соответствующей аппаратуры и скорость выполнения операций

ЦОС контролировалась, например, с использованием системного таймера компьютера.

Конечной целью разработки и исследования различных алгоритмов ЦОС является их практическая реализация в устройствах на базе ПЛИС и сигнальных процессоров, таких, например, как модуль ЦОС SAMC-401, содержащий ПЛИС серии Virtex-4, в которой интегрированы два процессора Power PC с тактовой частотой до 450 МГц, и сигнальный процессор Texas Instruments TMS320C6455, работающий на частоте 1.2ГГц с субмодулем АЦП SAMC-ADC, реализующий аналого-цифровое преобразование 12/14 бит с предельной частотой тактирования до 210 МГц.

Основные стадии разработки алгоритмов ЦОС:

1. Высокоуровневая программная (MATLAB).
2. Низкоуровневая программная (LabWindows/CVI, C/C++).
3. Аппаратно-программная (LabWindows/CVI, C/C++).
4. Аппаратная (VHDL, C/C++).

На первом этапе разработки наиболее совершенным средством является MATLAB, т.к. он содержит библиотеки функций для сложных видов математической обработки, таких как быстрое преобразование Фурье, цифровой фильтрации, корреляционной обработки и др., библиотеки программ для создания объектов графического пользовательского интерфейса (панелей, кнопок управления, окон цифрового ввода/вывода, окон графического вывода и др.).

Программные средства, создаваемые для работы в среде MATLAB, содержат две составляющие:

1. Собственно программу (mat-файл) на языке С, в которой содержатся необходимые функции математической обработки, отображения таблиц и графиков. При ее создании следует пользоваться описаниями функций библиотек математической обработки и графического пользовательского интерфейса.

2. Файлы ресурсов графического пользовательского интерфейса <имя программы>.fig.

Создаваемая пользователем прикладная программа представляет проект, содержащий два файла:

- файл основной программы <имя>.mat;

- файл макета <имя>.fig;

Программа на языке С может быть написана и редактироваться пользователем. Файлы макета <имя>.fig создаются автоматически при создании и редактировании пользователем графических панелей. Эти файлы нельзя редактировать!

Создание средств графического пользовательского интерфейса в традиционных системах программирования, таких как Visual C++ возможно, но достаточно сложно, так как для этого необходимо создание большого количества нестандартных графических объектов.

Создание прикладной программы ЦОС в среде MATLAB

1. Создать панель интерфейса пользователя: File/New/GUI/Blank GUI. В результате появится всплывающее диалоговое окно untitled.fig

2. Разместить элементы управления (кнопки управления PushButton, окна ввода/вывода текста EditText, окна графического вывода Axes и др.), например, так, как показано на рисунке 5.1 и

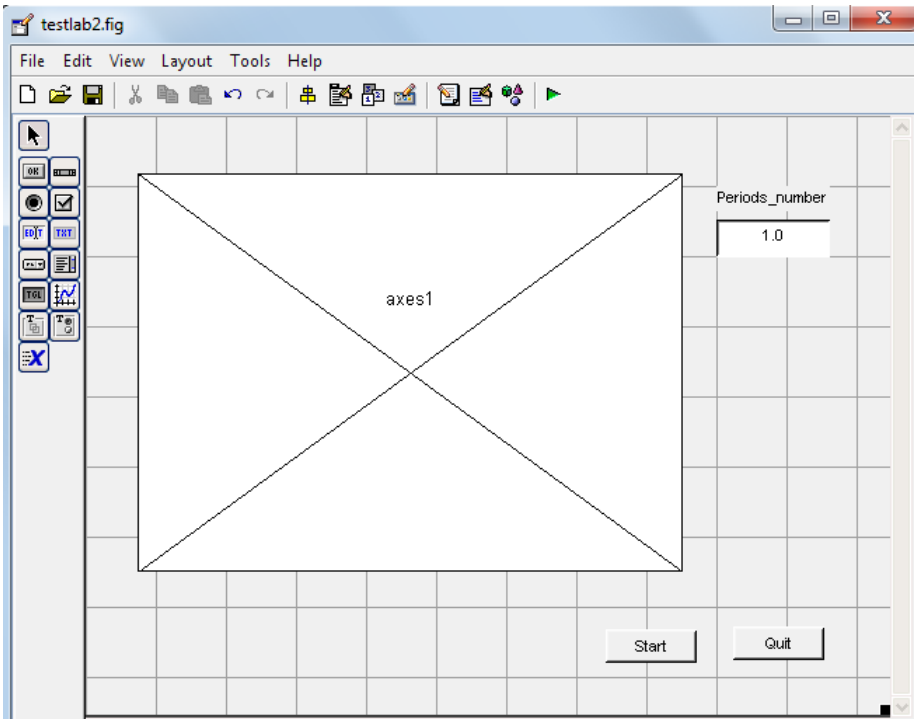


Рисунок 5.1 – Панель интерфейса пользователя

задать их атрибуты. Панель атрибутов элемента управления вызывается двойным щелчком левой кнопки мыши на изображении элемента. Основные атрибуты кнопки управления и окна ввода/вывода текста – Callback, CreateFcn и *tag*. Например, при задании атрибуту *tag* окна ввода/вывода текста значения freq имя окна будет freq. Такое же имя может быть дано и переменной, значение которой читается из этого окна.

Атрибуты Callback, CreateFcn создаются автоматически после задания *tag*.

3. Сохранить untitled.fig в файле с именем <имя>.fig. В результате будет не только сохранен файл интерфейса пользователя, но и создан и сохранен в файле <имя>.m «скелет» программы.

4. Создать содержательную часть программы в блоке функции, вызываемой, например, нажатием кнопки управления (Приложение 2).

В итоге будут созданы два файла: файл программы с расширением .m и файл GUI с расширением .fig. Для внесения изменений в интерфейс пользователя нужно вызвать файл командой >>guide (‘<полное имя файла, включая путь>.fig’).

Загрузка и запуск MATLAB

Для загрузки и запуска LabWindows CVI 8.0 в среде ОС Windows XP/7 нужно вызвать из меню «Пуск»:

Programs/MATLAB

Создание программы

Создать файл макета.

а) File/New/GUI/Blank GUI

б) сохранить созданный файл макета File/Save Untitled.fig As.../{имя файла}

В результате на экране дисплея появляется изображение панели GUI. Размеры и место расположения панели на экране

можно установить с помощью мыши. Далее нужно установить атрибуты панели. Для этого перейти в режим редактирования панели двойным щелчком мыши и во всплывающей панели задать:

- Name - идентификатор панели, который определяет имя m-файла и fig-файла.
- и другие.

4. Создать элементы GUI (графические окна вывода, текстовые окна ввода/вывода, командные кнопки и т.д.) из библиотеки элементов и разместить их на созданной панели.

Объекты GUI выбираются из меню и размещаются на экране методом "drag and drop" с помощью "мыши".

MATLAB предоставляет широкий выбор выбор объектов:

Edit text - окно ввода/вывода текста;

Static Text – окно вывода текстового сообщения;

Push Button – кнопка с двумя состояниями;

Toggle Button – кнопка с двумя состояниями (включено/выключено);

Check Box – флажок;

Radio Button – аналог Option Button;

Pop-up-menu – меню;

Slider – линейка прокрутки;

List box – поле со списком

Axis1 - окно вывода массивов данных в графической форме;

При выполнении лабораторной работы рекомендуется использовать следующие элементы управления и ввода/вывода: Edit text, Axis1, Pushbutton.

4.1 Задание атрибутов элементов GUI.

Сделать двойной щелчок левой клавишей мыши на объекте GUI, далее во всплывающей панели задать его атрибуты, в первую очередь tag.

Далее нужно сохранить сделанные установки.

File/Save

/Save As.../{имя файла}

5. Создать программный код. Создание fig-файла производится автоматически при сохранении GUI-образа. Одновременно создается m-файл.

В созданном программном коде m-файла объектам GUI будет соответствовать " программная оболочка" функций на языки. C++. Эти функции (за исключением функции работы таймера) будут запускаться на исполнение пользователем с панели GUI. В приложении 1 приведен пример созданной таким путем «программной оболочки».

Далее в эту оболочку вносится программный код прикладной программы на языке C++.

Если производится редактирование ранее уже созданного программного кода, например, при добавлении какого-либо объекта GUI, то для добавления в ранее созданный программный код изменений, связанных с добавленными элементами GUI нужно открыть файл <>.fig для редактирования командой:

>>guide (‘<полное имя файла, включая путь>.fig’)

в окне Command Window. Путь к файлу можно не указывать, если поместить его в папку MATLAB.

6. Добавить в программу функцию выхода.

Для этого в уже сгенерированную по п.5 функцию выхода программы необходимо внести программный код:

```
fclose('all');
close ('all');
```

7. Добавить в функцию, вызываемую кнопкой на панели GUI (в приведенном примере - Start), строки обращения к элементам ввода/вывода GUI, текст программы. Ниже приведен пример программирования ввода текста из окна ввода/вывода (TextBox) с преобразованием в число типа double, вывода значения с преобразованием в текст, чтения номера элемента из поля со списком (ComboBox), вывода числового значения в TextBox:

```
periods_number=str2double(get(handles. periods_number,'string'));
                    set(handles.chastotan,'String',fr_int);
                    regim=get(handles.regim,'Value');
                    set(handles.lampa,'Value',1);
```

Примечания:

1. Элементы поля со списком нумеруются, начиная с 1. Так же в MATLAB нумеруются и элементы массивов.
2. В атрибутах окна, предназначенного для вывода текстовых сообщений задать атрибут Style –Edit.
3. Для имитации сигнального индикатора можно использовать элемент radiobutton.

Описания типов переменных не обязательны, но возможны, если требуется. В случае необходимости передавать значения параметров из одной функции в другую нужно использовать описания глобальных переменных, например:

```
global N.
```

В результате будет получен полный программный код прикладной программы. Простой пример такой программы генерации и графического отображения массива случайных чисел приведен ниже. Количество периодов сигнала задается переменной freq, получаемой с элемента GUI:

```
freq=str2double(get(handles.freq,'string'));
for i=1:1000
    y(i)=sin((2*pi*i*freq)/1000.0);
end
i=1:1000;
plot(i,y);
```

Редактирование программного кода, в том числе строк обращения к объектам GUI производится точно так же, как в любом C/C++.

10. Запустить программу.

Debug/Run

Если компилятор найдет ошибки, то устранить ошибки и повторить предыдущее действие.

Порядок выполнения работы

1. Изучить состав и функции библиотек программ MATLAB для работы с аппаратурой, математической обработки, создания средств графического пользовательского интерфейса.
2. Разработать программу генерации синусоидального сигнала.
3. Разработать программу генерации стандартных сигналов: гармонического, пилообразного, треугольного, прямоугольного с нормально распределенным и белым шумом. Выбор вида сигнала, амплитуду сигнала и уровень шума сделать регулируемыми.

Задание

1. Изучить основы программирования в MATLAB по описанию к лабораторной работе и пользуясь справочной системой.
2. Создать графический пользовательский интерфейс и программу генерации синусоидального сигнала. Количество периодов сигнала сделать регулируемым.
3. Разработать прикладную программу генерации различных стандартных сигналов:

- ♦ гармонического (синусоидального);
- ♦ пилообразного;
- ♦ треугольного;
- ♦ прямоугольных импульсов.

без шума и с нормально распределенным и белым шумом. Выбор вида сигнала, амплитуду, частоту сигналов и среднеквадратическую величину шума сделать регулируемыми.

3.1. Для генерации нормально распределенного и белого шума использовать функции randn и wgn. Пример фрагмента программы генерации модельного сигнала с шумом приведен ниже.

```
%Генерация нормального и белого шума
  %noise=randn(points_number);%нормально распределенный
   noise=wgn(points_number,1,0);%белый
for i=1:points_number %генерация модельного сигнала с шумом
```

```
    x(i)=sin(2*3.14*kp*i/kt)+noise_sko*noise(i);
end
```

Здесь points_number – количество элементов массива (количество значений сигнала), noise_sko – среднеквадратическое значение шума.

3.2. На панели графического пользовательского интерфейса предусмотреть (рисунок 5.2):

- ♦ переключатель выбора вида сигнала;
- ♦ элемент ввода уровня шума;
- ♦ элемент ввода количества точек за период генерируемого сигнала;
- ♦ окно графического вывода генерируемого сигнала.

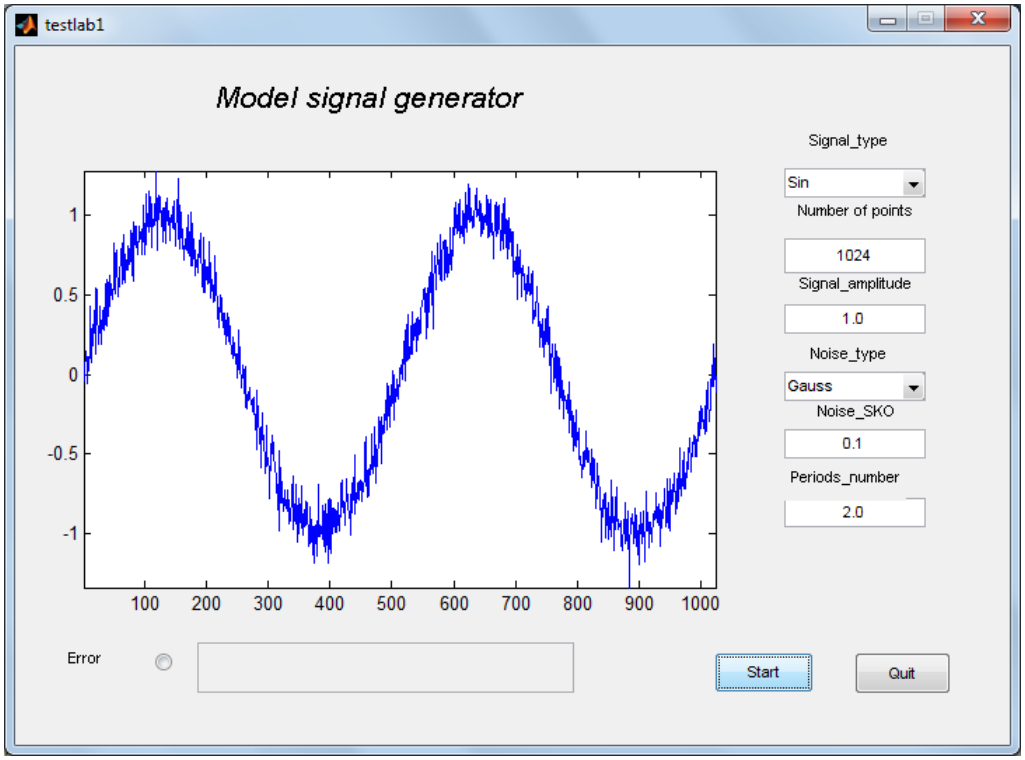


Рисунок 5.2 – Панель графического пользовательского интерфейса

3.3. Дополнить программу, предусмотрев в ней возможность накопления (суммирования с усреднением) генерируемых сигналов.

На панели графического пользовательского интерфейса дополнительно предусмотреть переключатель задания количества накоплений.

Содержание отчета

1. Задание к работе.
2. Текст программы с комментариями.
3. Вид панели интерфейса пользователя при генерации различных сигналов с различным шумом.

Контрольные вопросы

1. Перечислите основные стадии разработки алгоритмов ЦОС.
2. Какие две составляющие содержат программные средства, создаваемые для работы в среде MATLAB?
3. Укажите последовательность создания пользовательского интерфейса.
4. Укажите последовательность задания атрибутов элементов GUI.
5. Как можно произвести генерацию нормально распределенного и белого шума?

Лабораторная работа №6
Обработка результатов однофакторного эксперимента в среде MATLAB. Подгонка кривых

Цель работы: приобретение навыков обработки результатов однофакторного эксперимента в среде MATLAB/

Теоретические сведения

Обработка результатов однофакторного эксперимента включает 3 этапа:

- Выбор вида аппроксимирующей зависимости.
- Расчет коэффициентов аппроксимирующей зависимости.
- Проверку адекватности полученной зависимости экспериментальным данным.

Наиболее совершенным программным средством для решения данной задачи является пакет программ MATLAB. MATLAB является средой разработки программ обработки данных и, одновременно, содержит большое количество готовых программ, в частности, программу подгонки кривых CurveFitting.

Программа подгонки кривых CurveFitting предоставляет следующие виды функций для аппроксимации и интерполяции одномерных массивов данных:

а) экспоненциальную;

$$y = ae^{bx}$$

$$y = ae^{bx} + cd^{dx}$$

б)гауссиан

$$y = \sum_{i=1}^{n} a_i e^{\left[-\left(\frac{x-b_i}{c_i}\right)^2\right]}$$

в)Фурье

$$y = a_0 + \sum_{i=1}^{n} a_i \cos(n\omega x) + b_i \sin(n\omega x)$$

г)полиномиальную;

$$y = \sum_{i=1}^{n=1} p_i x^{n+1-i}$$

д)показательную;

$$y = ax^b$$

$$y = a + bx^c$$

е)рациональную.

$$y = \frac{\sum_{i=1}^{n+1} p_i x^{n+1-i}}{x^m + \sum_{i=1}^{m} q_i x^{m-i}}$$

Критерием качества подгонки являются вычисляемые программой коэффициенты множественной детерминации (R-square и Adjusted R-square). Коэффициент R-square показывает насколько успешно разброс данных относительно аппроксимирующей кривой может быть объяснен наличием случайных погрешностей в данных. Максимально возможное значение R-square равно 1.

Коэффициент Adjusted R-square характеризует то же самое, но учитывает количество степеней свободы дисперсий воспроизводимости и адекватности. Поэтому Adjusted R-square лучше характеризует степень соответствия экспериментальных данных аппроксимирующей кривой в том случае, если мы увеличиваем степень аппроксимирующего полинома и хотим проверить, происходит ли при этом улучшение качества подгонки. Максимально возможное значение Adjusted R-square также равно 1.

$$SSE = \sum_{i=1}^{n} w_i(y_i - \hat{y}_i)^2$$ SSE – sum square error

$$SST = \sum_{i=1}^{n} w_i(y_i - \bar{y})^2$$ SST – sum square total,

$w_i = \frac{1}{\sigma_i^2}$ - коэффициент, выравнивающий степень разброса данных в опытах;

$\hat{y}$ - значение полученное по аппроксимирующей зависимости

$$R - square = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$Adjusted\ R - square = 1 - \frac{SSE(n-1)}{SST(n-m)}$$

где n- количество результатов измерений;

m – количество членов в уравнении регрессии;

n-1 – количество степеней свободы дисперсии SSE;

n-m - количество степеней свободы дисперсии SST.

Порядок выполнения работы

1. Изучить теоретические основы подгонки кривых, положенные в основу программы Curve Fitting и описание библиотеки аппроксимирующих функций Fitting Library в MATLAB Help: CurveFittingToolbox/FittingData/ParametricFitting (Library Models, Custom Equations, Evaluation of goodness of Fitting и др.).

2. Выполнить подгонку кривых, выбрать наилучшую функциональную зависимость и обосновать сделанный выбор для зашумленного массива данных: файл censuN.mat) в папке E:\DSP_LAB_2012\Lab_4_MATLAB.

3. При работе с программой CurveFitting подгонку кривых производить в следующей последовательности:

а) Запустить MATLAB.

б) открыть окно Workspace (View/Workspace) и загрузить в это окно mat-файл данных по заданию преподавателя (census.mat, censu1.mat, censu2.mat и т.д.)

в)открыть панель CurveFitting для чего выбрать Start/ Toolboxes/ CurveFitting в окне MATLAB;

г) на панели CurveFittingTool активизировать кнопку Data. В окнах Xdata, Ydata станут доступными составляющие cdate (X) и pop (Y) из файла census.mat;

д) произвести выбор cdate и pop, затем активизировать кнопку CreateDataSet;

е) активизировать кнопку Fitting;

ж) выбрать вид аппроксимирующей функции и активизировать кнопку Apply;

Результаты аппроксимации будут представлены в графической форме в окне графического вывода на панели CurveFittingTool (рисунок 6.1) и в численной форме в окне Results на панели Tables of Fits (рисунок 6.2).

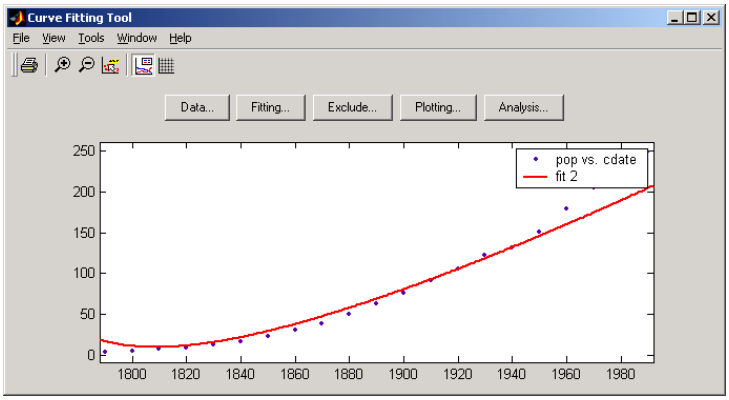


Рисунок 6.1 – Результаты аппроксимации на панели CurveFittingTool

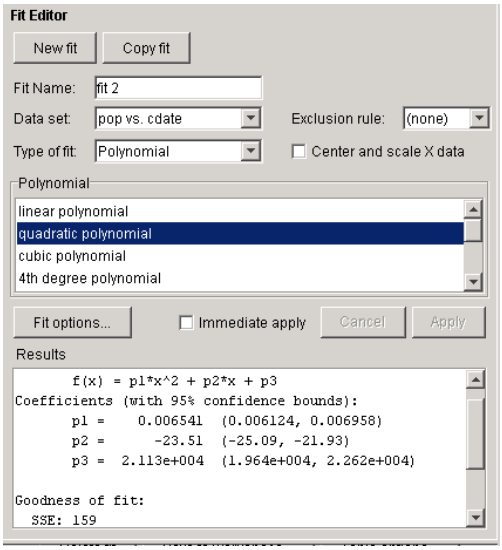


Рисунок 6.2 - Результаты аппроксимации в численной форме в окне Results на панели Tables of Fits

з) выбрать пункт меню View/Residuals/Line или View/Residuals/Scatter. В результате в окне графического вывода на панели CurveFittingTool будут одновременно выведен график ошибки интерполяции (пример на рисунке 6.3).

Окно графического вывода можно вывести на печать или выполнить в виде рисунка: File/Print to Figure.

Полученные в процессе аппроксимации результаты занести в таблицу:

Вид модели	Порядок модели	Качество подгонки (+ / -)	Диапазон погрешностей	R-square	Adjusted R-square	RMSE

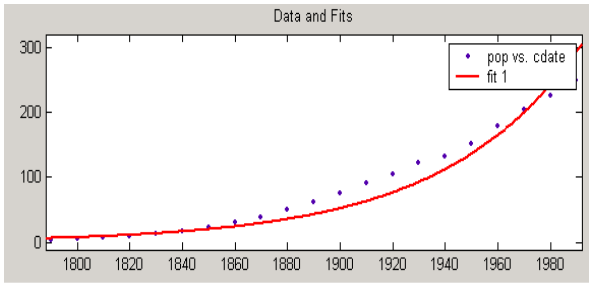
Исследовать таким образом все возможные виды моделей, порядок модели - до 5.

е) по полученным и отраженным в таблице данным выбрать три наилучшие варианта аппроксимирующих функций. Для этих вариантов получить значения график аппроксимирующей функции и график погрешности аппроксимации, аналитические выражения с численными значениями коэффициентов для аппроксимирующей зависимости.

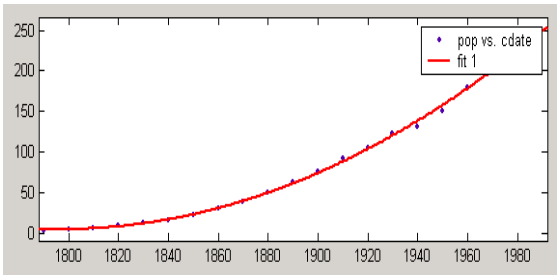
4. Оценку качества подгонки кривых и окончательный выбор аппроксимирующей зависимости производить последовательно:

1. Выбор «хороших» моделей.

1.1.Визуально, по степени согласованности графика полученной в результате подгонки аппроксимирующей кривой с отображенными в этом же окне графического вывода значениями y_i н. Качество подгонки считается удовлетворительным, если значения y_i «вытянуты» вдоль аппроксимирующей кривой. На рисунке 6.3 приведены результаты подгонки, которые можно считать неудовлетворительными (А) и удовлетворительными (Б).



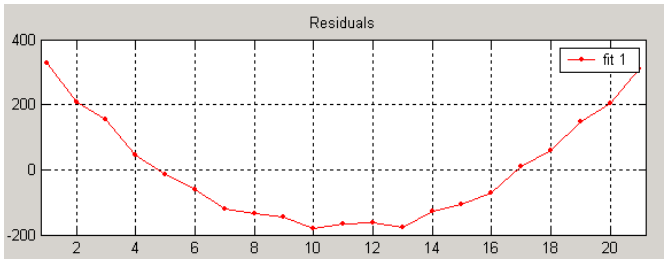
А



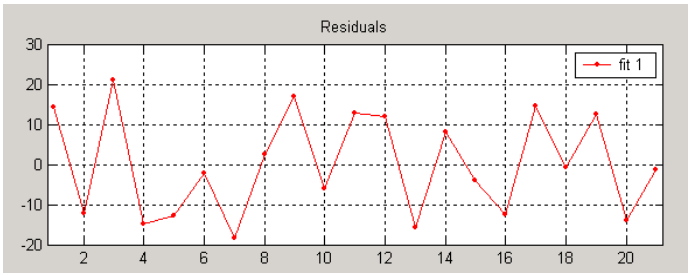
Б

Рисунок 6.3 - Результаты подгонки (неудовлетворительные (А) и удовлетворительные (Б))

1.2. Визуально, по графику разности между значениями y_i и вычисленными по аппроксимирующей функции значениями $f(x_i)$. Качество подгонки считается удовлетворительным, если график разности хорошо аппроксимируется «на глаз» функцией y=0. На рисунке 6.4 приведены результаты подгонки, которые можно считать неудовлетворительными (А) и удовлетворительными (Б).



А



Б

Рисунок 6.4 - Результаты подгонки (неудовлетворительные (А) и удовлетворительные (Б))

2. Выбор лучшей кривой в каждом классе аппроксимирующих зависимостей – по Adjusted R-square. Лучшей кривой в классе является та, которой соответствует максимальное значение Adjusted

R-square. Если максимальное значение соответствует двум или более кривым – выбирается кривая с наиболее простым аналитическим описанием.

3. Окончательный выбор самой лучшей аппроксимирующей зависимости – выбрать самую простую модель из лучших с условием примерно тех же значений диапазона погрешностей экспериментальных данных от аппроксимирующей кривой и RMSE (Root Mean Squared Error).

Задание

1. Освоить технологию работы с программой CurveFitting пакета . MATLAB.
2. Научиться производить обоснованный выбор наилучшей аппроксимирующей зависимости из возможных.
3. В соответствии с порядком выполнения лабораторной работы сделать аппроксимацию экспериментальных данных.

Содержание отчета

1. Постановка задачи.
2. Результаты аппроксимации экспериментальных данных из файла census.mat и censusN.mat и показатели качества подгонки R-square и Ajusted R-square для всех основных классов аппроксимирующих функций (экспоненциальной, гауссиана, показательной, Фурье, полиномиальной, показательной, рациональной) в форме таблицы результатов.
3. Для трех наиболее подходящих аппроксимирующих зависимостей - график аппроксимирующей функции и график погрешности аппроксимации, аналитические выражения для аппроксимирующей зависимости.
4. Обоснование выбора наилучшей аппроксимирующей зависимости.
5. Выводы.

Контрольные вопросы

1. Сколько этапов включает в себя обработка результатов однофакторного эксперимента?
2. Какие виды функций для аппроксимации и интерполяции одномерных массивов данных предоставляет программа подгонки кривых CurveFitting?
3. Приведите формулу показательной функции.
4. Приведите формулу функции Фурье.
5. Приведите формулу рациональной функции.

Лабораторная №6.1
Цифровая фильтрация шумов в среде MATLAB

Цель работы: изучение методики разработки программ цифровой обработки сигналов, включающей различные способы улучшения отношения сигнал/шум (накопление, использование НЧ и ВЧ-фильтров, оптимального фильтра Колмогорова-Винера, прямого и обратного БПФ).

Теоретические сведения

Цифровая обработка сигналов решает задачи обнаружения и определения параметров информативных сигналов и изображений, искаженных шумами и помехами. Для этой цели используются различные средства:

- цифровые частотные фильтры (высокой частоты, низкой частоты, полосовые фильтры);
- накопление (временная фильтрация);
- сглаживание (медианные фильтры);
- оптимальные фильтры (фильтр Колмогорова-Винера, LMS и RLS-фильтры);
- адаптивные фильтры (функцию адаптивных фильтров могут выполнять фильтр Колмогорова-Винера, LMS и RLS-фильтры).
- фильтры для борьбы с шумами при нелинейных и нестационарных процессах (фильтр Гильберта-Хуанга).

Выбор способа борьбы с шумами должен производится с учетом свойств и особенностей информативного сигнала и помехи. Чем в большей степени свойства сигнала и шума априори известны, тем может быть получен больший эффект от цифровой обработки. Кроме того, несмотря на обилие стандартных, доведенных до уровня готовых программ цифровой обработки, с учетом конкретных априори известных свойствах информативного сигнала и шума может оказаться полезным разработка новых методов и алгоритмов для борьбы с шумами.

Цифровой низкочастотный фильтр

Для фильтрации высокочастотного шума может быть применен фильтр низких частот (ФНЧ). Частотная характеристика ФНЧ выражается как

$$K(\omega) = \frac{1}{1 + j\omega},$$

где ω - частота среза НЧ-фильтра.

Для фильтрации сигнала нужно вычислить частотный спектр сигнала с помощью преобразования Фурье, затем перемножить частотный спектр сигнала и частотную функцию фильтра и выполнить обратное преобразование Фурье. Второй способ – вычислить импульсную переходную характеристику фильтра (реакцию на единичный импульс), затем выполнить операцию свертки входного сигнала с импульсной переходной характеристикой фильтра.

В программах обработки дискретизированных сигналов, представленных в форме числовых массивов, цикл

```
for k=1:NFFT
x(k) = A*sin(2*pi*KP1*k/NFFT);
end
```

создает KP1 периодов дискретизированного синусоидального сигнала в числовом массиве, содержащем NFFT значений, понятия частоты в этом случае отсутствует и появляется только в том случае, если задать шаг дискретности по времени. Аналогично этому, в результате выполнения быстрого преобразования Фурье

```
I=1:NFFT;
XX1=fft(x,NFFT);
```

мы получаем числовой массив, который может содержать NFFT элементов, причем информативной будет только первая половина массива, вторая будет зеркально отображать первую половину. В

первой половине массива, содержащей NFFT/2 элементов будет представлен «частотный» спектр. Спектр будет отражать не частоту сигнала, установить которую по числовому массиву, представляющему сигнал во временной области, невозможно, а количество периодов. Т.е. в приведенном выше примере «всплеск» в массиве частотной области будет в элементе с номером KP1. Таким образом, массив частотной области укажет на количество периодов сигнала во временной области.

Частотная характеристика линейного фильтра низких частот может быть вычислена следующим образом:

```
for i=1:NFFT
H(i)=1/((1+j*i/NC));
end
```

Здесь NC - полоса пропускания фильтра по уровню 0,7 амплитуды выражена в количестве отчетов спектра БПФ, пропускаемых фильтром. Остальные отсчеты в массиве частотного спектра будут ослабляться по амплитуде. Таким образом, понятие постоянной времени фильтра, равно как и полосы пропускания при дискретизированном представлении линейного фильтра отсутствует.

Ниже приведена программа фильтрации сигналов и временные диаграммы.

```
%Низкочастотный фильтр
A=20; %амплитуда сигнала
Q=5; %амплитуда шума
KP1=12;% - количество периодов первого сигнала
NFFT=1024;%количество точек расчета
Fs=1024;
for k=1:NFFT % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/NFFT);
q(k)=Q*(randn(size(Fs))); %шум
x(k)=s(k)+q(k); % суммирование сигнала и шума
end
figure
```

```
plot(x);
title('Зашумленный сигнал до фильтра');
Y=fft(x,NFFT)/NFFT; %БПФ сигнала с шумом
SS1=Y.*conj(Y)/Fs; %спектр мощности сигнала с шумом
figure
plot(f,2*abs(SS1(1:NFFT/2)));
title('Частотный спектр сигнала с шумом');
NC=12; %NC - полоса пропускания фильтра по уровню 0,7 амплитуды
% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром
% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде
for i=1:NFFT
H(i)=1/((1+j*i/NC)); %передаточная функция простого фильтра
%в частотной области
End
i=1:400;
figure
%plot(i,abs(H(1:400))); %вывод графика частотной характеристики фильтра
semilogx(i,abs(H(1:400)));%то же, что и plot, но в логарифмическом
%масштабе по X
title('Частотная хар-ка НЧ-фильтра');
i=1:NFFT;
XX1=fft(x,NFFT); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);   %свертка зашумленного сигнала с частотной хар-кой фильтра
i=1:NFFT;
figure
plot(i,2*Z(1:NFFT)); %вывод отфильтрованного сигнала
title('Сигнал после фильтра');
pause;
close all;
```

На рисунке 6.1.1 представлен результат работы программы.

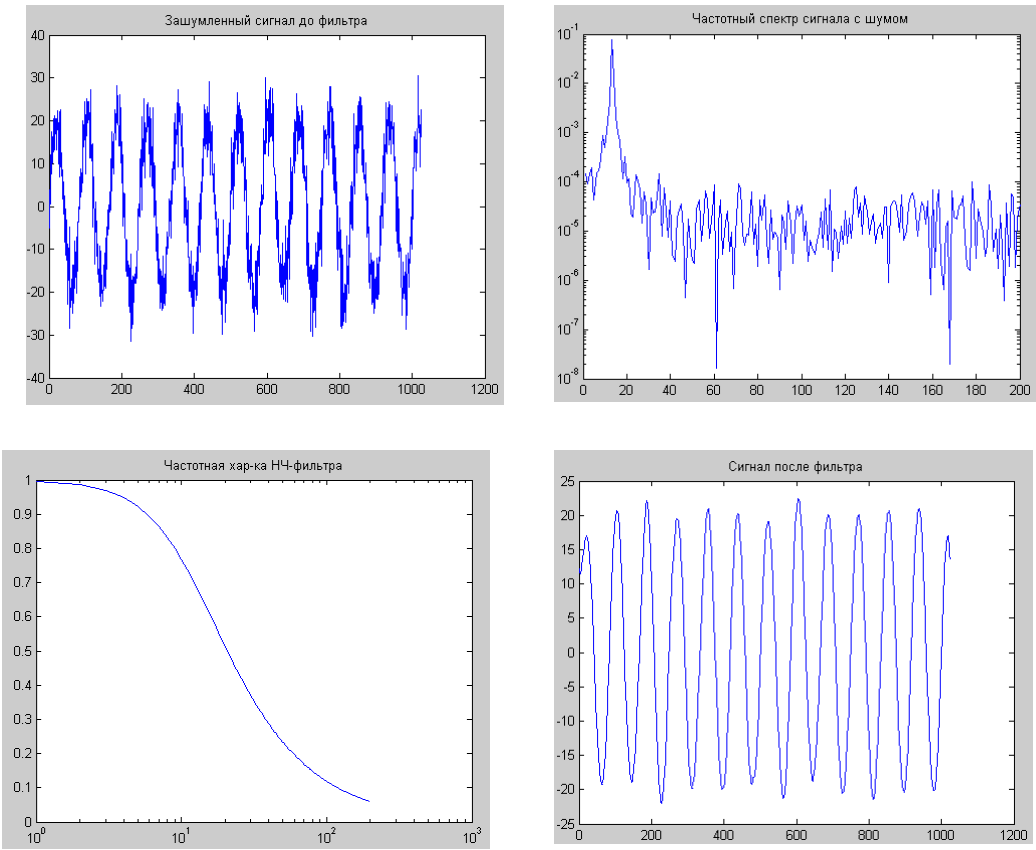


Рисунок 6.1.1 - Исходный сигнал с шумом (А), его частотный спектр (Б), частотная характеристика фильтра в полулогарифмическом масштабе (В) и сигнал после фильтра (Г)

Оптимальный фильтр Колмогорова-Винера

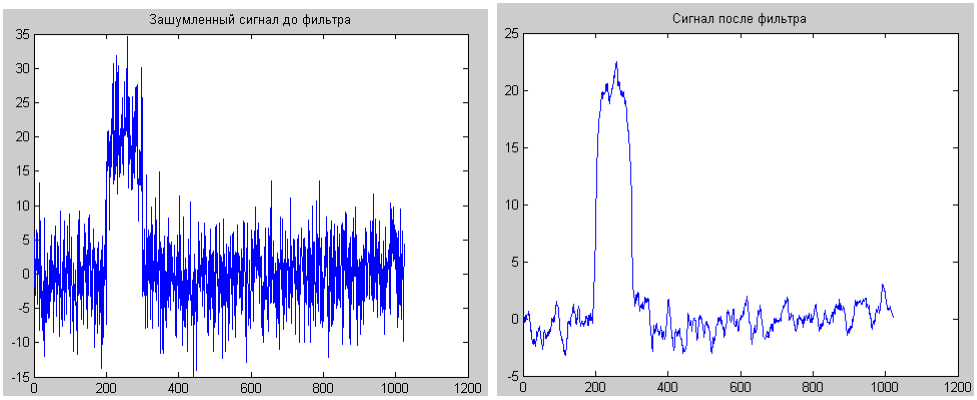
Фильтры низкой частоты, высокой частоты и полосовые фильтры эффективны в том случае, когда частотные спектры сигнала и шума не перекрываются.

Наилучшее разделение сигнала и шума цифровыми методами обеспечивает оптимальный фильтр Колмогорова-Винера.

Частотная характеристика фильтра Колмогорова-Винера:

$$H(w) = Ws(w) / [Ws(w)+Wq(w)],$$

где Ws(w) и Wq(w) - энергетические спектры (плотности мощности) сигнала и помех.



А Б

Рисунок 6.1.2 - Исходный сигнал с шумом (А) и после фильтра (Б)

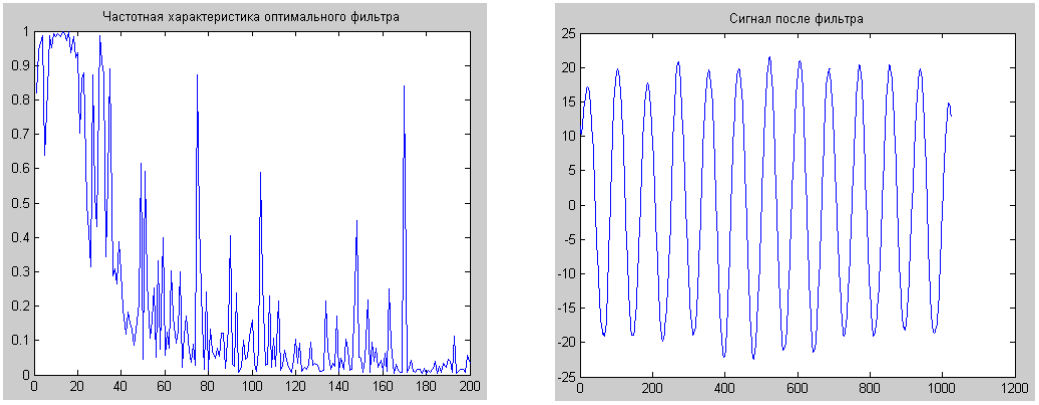


Рисунок 6.1.3 - Частотная характеристика оптимального фильтра (А) и сигнал после фильтра (Б)

Программа, реализующая оптимальный фильтр Колмогорова – Винера в среде MATLAB, приведена ниже.

```
%Программа оптимального фильтра Колмогорова - Винера
A=20; %амплитуда сигнала
Q=20; %амплитуда шума
Fs=1024;
NFFT=1024;%количество точек расчета
L=1024;
for k=1:NFFT %цикл вычисления сигнала и шума
%s1(k)=A*exp(-0.00003*(k-500)^2.0);        %колоколообразный сигнал
%s1(k)=A*sin(2*pi*12*k/1000.0)%синусоидальный сигнал
s1(k)=0;
```

```
if (k>400)&(k<600) % сигнал прямоугольной формы
  s1(k)=A;
end
q(k)=Q*(randn(size(Fs))); %шум
x1(k)=s1(k)+q(k); % суммирование сигнала и шума
end
figure
plot(x1);
title('Зашумленный сигнал до фильтра');

Y=fft(s1,NFFT)/L; %БПФ сигнала без шума
f = Fs/2*linspace(0,1,NFFT/2);%вычисление шкалы для частотной области
% figure
% plot(f,2*abs(Y(1:NFFT/2)));
SS1=Y.*conj(Y)/Fs; %спектр мощности сигнала без шума
% figure
% plot(f,2*abs(SS1(1:NFFT/2)));

Y1=fft(q,NFFT)/L; %БПФ  шума
f = Fs/2*linspace(0,1,NFFT/2);
% figure
% plot(f,2*abs(Y1(1:NFFT/2)))
SS2=Y1.*conj(Y1)/Fs; %спектр мощности  шума
% figure
% plot(f,2*abs(SS2(1:NFFT/2)));
for i=1:NFFT
H(i)=SS1(i)/(SS1(i)+SS2(i)); %передаточная функция оптимального фильтра
%в частотной области
end
% i=1:40;
% figure
% plot(i,abs(H(1:40)));
i=1:NFFT;
h=ifft(H);
XX2=conv(x1,h);
```

```
figure
plot(i,XX2 (1:NFFT)); %вывод отфильтрованного сигнала
title('Сигнал после фильтра 2');
i=1:NFFT;
XX1=fft(x1,NFFT); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);   %свертка зашумленного сигнала с частотной характеристикой фильтра
figure
plot(i,Z (1:NFFT)); %вывод отфильтрованного сигнала
title('Сигнал после фильтра');
```

На рисунках 6.1.2, 6.1.3 представлен результат работы программы.

Фильтрация шумов с использованием прямого и обратного БПФ

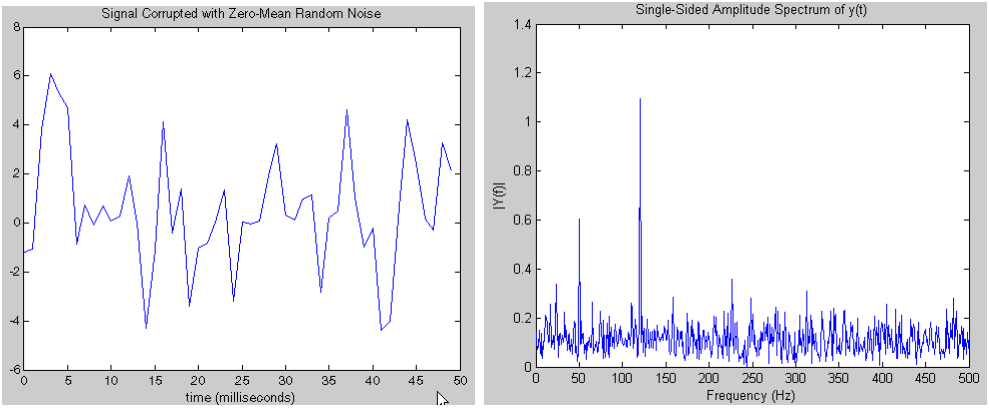
В том случае, когда частотные спектры сигнала и шума не перекрываются, фильтрация шума может быть произведена путем выполнения БПФ, обнуления спектральных линий шума и последующего обратного БПФ.

```
%Программа БПФ. Сигнал содержит две частотные составляющие.
Fs = 1000;                  % Sampling frequency
T = 1/Fs;                   % Sample time
L = 1000;                   % Length of signal
t = (0:L-1)*T;              % Time vector
% Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
x = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 2*randn(size(t));     % Sinusoids plus noise
figure
plot(Fs*t(1:50),y(1:50))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
%NFFT = 2^nextpow2(L); % Next power of 2 from length of y
NFFT=1024;
```

```
Y = fft(y,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2);

% Plot single-sided amplitude spectrum.
figure
plot(f,2*abs(Y(1:NFFT/2)))
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
```

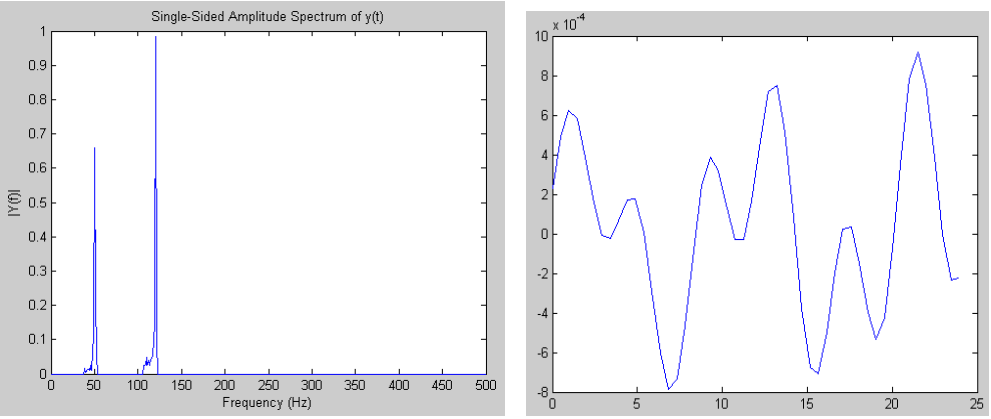
На рисунке 6.1.4 представлен зашумленный сигнал, представляющий сумму двух синусоидальных сигналов разных частот (количество периодов во временном окне 50 и 120) и частотный спектр, полученный с помощью БПФ.



А Б

Рисунок 6.1.4 - Зашумленный сигнал, представляющий сумму двух синусоидальных сигналов разных частот (количество периодов во временном окне 50 и 120) (А) и частотный спектр, полученный с помощью БПФ (Б)

Если теперь обнулить участки спектра от 0 до 40, от 55 до 110 и от 125 до 500, а затем выполнить обратное преобразование БПФ (ifft), то получим спектр и сигнал, представленные на рисунке 6.1.5.



А Б

Рисунок 6.1.5 - Частотный спектр после удаления спектральных составляющих шума (А) и восстановленный с помощью обратного преобразования Фурье сигнал (Б)

Текст программы фильтрации помех с помощью прямого и обратного преобразования Фурье приведен ниже.

```
%Программа фильтрации помех с использованием БПФ-ОБПФ
Fs = 1000;                  % Sampling frequency
T = 1/Fs;                   % Sample time
L = 1000;                   % Length of signal
t = (0:L-1)*T;              % Time vector
% Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
x = 1*sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 0.2*randn(size(t));     % Синусоиды+шум
figure
plot(Fs*t(1:50),y(1:50))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
%NFFT = 2^nextpow2(L); % Next power of 2 from length of y
NFFT=1024;
Y = fft(y,NFFT)/L;%вычисление спектра сигнала с шумом
for i=1:NFFT %"вырезание" спектральных составляющих шума
if (i>125)|(i<40)|((i>55)&(i<110))
   Y(i)=0;
end
end
```

```
f = Fs/2*linspace(0,1,NFFT/2);
figure
plot(f,2*abs(Y(1:NFFT/2)))
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')

%восстановление исходного сигнала после "вырезания"
%спектральных составляющих шума
z=ifft(Y);
f = Fs/2*linspace(0,1,NFFT);
figure
plot(f(1:50),z(1:50));

pause;
clear all; %стирание всех окон графического вывода
```

Фильтр скользящего среднего

Пусть мы имеем массив N значений измеренного сигнала, представленный в цифровой форме:

$$\{f_1, f_2, ..., f_N\},\ i = 1,2,3,...,N.$$

Для нахождения скользящего среднего в окрестности точки f_i берем среднее арифметическое от K предыдущих и K последующих точек, включая и f_i. Таким же образом производим обработку для всех значений i. В результате вычисляем новый массив g_i:

$$g_i = \frac{1}{2K+1}(f_{1-K} + f_{1-K-1} + ... + f_i + ... + f_{i-K})$$

или

$$g_i = \frac{1}{2K+1}\sum_{j=-K}^{K} f_{i+j}.$$

Текст программы фильтрации помех с помощью фильтра скользящего среднего приведен ниже.

```
%Фильтр скользящего среднего

A=20; %амплитуда сигнала
Q=1; %СКО шума
KP1=6;% - количество периодов первого сигнала
KP2=5;% - количество периодов второго сигнала
N=256;%количество точек расчета

for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/1000);
q(k)=Q*(randn(size(N))); %СКО шума, амплитуда равна 3Q
x(k)=s(k)+q(k); % суммирование сигнала и шума
end
figure
plot(x);
hold on;
title('Зашумленный сигнал до фильтра');
S=x(1)+x(2)+x(3)+x(4)+x(5)+x(6)+x(7);
y(4)=S/7;
for i=1:N-7 %сглаживание зашумленного сигнала
   S=S-x(i)+x(i+7);
   y(i+4)=S/7;
end
i=1:N-7;
%figure;
plot(i,y(1:N-7),'r-');
title('Сигнал после фильтра');
xlabel('Номер отсчета'); % подпись по оси X
i=1:N-7;
DX(i)=x(i)-y(i);
figure
plot(i,DX(i:N-7)); %вывод погрешности отфильтрованного сигнала
title('Погрешность отфильтрованного сигнала');
```

```
ylabel('Случайная погрешность'); % подпись по оси Y
pause;
clear all; %стирание всех окон графического вывода
```

Порядок выполнения работы

1. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря накоплению построить зависимость отношения сигнал/шум и разности исходного и обработанного сигналов от количества накоплений.
2. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря использованию фильтров (НЧ, Баттерворта, БПФ-ОБПФ) найти оптимальную полосу пропускания фильтра, при которой эффект будет наибольшим с точки зрения улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала отдельно.
3. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря использованию фильтра скользящего среднего (Smoothing) определить зависимость степени подавления шумов от ширины окна (при рамере окна 3, 5, 7 элементов) при различном уровне шумов и влияние параметров фильтра на изменения амплитуды и фазы сигнала на выходе фильтра.
4. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала произвести сравнение эффективности исследуемых цифровых фильтров с оптимальным фильтром Колмогорова-Винера при обработке сигналов различной формы: гармонических, импульсных, колоколообразных при различных параметрах частоты и длительности.

Задание

Имеется набор экспериментальных данных в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую различные способы улучшения отношения сигнал/шум:

1. Накопление.
2. НЧ-фильтр.
3. Фильтр скользящего среднего.
4. Фильтр Баттерворта.
5. Прямое и обратное БПФ.
6. Оптимальный фильтр Колмогорова-Винера.

и оценить сравнительный эффект улучшения отношения сигнал/шум и степень искажения сигнала в результате обработки.

Содержание отчета

1. Цель работы.
2. Задание к работе.
3. Таблица исходных данных.
4. Тексты программ фильтрации с комментариями.
5. Графики исходного и обработанного сигналов, график погрешности при различных параметрах обработки.
6. Выводы.

Контрольные вопросы

1. Какие задачи решает цифровая обработка сигналов?
2. Какие средства используются для обнаружения и определения параметров информативных сигналов и изображений, искаженных шумами и помехами?
3. Как выражается частотная характеристика ФНЧ?
4. Приведите алгоритм фильтрации сигнала.
5. В каких случаях эффективны ФНЧ, ФВЧ и полосовые фильтры?
6. Какой фильтр обеспечивает наилучшее разделение сигнала и шума цифровыми методами?

7. Как реализуется частотная характеристика фильтра Колмогорова-Винера?
8. Приведите алгоритм фильтрации шумов с использованием прямого и обратного БПФ?
9. Что представляет собой фильтр скользящего среднего?
10. Как уменьшается погрешность обработанного сигнала?

Лабораторная работа №6.2
Цифровая обработка сигналов в среде MATLAB

Цель работы: изучение методики разработки программ сложных видов цифровой обработки сигналов, включающей комбинацию ключевых операций (БПФ, корреляции, сплайн-аппроксимации и передискретизации).

Теоретические сведения

Одной из важнейших задач цифровой обработки зашумленных сигналов является обнаружение информативного сигнала в потоке данных, искаженных шумами и помехами, и определение его параметров. Для этого применяются различные методы, такие как временная фильтрация (накопление), оптимальная частотная фильтрация, прямое и обратное преобразование Фурье, корреляционная обработка. Каждая из этих операций позволяет выполнять преобразования исходного сигнала, например, переход сигнала из временной области в частотную или наоборот, причем при этом производится уменьшение уровня шумов в обработанном сигнале. В задачах обнаружения и определения параметров защумленных сигналов усиление эффекта подавления шумов и увеличения точности определения параметров сигнала можно достичь, используя несколько методов цифровой обработки в комплексе. Примером может быть задача обработки эхо-сигнала спектрометра ЯМР.

Результатом БПФ дискретизированного эхо-сигнала спектрометра ЯМР определенной частоты является количество периодов сигнала во временном окне. Если частота отсчетов или интервал дискретности по времени при измерении сигнала известен, то по количеству периодов во временном окне можно установить и частоту измеряемого сигнала. Точность определения частоты в спектре входного сигнала вполне определенна и зависит от количества периодов p сигнала. Если количество периодов целое, то частота с помощью БПФ находится абсолютно точно (при отсутствии зашумленности сигнала). Если же количество периодов не является целым, то появляется погрешность определения

частоты. Максимальное значение погрешности равняется 1/p. В некоторых, практически важных случаях, например, при обработке эхо-сигналов (рисунок 6.2.1 А) импульсных спектрометров ЯМР, количество периодов анализируемого сигнала во временном окне принципиально ограничено величиной около 10. В этом случае погрешность определения частоты с помощью БПФ достигает 1/10, т. е. 10%.

Влияние шума в регистрируемом сигнале во временной области можно значительно уменьшать за счет многократного повторения эксперимента и синхронного накопления эхо-сигналов (рисунок 6.2.1 Б).

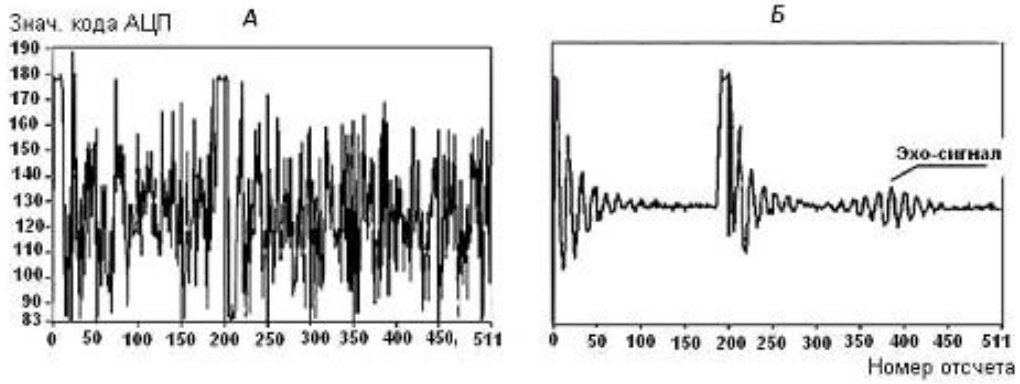


Рисунок 6.2.1 - Исходный (А) и накопленный сигнал (Б) в борате железа

Однако увеличение количества накоплений позволяет улучшать отношение сигнал/шум без искажения формы и уменьшения амплитуды накопленного эхо-сигнала лишь до некоторого предела. После этого предела накопление уже не приносит ощутимого улучшения качества. В частности, при большом времени накопления на эхо-сигнал начинают влиять постепенные изменения параметров приборов, входящих в состав импульсного спектрометра ЯМР. При ограничении времени проведения анализа веществ количество возможных накоплений сигнала должно быть ограничено или вообще должно отсутствовать. Проблема обработки эхо-сигналов в условиях значительных шумов возникает еще и в том случае, когда имеется необходимость исследования веществ малой концентрации.

Поэтому задача выполнения частотного анализа зашумленных эхо-сигналов актуальна.

Непосредственное использование БПФ для зашумленного сигнала не позволяет получить точное значение количества периодов во временном окне и частоты измеряемого сигнала в том случае, когда количество периодов не является целым, когда анализируемый сигнал занимает только часть временной области, модулирован по амплитуде и зашумлен. В практически важных случаях частотного анализа сигналов спектрометров ЯМР форма и начальная фаза эхо-сигналов известны. Это дает возможность создать эталонные сигналы, соответствующие ожидаемому эхо-сигналу по форме и начальной фазе, и произвести их корреляционное сравнение. Коэффициент корреляции эхо-сигнала с эталонным сигналом будет равен единице, если частоты эхо-сигнала и эталонного сигнала равны и эхо-сигнал не зашумлен. Поэтому при отсутствии шумов найти частоту эхо-сигнала можно производя корреляционное сравнение с эталонными сигналами, частоту эталонных сигналов подбирать до выполнения условия, когда коэффициент корреляции будет равен единице. Однако коэффициент корреляции уменьшается как при разнице частот эхо и эталонного сигналов, так и при совпадении частот, но из-за наличия шума. Поэтому таким способом определить частоту зашумленного сигнала невозможно.

Усовершенствование алгоритма определения частоты сигнала достигается за счет сочетания положительных качеств корреляционного подхода и БПФ с целью повышения точности определения частоты дискретизированного сигнала в условиях, когда обрабатываемый сигнал во временной области зашумлен, количество периодов не является целым, сигнал модулирован по амплитуде, количество периодов сигнала мало.

Идея предлагаемого алгоритма заключается в том, что в небольшой окрестности от предполагаемой частоты сигнала или количества периодов (приближенное значение частоты сигнала может быть найдено с помощью быстрого преобразования Фурье) вычисляются коэффициенты корреляции с несколькими эталонными сигналами в некоторой окрестности от приближенного значения, затем с помощью сплайн-интерполяции и

передискретизации строится функция, выражающая зависимость коэффициента корреляции от частоты эталонов и находится максимум этой функции, по положению максимума определяется частота эталонного сигнала. Функция, построенная таким образом, имеет вид параболы с явно выраженным максимумом (рисунок 6.2.2) как в случае незашумленного так и зашумленного сигнала, что и позволяет определить частоту эхо-сигнала более точно, чем это позволяет сделать БПФ. При наличии шума форма функции сохраняется, уменьшается лишь абсолютное значение максимума.

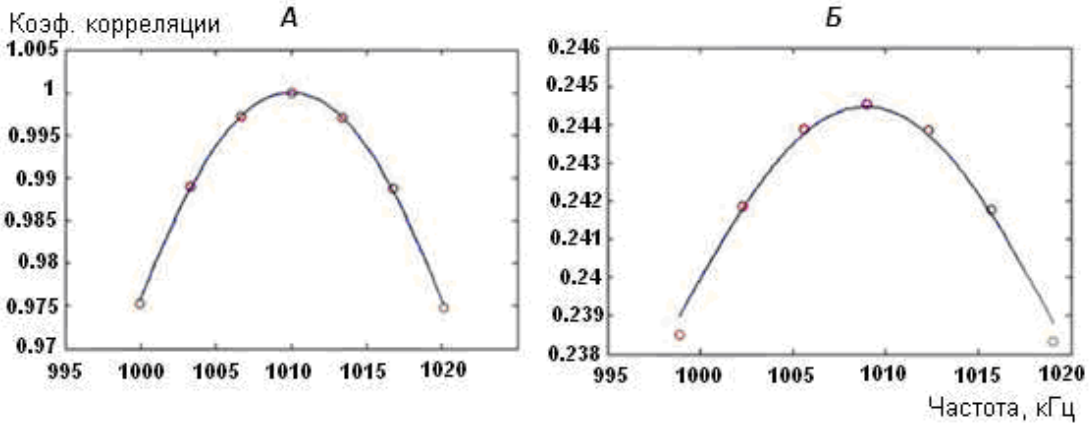


Рисунок 6.2.2 - Зависимость коэффициента корреляции от частоты при отсутствии шума (А) и при отношении сигнал/шум 1/3 (Б)

Точное значение частоты равно 1010. Аппроксимирующие кривые построены с помощью функции сплайн-аппроксимации spaps в MATLAB.

Точность определения частоты сигнала тем выше, чем ближе начальное приближение к истинной частоте. Поэтому применяется итерационное вычисление, на каждом этапе итерации в качестве начального приближения используется уточненное значение частоты, полученное на предыдущем этапе. В качестве первого приближения берется частота, определенная с помощью БПФ.

Количество итераций для расчета частоты с заданной точностью с помощью описанного алгоритма зависит от того, насколько близко к искомой частоте будет находиться начальное приближение.

Порядок выполнения работы

1. Использовать в качестве основы программу, приведенную в приложении 4.

2. Расчеты сделать по результатам 1-й и 2-й итерации отдельно. В качестве приближенного значения количества периодов на 1-й итерации взять результат, полученный с помощью БПФ, для 2-й итерации – уточненное в результате 1-й итерации значение количества периодов, величину окрестности вблизи уточненного значения количества периодов, в которой создаются эталонные сигналы (параметр procch в программе) уменьшить в m раз, $m \le 10$.

3. При исследовании эффекта увеличения точности определения количества периодов и частоты сигнала по сравнению с БПФ за счет дополнительных операций цифровой обработки произвести измерения количества периодов и частоты сигнала в диапазоне количества периодов от К до К+1 с шагом 0.05 и вычислить максимальную, среднюю и среднеквадратическую погрешности определения количества периодов и частоты сигнала на этом интервале.

4. При исследовании эффекта увеличения точности определения количества периодов и частоты сигнала по сравнению с БПФ за счет дополнительных операций цифровой обработки при зашумленности сигнала произвести измерения количества периодов и частоты сигнала в диапазоне количества периодов от К до К+1 с шагом 0.05 при значениях амплитуды шума от 0 до 1 с шагом 0.1 и вычислить максимальную, среднюю и среднеквадратическую погрешности определения количества периодов и частоты сигнала на этом интервале для каждого уровня шума.

5. При исследовании зависимости погрешности определения количества периодов и частоты сигнала от количества периодов сигнала с помощью БПФ и комбинированного способа выполнить п.2 и 3 при значениях К=2, 3, …10, и построить графики зависимости максимальной, средней и среднеквадратической погрешностей от количества периодов сигнала.

Задание

1. Имеется набор экспериментальных данных в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую точное определение количества периодов сигнала и частоту во временной области с использованием нескольких ключевых операций ЦОС: БПФ, корреляции, сплайн-аппроксимации и передискретизации.
2. Исследовать эффективность использования комбинированнного использования нескольких операции цифровой обработки и итерационного процесса для определения количества периодов и частоты сигнала по сравнению с использованием для решения этой задачи быстрого преобразования Фурье. Исследование провести для различных условий:
   - Для различного количества итераций.
   - При целом и нецелом количестве периодов сигнала, меньшем 10.
   - При различном уровне зашумленности сигнала

Содержание отчета

1. Цель работы.
2. Задание к работе.
3. Текст разработанной программы.
4. Графики зависимости максимальной, средней и среднеквадратической погрешностей при различном уровне шума от количества периодов сигнала для БПФ и исследуемого способа комбинированной обработки.
5. Выводы.

Контрольные вопросы

1. Какие методы применяются для цифровой обработки зашумленных сигналов?
2. Как можно значительно уменьшить влияние шума в регистрируемом сигнале?

3. Перечислите алгоритм определения частоты сигнала.
4. Перечислите алгоритм цифровой обработки сигнала в середе MATLAB.
5. Как осуществляется генерация модельного сигнала в системе MATLAB?

Лабораторная работа №6.3
«Работа с речевыми данными в среде MATLAB»

Цель работы: освоить навыки записи и считывания данных из речевого сигнала в системе MATLAB; исследовать особенности мужского и женского голоса.

Теоретические сведения

Характеристики голоса

Разложим голос на составляющие, основными параметрами являются: частота, сила, длительность и тембр, которые, как величины, можно анализировать и по отдельности. В действительности, однако, подобный анализ не представляет собой реального выражения голоса, поскольку эти качества образуют единый неделимый комплекс.

Высота издаваемого звука зависит от числа колебаний голосовых складок в 1 секунду. Голосовые складки способны приходить в колебательные движения не только целиком, всей своей массой, но и отдельными участками. Только этим можно объяснить то, что одни и те же голосовые складки могут колебаться с различной частотой: примерно от 80 до 10 000 колебаний в секунду и даже больше.

Тоновый диапазон человеческого голоса представлен последовательностью тонов, которые могут быть произведены голосовым аппаратом в пределах границ между самым низким и самым высоким звуками. Человеческий голос обычно включает в себя тоны от 64 до 1300 герц. В двух формах проявления человеческого голоса - пении и разговоре - качества голоса представлены несколько различно. Разговорный голос составляет лишь 110 от общего диапазона голоса. Тоновый охват певческого голоса значительно шире разговорного и зависит от вокального образования. Сила подаваемого звука определяется интенсивностью напряжения голосовых складок и величиной давления воздуха в подсвязочном пространстве. И тот и другой процесс регулируются центральной нервной системой. Контроль

осуществляется с помощью слуха. Если же взаимоотношения между этими процессами нарушаются, например, при крике ужаса, то превалирование давления внутри трахеи вызывает звук, который характеризуется отсутствием чистой тональности.

Ниже представлен диапазон человеческого голоса (в Герцах):

Бас - 75-300.

Баритон - 100-400 - Тенор - 120-500.

Контральто - 170-780.

Меццо-сопрано - 200-900.

Сопрано - 230-1000.

Колоратурное сопрано - 260-1400.

Для сравнения приведен частотный диапазон музыкальных инструментов:

Контрабас - 40-300.

Виолончель - 65-880.

Альт - 130-1240.

Скрипка - 210-2800.

Фагот - 60-630.

Кларнет - 140-1980.

Гобой - 230-1480.

Флейта - 240-2300.

Туба - 45-320.

Тромбон - 80-500.

Валторны - 60-740.

Труба - 160-990.

Таким образом, человеческий голос имеет диапазон звучания от 75 до 110 Герц, который так или иначе перекрывает (заглушается, смешивается) с любым музыкальным инструментом (оптимальная точка - 300 Герц).

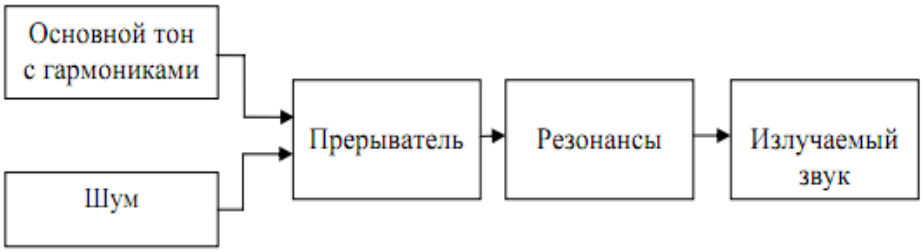
Также необходимо учитывать силу звучания (динамический диапазон) данных инструментов.

Динамический диапазон гитары составляет 15 дБ; органа - 35 дБ; рояля - 45 дБ; женский голос - 20-35 дБ; мужской голос - 20-45 дБ, эстрадного оркестра-45-55 дБ, симфонический оркестр 60-75 дБ.

Сообщение, передаваемое с помощью речевого сигнала, - дискретно, т.е. может быть представлено в виде

последовательности символов из конечного их числа. Звуковые символы, из которых составлен речевой сигнал, называются фонемами.

Речь с физической точки зрения состоит из последовательности звуков с паузами между их группами. Схема речеобразования у человека представлена на рисунке 6.3.1.



Рисунка 6.3.1 - Схема речеобразования у человека

При нормальном темпе речи паузы появляются между отрывками фраз. Как правило, слова произносятся слитно, хотя слушающий воспринимает слова по отдельности. При замедленном темпе речи, например, при диктовке, паузы могут делаться между словами и даже их частями. Предлоги, союзы звучат всегда слитно с последующим словом.

Частотный диапазон речи находится в пределах 70 - 1400 Гц.

Один и гот же звук речи разные люди произносят по-разному. Произношение звуков речи зависит от ударения, соседних звуков и т.п. Но при всем многообразии в их произношении звуки являются физическими реализациями (произнесением) ограниченного числа обобщенных звуков речи (фонем). Фонема - это то, что человек должен произнести, а звук-то, что человек фактически произносит. Фонема по отношению к звуку речи играет ту же роль, что и образцовая буква по отношению к ее рукописной форме в конкретном написании.

Звуки речи делятся на звонкие и глухие. Звонкие звуки образуются с участием голосовых связок, в этом случае находящихся в напряженном состоянии. Под напором воздуха, идущего из легких, они периодически раздвигаются, в результате чего создается прерывистый поток воздуха. Импульсы потока воздуха, создаваемые голосовыми связками, с достаточной

точностью могут считаться периодическими. Соответствующий период повторения импульсов называют периодом основного тона голоса T_0 - а обратную величину $1/T_0$ - частотой основного тона. Если связки тонкие и сильно напряжены, то период получается коротким и частота основного тона высокой; для толстых, слабо напряженных связок - низкой. Частота основного тона для всех голосов лежит в пределах 70 - 450 Гц. При произнесении речи она непрерывно изменяется в соответствии с ударением, подчеркиванием звуков и слов, а также с проявлением эмоций (вопрос, восклицание, удивление и т.д.). Изменение частоты основного тона называется интонацией. У каждого человека свой диапазон изменения основного тона (обычно он бывает немногим более октавы) и своя интонация. Последняя имеет большое значение для узнавания говорящего.

Основной тон, интонация, устный почерк и тембр голоса служат для опознавания человека, и степень достоверности такая же высокая, как по отпечаткам пальцев. Импульсы основного тона имеют пилообразную форму, и поэтому при их периодическом повторении получается дискретный спектр с большим числом гармоник (до 40), частоты которых кратны частоте основного тона. Огибающая спектра основного тона имеет спад в сторону высоких частот с крутизной около 6 дБ/окт, поэтому для мужского голоса уровень составляющих на частоте 3000 Гц ниже их уровня на частоте 100 Гц примерно на 30 дБ. При произнесении глухих звуков связки находятся в расслабленном состоянии, поток воздуха из легких свободно проходит в полость рта. Встречая на своем пути различные преграды в виде языка, зубов, губ, он образует завихрения, создающие шум со сплошным спектром.

Согласные по способу образования делятся на сонорные (л, ль, р, рь. м, мь, и, нь, й), щелевые (ж, з, зь, в, вь, ш, с, сь, ф, фь, х, хь), взрывные (б, бь, д, дь, г, гь, и, иь, т, ть, к, кь) и аффрикаты (ц, ч - комбинация глухих взрывных и щелевых). Гласных фонем всего шесть: а, о, у, э, и, ы (гласные е, я, ё, ю - составные из и краткого или мягкого знака и гласных э, а, о, у).

Звонкие звуки речи, особенно гласные, имеют высокий уровень интенсивности, глухие - самый низкий. Громкость речи непрерывно изменяется, особенно резко при произнесении

взрывных звуков. Динамический диапазон уровней речи находится в пределах 35 - 45 дБ. Гласные звуки речи имеют в среднем длительность около 0,15 с, согласные - около 0,08 (звук и - около 30 мс).

Звуки речи неодинаково информативны. Так, гласные звуки содержат малую информацию о смысле речи, а глухие согласные наиболее информативны (например, в слове «посылка» последовательность «о, ы, а» ничего не говорит, а «п, с, лк» дает почти однозначный ответ о смысле). Поэтому разборчивость речи снижается при действии шумов, в первую очередь из-за маскировки глухих звуков.

Известно, что для передачи одного и того же сообщения по телеграфу и по речевому тракту требуется различная пропускная способность. Для телеграфного сообщения достаточна пропускная способность не более 100 бит с, а для речевого - около 100000 бит с (полоса равна 7000 Гц, динамический диапазон 42 дБ, т.е. требуется семизначный код, откуда имеем 2*7000 -7=98 000 бит/с), т.е. в 100 раз большая.

Образование звуков речи происходит путем подачи команд к мускулам артикуляционных органов речи от речевого центра мозга. Общий поток сообщений от него составляет в среднем не более 100 бит/с. Вся остальная информация в речевом сигнале называется сопутствующей.

Запись и считывание данных из речевого сигнала в MATLAB

В MATLAB предусмотрены средства для воспроизведения и записи звука (речи), а также для работы со звуковыми файлами формата wav.

Чтение wav-файлов. Для считывания wav-файлов в MATLAB используется функция wavread. В простейшем случае она может быть использована следующим образом:

y=wavread ('filename'), где 'filename' - имя звукового файла (расширение wav указывать не обязательно). В имя файла необходимо включить полный путь, за исключением тех случаев, когда файл находится в текущем (для MATLAB) каталоге или в одном из каталогов, входящих в список поиска MATLAB. Другой

способ, не требующий указания имени файла, - полный путь, который заключается в определении местонахождения файла на жестком диске с помощью меню MATLAB.

В результате вызова функции в переменную y будет помещено все содержимое указанного файла. Строки матрицы y соответствуют отсчетам сигнала, столбцы - каналам, которых в wav-файле может быть один (моно - канал) или два (стереоканал).

Помимо отсчетов сигнала в wav-файлах хранится и служебная информация, которая содержит следующие параметры:

- частоту дискретизации, для определения которой в указанную функцию необходимо включить второй выходной параметр:

[y, Fs] = wavread ('filename'),

где Fs - частота дискретизации, Гц;

- число бит на отсчет, для определения которого необходимо добавить еще один выходной параметр:

[y, Fs, bils] = wavread ('filename');

- число отсчетов и каналов записи. Для получения данной информации необходимо вызвать функцию wavread с двумя входными параметрами: именем файла и текстовой строкой 'size': wavesize=wavread ('filename*, 'size').

При вызове такой функции из wav-файла извлекается служебная информация, которая возвращается в виде двухэлементного вектор-строки, первый элемент которого содержит число отсчетов, второй - число каналов;

- продолжительность звучания сигнала (в секундах), которую можно определить следующим образом:

wavesize (1) Fs, где 1 указывает на первый параметр вектора wavesize.

Имеются и возможности считывания данных из wav-файла не целиком, а отдельными фрагментами. Для этого используется второй входной параметр функции wavread. Если этот параметр

является числом, будет считано соответствующее количество отсчетов, начиная с первого:

y = wavread ('filename', N).

Если нужный фрагмент расположен не в начале файла, придется указать его начало и конец:

y = wavread ('filename', [nl, n2]).

В результате в переменную y будут считаны отсчеты с номерами от n_1 до n_2 включительно (нумерация отсчетов начинается с единицы).

Чтобы узнать объем памяти (в килобайтах), требуемый в MATLAB для хранения записи, необходимо использовать следующую функцию:

prod (wavesize)*8/1024.

Для просмотра речевого (звукового) сигнала выведем его в виде графика с помощью следующей функции: plot (y).

Если необходимо вывести график по каналам стереозаписи, то применяют следующие функции:

subplot (2, 1, 1); plot (:, 1); subplot (2, 1, 2); plot (:, 2) или просто plot(y).

Если сигнал имеет большую длину, то можно использовать следующую функцию (фрагменты выводятся друг под другом):

strips (x, N),

где x - вектор отсчетов сигнала (двумерный массив не допускается), N - число отсчетов в каждом фрагменте (этот параметр можно опустить, по умолчанию размер фрагмента составляет 200 отсчетов).

Запись wav-файлов. Для записи вектора (или матрицы) на диск в виде wav-файла используется функция wavwrite'.

wavwrite (y, Fs, N, 'filename'), где y - записываемые данные, Fs - частота дискретизации, Гц, N - число бит на отсчет (8 или 16), 'filename' - имя создаваемого файла. Параметры Fs и N можно опускать, при этом используются значения по умолчанию: Fs = 8 000 Гц, N= 16.

Записываемые данные должны быть вещественными и лежать в диапазоне от -1 до 1. Значения, выходящие из этого диапазона, будут обрезаны и сделаны равными.

Воспроизведение звуковых файлов. Помимо работы с wav-файлами можно воспроизводить вектор и матрицу в звуковом в виде с использованием следующих функций:

sound, синтаксис которой записывается следующим образом:

sound (y, Fs, bits),

где y - вектор или двухстолбцовая матрица сигнала, Fs - частота дискретизации, Гц, bits - число бит на отсчет (8 или 16).

Параметры Fs и bits можно опускать, при этом их значения будут приниматься по умолчанию.

Выходных параметров у y функции нет. После вызова она передает вектор y звуковой карте для воспроизведения и сразу же, не дожидаясь окончания звука, возвращает управление MATLAB;

wavplay, синтаксис которой имеет следующий вид:

wavplay (y, Fs, 'mode'), где параметр mode управляет режимом воспроизведения, который может принимать два значения:

'sync' - синхронный режим, означающий что функция вернет управление интерпретатору MATLAB только после окончание звука;

'async' - асинхронный режим, при котором функция передает данные для воспроизведения звуковым драйверам Windows и сразу же возвращает управление системе MATLAB. не дожидаясь окончания звука.

Параметры Fs и mode можно опускать, при этом их значения принимаются по умолчанию: Fs =11025 Гц и 'mode' = 'async'.

Запись звука (речи). Функция wavrecord позволяет записать звук в переменную MATLAB с помощью звуковой карты компьютера:

y = wavrecord (n, Fs, ch, 'dtype'),

где n - число записываемых отсчетов, Fs - частота дискретизации, Гц. ch - число каналов записи, 'dtype' - тип записываемых данных.

Возвращаемый результату - матрица, каждый столбец которой соответствует одному каналу записи. При стереозаписи первый столбец - левый канал, второй - правый канал.

Для параметра dtype возможны следующие значения:

'double’ - 16-битная запись, данные масштабируются к диапазону от -1 до 1 и представляются в восьмибайтовом формате с плавающей запятой;

'single' - 16-битная запись, данные масштабируются к диапазону -1…1 и представляются в четырехбайтовом формате с плавающей запятой;

‘untl6’ - 16-битная запись, данные представляются в двухбайтовом целочисленном формате (диапазон от-32 768 до 32 767);

'unint8' - 8-битная запись, данные представляются в однобайтовом беззнаковом целочисленном формате (диапазон от 0 до 255, нулевому напряжению на входе соответствует значение «128»).

Входные параметры Fs, ch, dtype можно опускать, при этом их значения будут приниматься но умолчанию: Fs =11 025 Гц, ch =1, dtype='double’.

Спектрограмма

Если спектр сигнала меняется во времени, то для оценки спектра целесообразно использовать спектрограмму сигнала. Спектрограммой (spectrogram) сигнала называется его мгновенный спектр, зависящий от времени. Для вычисления спектрограммы вектор сигнала разбивается на сегменты (в общем случае с перекрытием). Для каждого сегмента вычисляется спектр с

помощью функции fft. Набор спектров всех сегментов и образует спектрограмму. Для вычисления спектрограммы служит функция spectrogram.

Синтаксис вызова функции:

[S, F, T]=spectrogram (x, window, nover - lap, nfft, Fs),

где x - вектор сигнала; window - вектор весовой функции (если вместо вектора используется целое число, то используется весовая функция по умолчанию - функция Хэмминга соответствующей длины); noverlap - величина перекрытия соседних сегментов сигнала; nfft - число точек преобразования Фурье; Fs - частота дискретизации. S - матрица, каждая колонка которой содержит (nfft/2+l) отсчетов спектра для данного момента времени (если nfft - нечетное число, количество отсчетов равно (nfft+l)/2). Число колонок k=nx((nx-noverlap)/(length(window) - noverlap)), где nx - длина вектора сигнала. Параметр F - вектор частот, T - вектор моментов времени, его длина равна k.

Если выходные параметры функции не указываются (spectrogram (x, window, noverlap, nfft, Fs)), то строится трехмерный график спектральной плотности мощности в координатах: время, частота, уровень.

Обязательным входным параметром функции является вектор значений сигнала x, остальные параметры имеют значения по умолчанию, которые используются, если в качестве параметра указана пустая матрица ([]) или если несколько последних параметров при вызове опущены.

[V, fs, b]=wavread (‘c:\1\female1f.wav’); spectrogram (V, 256,128, [], fs, 'yaxis'); (рисунок 6.3.2).

Здесь во второй строке примера для параметра nnft используется значение по умолчанию - максимальное из двух чисел: 256 и 2^k. Значение к таково, что выполняется условие $2^k > \text{window}$.

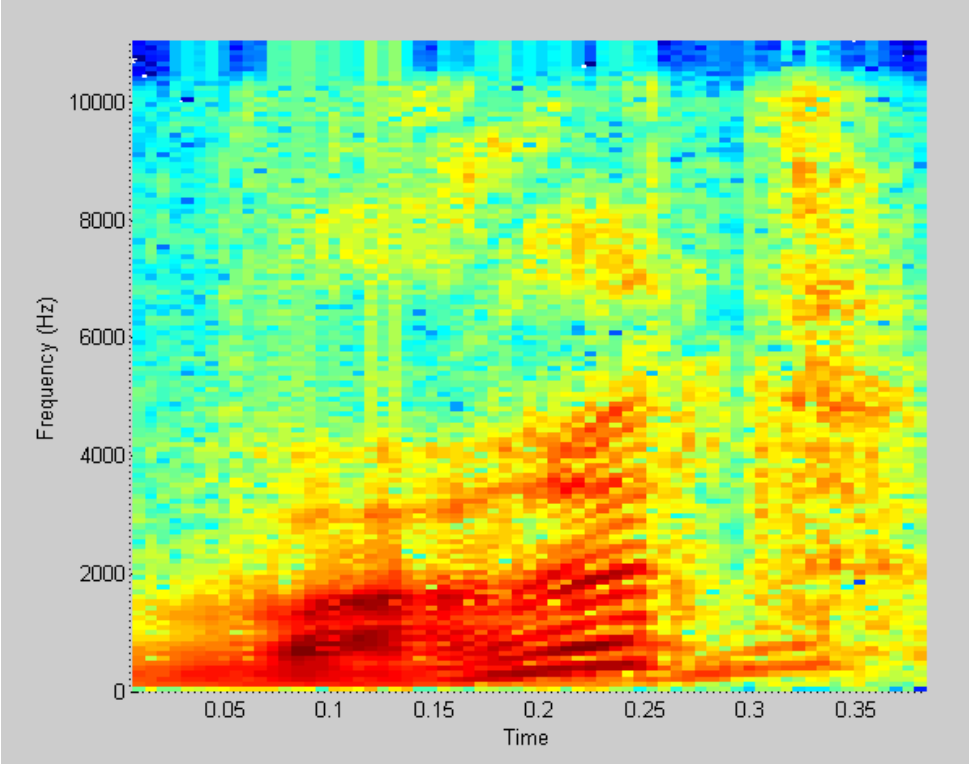


Рисунок 6.3.2 - Спектрограмма файла female1f.wav

Операции над звуковыми файлами

Система MATLAB позволяет читать и записывать файлы формата wav. Чтение осуществляется с помощью функции wavread, а запись - с помощью функции wavwrite.

```
[V, f, b] = wavread ('*.wav');
```

Звуковая информация из файла считывается в матрицу V, состоящую из двух столбцов, в скалярную переменную f считывается значение частоты дискретизации, а в переменную b - число двоичных разрядов.

Добавим к матрице V белый шум.

```
[m, n] = size(V); VI = randn (m, n); s = 0.05; V2 = V + 5* VI;
```

Здесь в первой строке определяется число строк и столбцов матрицы V. Вторая строка с помощью функции randn (m, n) создает матрицу VI, которая содержит случайные числа, распределенные по закону Гаусса (по нормальному закону), и которая имеет тот же

размер, что и матрица V. Матрица VI умножается на коэффициент **s** и складывается с матрицей V, образуя матрицу V2 зашумленного звука. Функция sound осуществляет воспроизведение звука, получая в качестве аргумента вещественный вектор или матрицу размерами Nx2 (для стереозвука), содержащие последовательности измерений громкости звука.

sound (V, f, b) sound (V2, f, b)

Значения элементов матрицы отсчетов должны быть ограничены диапазоном от -1.0 до +1.0. Вне этого диапазона значения матрицы ограничиваются (имеет место клиппированный звук). Аналогичная функция soundsc перед воспроизведением звука обеспечивает автоматическое масштабирование значений матрицы до диапазона: **-**1… +1 - клиппирования звука не происходит.

soundsc (V, f, b) soundsc (V2, f, b)

Чтобы сохранить результаты экспериментов со звуком в звуковом файле, следует применить функцию wavwrite. В данном случае результаты эксперимента с зашумлением сохраним в файле 'Soundl.wav':

wavwrite (V2, f, b, 'Soundl.wav');

Здесь первым аргументом является матрица (для монофонического звука - это вектор) звуковых отсчетов, вторым - частота дискретизации, третьим - разрядность отсчетов, а последним - имя файла. Если не указывать пути к файлу, то он будет записан в текущий рабочий каталог пакета MATLAB.

Исследование мужского и женского голосов

Для примера возьмем два образца женского и два образца мужского голосов. В пакете MATLAB построим их спектрограммы, амплитудный и фазовый спектры (рисунки 6.3.3 – 6.3.10).

Амплитудный спектр сроим следующим образом:

[s, fs] = wavread (c:\1\female1f.wav');= abs (fft(s));= spec (1:end/2);= (fs/2)*(1:length(spec))/length(spec);(freq, spec);('Frequency: Hz'); ylabel ('Amplitude spectrum');

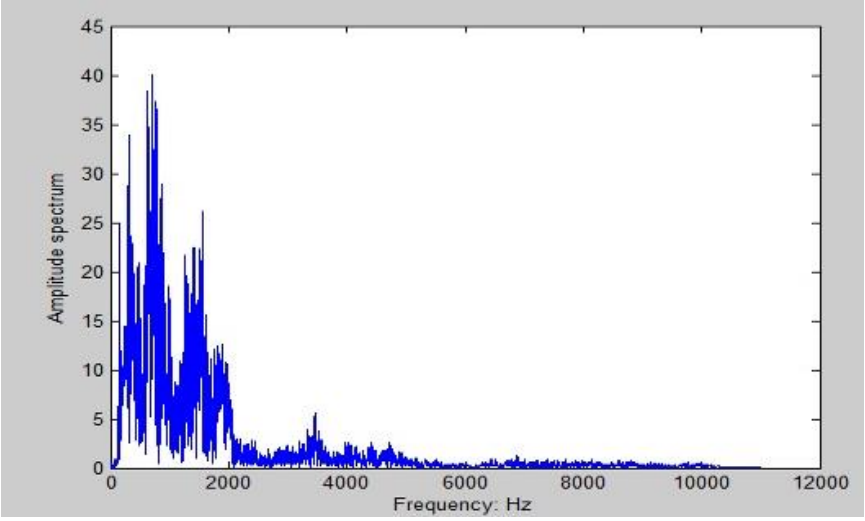


Рисунок 6.3.3 - Амплитудный спектр female1f.wav

Фазовый спектр

[s, fs] = wavread (c:\1\female1f.wav');= phase (fft(s));= spec (1:end/2);= (fs/2)*(1:length(spec))/length(spec);(freq, spec);('Frequency: Hz'); ylabel ('Phase spectrum');

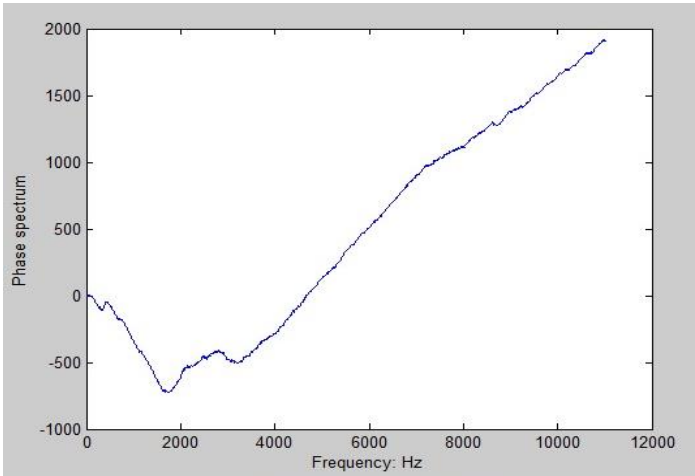


Рисунок 6.3.4 - Фазовый спектр female1f.wav

Аналогичные операции проделываем с остальными образцами голоса.

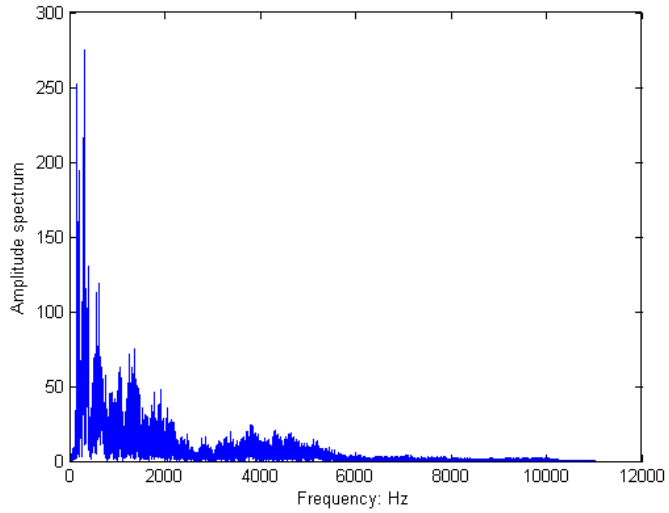


Рисунок 6.3.5 - Амплитудный спектр female2a.wav

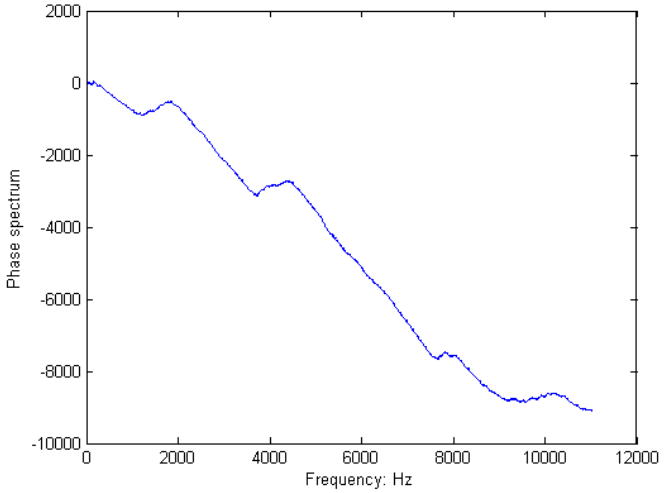


Рисунок 6.3.6 - Фазовый спектр female2a.wav

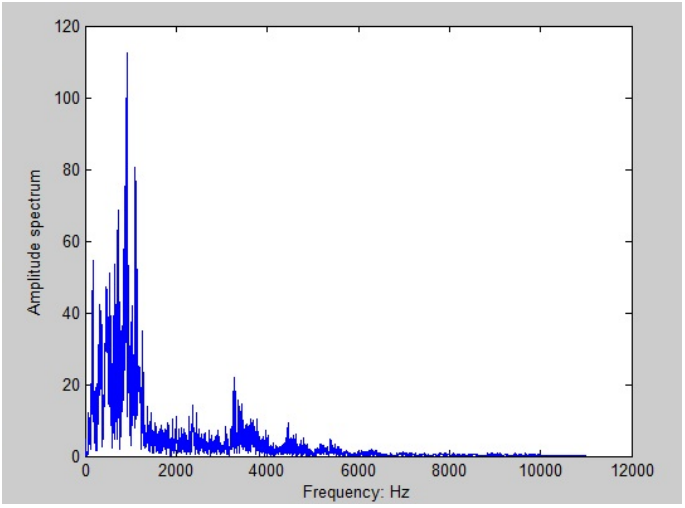


Рисунок 6.3.7 - Амплитудный спектр male1f.wav

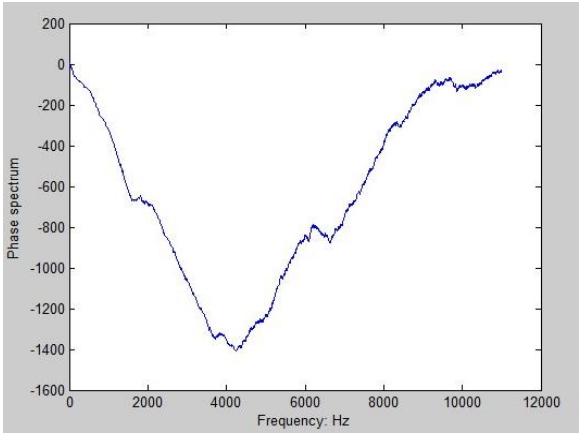


Рисунок 6.3.8 - Фазовый спектр male1f.wav

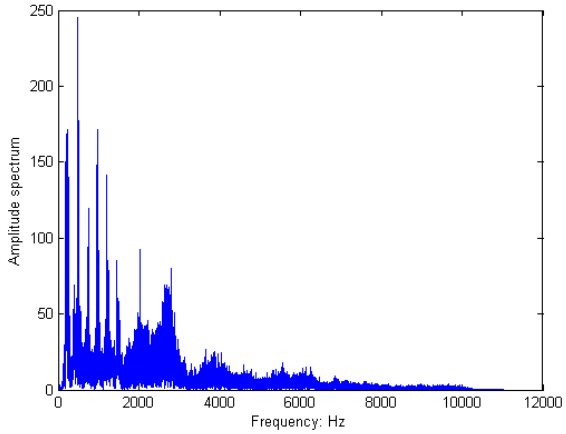


Рисунок 6.3.9 - Амплитудный спектр male2a.wav

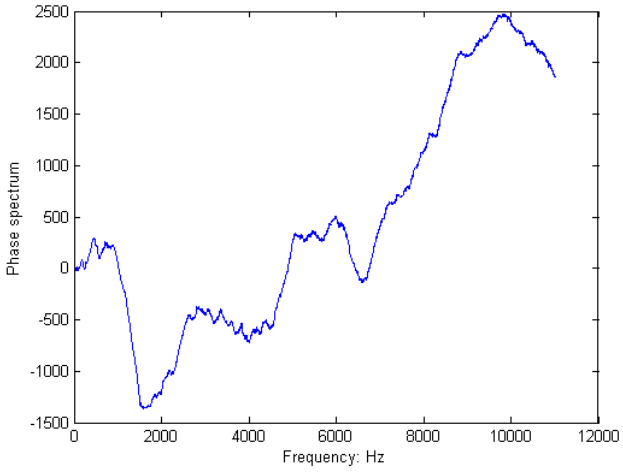


Рисунок 6.3.10 - Фазовый спектр male2a.wav

Построим спектрограммы голосов (рисунки 6.3.11-6.3.14)

[V, fs, b]=wavread (‘c:\1\female1f.wav’); spectrogram (V, 256,128, [], fs, 'yaxis');

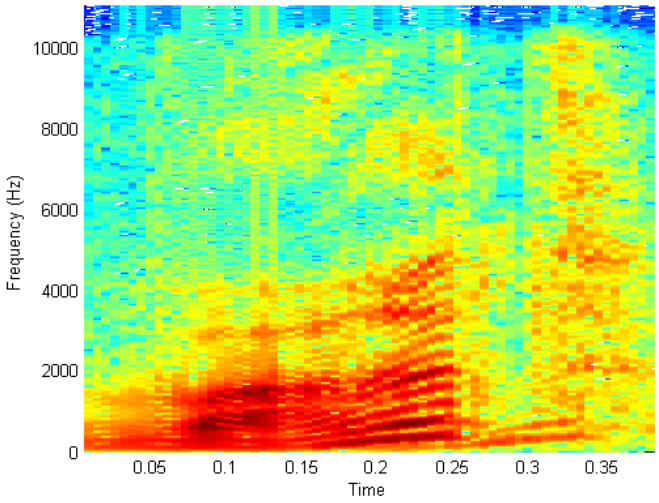


Рисунок 6.3.11 - Спектрограмма female1f.wav

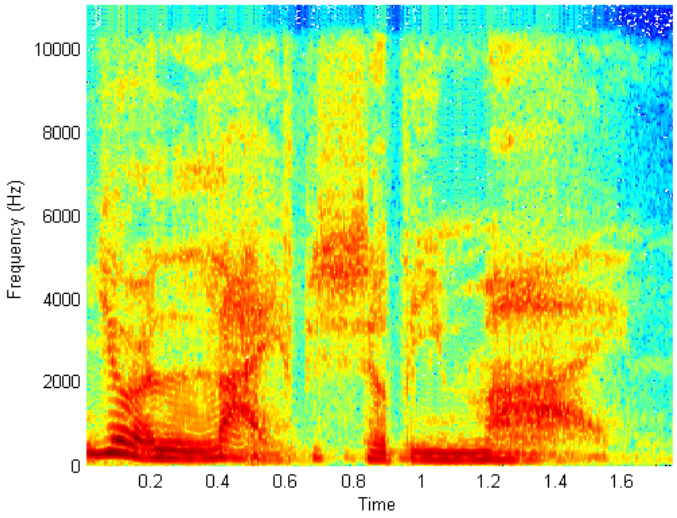


Рисунок 6.3.12 - Спектрограмма female2a.wav

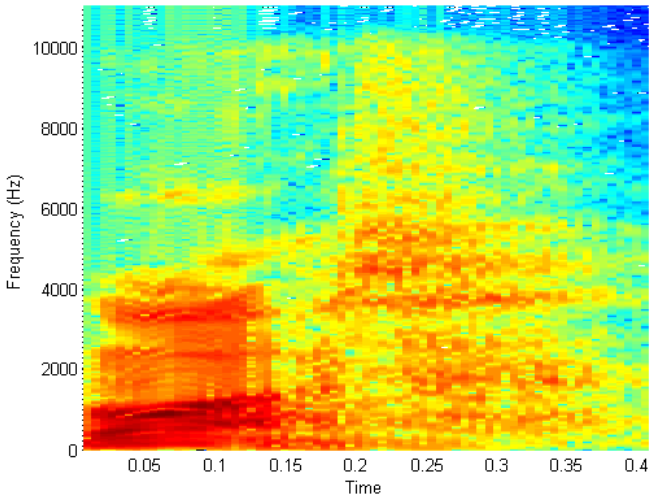


Рисунок 6.3.13 - Спектрограмма male1f.wav

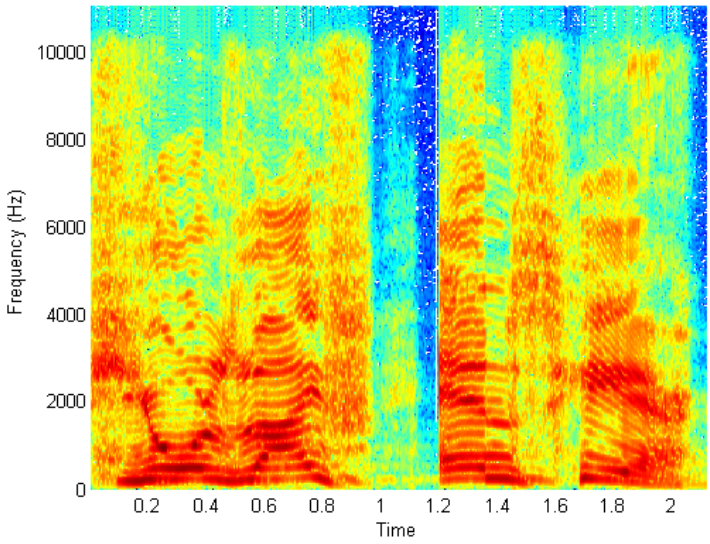


Рисунок 6.314 - Спектрограмма male2a.wav

Порядок выполнения работы

1. С помощью микрофонной гарнитуры введите в компьютер речевой сигнал (свою фамилию). Для этого удобно использовать программу «Звукозапись» из раздела «Стандартные – Развлечения».
2. С помощью команды wavread импортируйте речевой сигнал в среду Matlab. Определить параметры: Fs, bits.
3. Постройте график сигнала с помощью команды plot и subplot.
4. Определите время звучания и объем памяти для записанного сигнала.
5. Произведите обработку сигнала женского и мужского голоса.

Задание

1. Изучить теоретический материал.
2. Выберите данные фрагмента сигнала, где *n1*=5000, а *n2* рассчитывается по формуле *n2*=1000*N*, где *N* –номер студента в журнале.
3. Повторите пп. 3 и 4 порядка выполнения работы для фрагмента сигнала, полученного по пп. 2.

4. С помощью микрофонной гарнитуры и команды *wavrecord* введите в компьютер речевой сигнал (Фамилию студента)**,** применив Fs =8000; 11025.

5. Запишите вектор (или матрицу) по пп. 4 полученной записи на диск в виде wav-файла и mat-файла.

Содержание отчета

1. Цель работы.
2. Задание к работе.
3. Текст разработанной программы.
4. Таблицы и рисунки, полученные в результате выполнения лабораторной работы.
5. Выводы.

Контрольные вопросы

1. Приведите схему речеобразования у человека?
2. В каком частотном диапазоне находится женский голос?
3. В каком частотном диапазоне находится мужской голос?
4. Что такое фонема?
5. На какие группы делятся звуки речи?
6. На какие группы делятся согласные звуки?
7. Перечислите алгоритм записи данных из речевого сигнала в системе MATLAB.
8. Перечислите алгоритм считывания данных из речевого сигнала в системе MATLAB.
9. Каков синтаксис вызова функции спектрограммы?
10. Каковы особенности женского и мужского голоса?
11. С помощью каких программ можно осуществлять анализ и обработку речи?

Лабораторная работа №7
Базовые средства фильтрации шумов на изображении в системе MATLAB

Целью работы: изучить особенности работы и графические возможности системы MATLAB, способы выполнения команд в MATLAB; освоить возможности анализа и улучшения изображений, предоставляемые системой MATLAB и пакетом программ Image Processing Toolbox (IPT).

Теоретические сведения

Главной целью обработки изображений, является обнаружение объектов и идентификация этих объектов.

Целью обработки может являться также улучшение качества изображения для лучшего визуального восприятия, геометрические преобразования (масштабирование, поворот, в общем), нормализация изображений по яркости, контрастности, резкости, выделение границ изображений, автоматическая классификация и подсчет однотипных объектов на изображении, сжатие информации об изображении.

К основным видам искажений изображений, затрудняющих идентификацию, можно отнести:

- Недостаточную контрастность и яркость, связанную с недостаточной освещенностью объекта;
- Наличие шумов на изображении, вызванных различными факторами;
- Искажения, вызванные «зернистостью» из-за квантования изображения;
- Искажения, связанные с перемещением видеорегистратора или объекта во время экспозиции и связанную с этим «смазанность» изображения;
- Математические основы и методы обработки изображений применительно как к объектам живой, так и неживой природы.

Элементарные операции с изображениями

К элементарным операциям, используемых при обработке изображений, можно отнести:

1) Чтение и запись файлов с изображениями;
2) Преобразование графических форматов;
3) Улучшение качества изображений (путем выравнивания гистограммы, изменения яркости палитры, корректировки яркости и контрастности)
4) Масштабирование изображений;
5) Выполнение геометрических операций с изображениями (кадрирование, изменение размеров, поворот);
6) Функциональные преобразования (двумерное прямое и обратное дискретное косинусное преобразование, двумерное быстрое преобразование Фурье, прямое и обратное преобразование Радона);
7) Корреляционное сравнение изображений;
8) Фильтрация шумов на изображениях (медианная, ранговая, адаптивная и др.);
9) Выполнение операций с пикселами (построение контурных графиков объектов, вычисление признаков объектов, построение гистограммы изображения, построение профиля изображения);

Для обработки изображений может быть использована программная среда MATLAB, в которой имеется большое количество библиотечных функций для работы с изображениями и Visual C++ совместно с библиотекой обработки изображений Open CV (Computer Vision).

Чтение и запись файлов с изображениями

В пакете расширения Image Processing программы MATLAB используются различные форматы изображений: BMP, JPEG, PCX, PNG, TIFF, CUR, HDF, ICO, XWD и представляющих их файлов. Соответствующие типы файлов могут быть прочитаны с помощью функции imread. Функция A=imread(filename, fmt) читает из файла с именем filename полутоновое или полноцветное изображение и

создает двумерный массив A если изображение полутоновое или трехмерный массив размера $m \times n \times 3$ – если изображение полноцветное. Параметр fmt задает формат изображения и может принимать значения ‘bmp’, ‘jpg’, ‘jpeg’, ‘pcx’, ‘png’, ‘tif’,’cur’,’hdf’, ‘ico’, ‘xwd’.

Функция imwrite(A,filename,fmt) записывает изображение A в файл с именем filename с расширением fmt.

Вывод изображения на экран

```
J=imread ('moon.tif');
figure;
imshow (J);
```

Базовые средства фильтрации шумов на изображениях

Алгоритмов улучшения качества изображений известно достаточно много. Ряд из них ориентированы на уменьшения уровня шумов с целью обнаружения объектов на изображениях или идентификации объектов. Основные методы: фильтрация, прямое и обратное вейвлет-преобразование, деблюринг (устранение смазанности), накопление.

Применение фильтров

Для уменьшения шумов на изображениях предусмотрены несколько различных фильтров: усредняющий (average), медианный (medfilt2), Гаусса (gaussian).

Функции медианной фильтрации (medfilt2), ранговой фильтрации (ordfilt2) и адаптивной фильтрации (wiener2) наиболее эффективны для борьбы с зашумленностью изображения имеющего вид «снега».

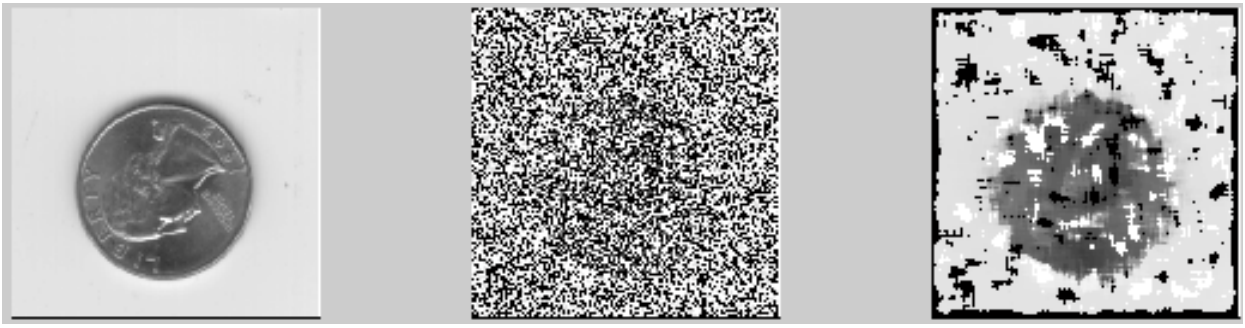
Усредняющий и медианный фильтры относится к классу фильтров низких частот. Они могут использоваться для уменьшения влияния шума на изображение, однако их применение приводит к размытию изображения и уменьшению его четкости.

Медианная фильтрация представляет собой нелинейную операцию, более эффективную, чем конволюция, когда целью

является одновременно уменьшение шума и сохранение границ объектов на изображении.

Функция медианной фильтрации J= medfilt2(I,[m n])фильтрует матрицу исходного изображения, используя маску фильтра размером $m \times n$. По умолчанию – маска имеет размер 3×3 пиксела. Центральный пиксел маски заменяют медианой всех ее пикселов. Маска применяется нерекурсивно ко всему изображению. На краях изображение дополняется нулевыми элементами при классе I – unit8 или единицами при классе I – double, поэтому в результате фильтрации крайние элементы изображения могут быть искажены.

Эффект улучшения изображения при использовании медианного фильтра показан на рисунке 7.1.

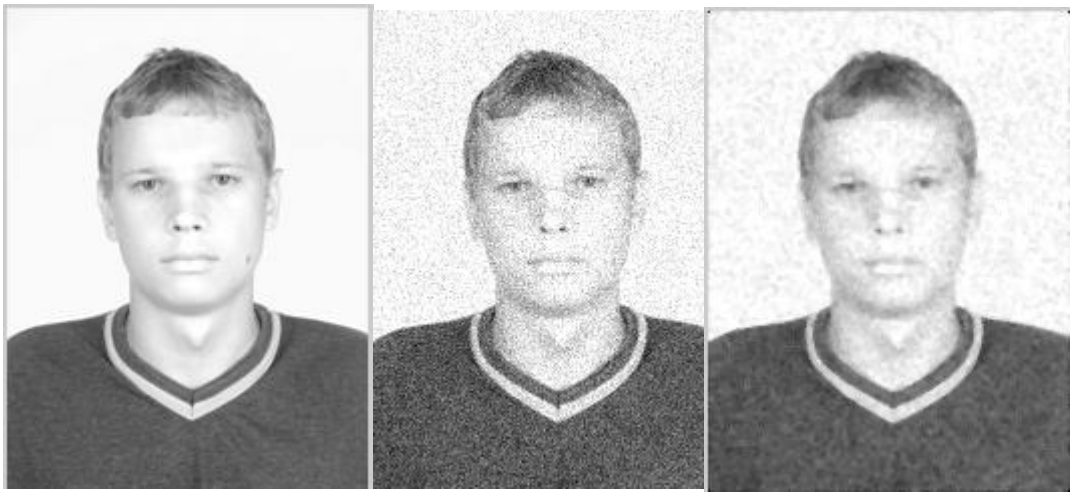


А *Б* *В*

Рисунок 7.1 - Исходное изображение хорошего качества(А), зашумленное (Б) и полученное в результате фильтрации (В)

В данном случае применение фильтра решает задачу обнаружения объекта на изображении. На рисунке 7.2 показан пример улучшения качества изображения, выполняемого с целью достижения максимальной идентичности, оцениваемой по коэффициенту кросскорреляции, отфильтрованного изображения с исходным незашумленным изображением.

Усредняющий фильтр (average) функционирует таким же образом, как и медианный, но отличается тем, что центральный пиксел маски вычисляется как среднеарифметическое значение всех ее пикселов. Если медианный фильтр эффективнее при шуме типа «соль и перец», то усредняющий фильтр имеет преимущество при гауссовом белом шуме.



А *Б* *В*

Рисунок 7.2 - Исходное (А), зашумленное (Б) и полученное в результате фильтрации (В) изображения. Вид шума – гауссовый белый, variance=0.005.

С точки зрения визуального восприятия зашумленное изображение выглядит лучше, чем отфильтрованное, однако отфильтрованное более близко к исходному при корреляционном сравнении. Коэффициент корреляции зашумленного изображения с исходным – 0.80, отфильтрованного с исходным – 0.92. Ниже приведен пример программ усредняющего и медианного фильтров.

```
%Усредняющий фильтр 'Average'
hsize=[3 3 ]; %маска фильтра
h= fspecial('average',hsize);
J=imfilter(I,h,'replicate');
figure;
imshow (J);
title('После усредняющего фильтра');

%Медианный фильтр 'medfilt2'
hsize=[3 3 ]; %маска фильтра
J= medfilt2(I,hsize);
figure;
imshow(J);
title('После медианного фильтра');
```

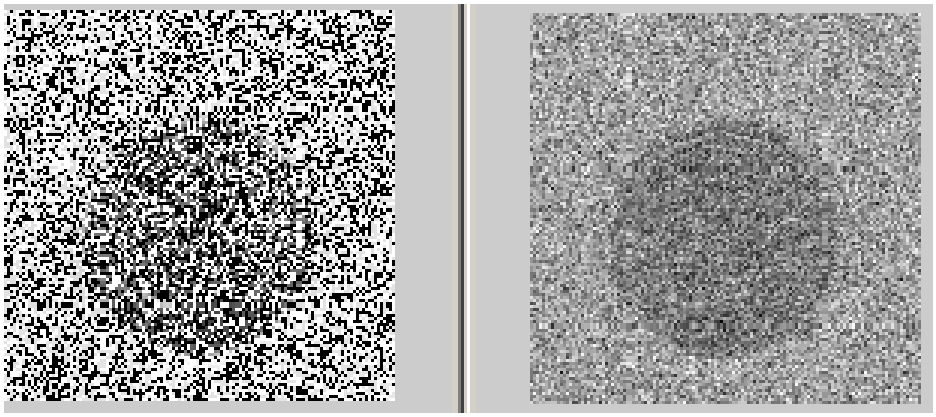
Фильтр Гаусса (gaussian) также относится к классу фильтров низких частот, но, по сравнению с усредняющим фильтром при гауссовом белом шуме, он меньше размывает изображение. Отфильтрованное изображение получается также близко к исходному при корреляционном сравнении. Коэффициент корреляции зашумленного изображения рис. 2Б при гауссовом белом шуме с variance = 0.005 с исходным – 0.80, отфильтрованного с исходным при использовании как усредняющего фильтра, так и фильтра Гаусса – 0.97. Ниже приведен пример программы фильтра Гаусса.

```
%Фильтр Гаусса 'gaussian'
hsize=[9 9];
sigma=0.99;
h= fspecial('gaussian',hsize,sigma);
J=imfilter(I,h,'replicate');
figure;
imshow(J);title('После фильтра Гаусса');
```

Однозначный выбор вида фильтра при обработке изображения может быть сделан с учетом решаемой задачи и только в случае, если точно известен характер шума. Если же характер шума неизвестен, нужно пробовать применить для обработки изображения последовательно все вышеупомянутые фильтры.

Улучшение отношения сигнал/шум путем накопления

Еще один возможный вид обработки изображений с целью улучшения отношения сигнал/шум – накопление, т.е. усреднение нескольких изображений. Благодаря тому, что изображение объекта остается неизменным, а шумовая составляющая изменяется, в результате накопления происходит улучшение отношения сигнал/шум. На рисунке 7.3 приведен пример эффекта улучшения отношения сигнал/шум в результате 10 накоплений.



А *Б*

Рисунок 7.3 - Зашумленное изображение (А) и результат 10-ти накоплений (Б)

Эффект улучшения отношения сигнал/шум может быть достигнут при условии, если положение объекта в кадре остается неизменным. Поэтому необходимо устранение эффектов смещения объекта в кадре, связанного с качкой и перемещением корабля, на котором установлена система.

Таким образом, средства работы с фотоизображениями, представленные в MATLAB, являются достаточно полными и эффективными при решении задачи улучшения качества изображений. Кроме того, MATLAB является открытой системой, т.е. дает возможность модифицировать имеющиеся программы обработки и создавать другие.

Время выполнения базовых функций MATLAB для работы с изображениями на компьютере IBM PC класса Pentium-4 приведено в таблице 7.1.

Таблица 7.1 - Время выполнения функций для работы с изображениями, мсек

№№	ВЫПОЛНЯЕМАЯ ФУНКЦИЯ	Формат изображения	
		640x480	800x600
11	Чтение файла изображения	32	36
22	Отображение изображения	420	420
66	Ранговая фильтрация (Улучшение фокусировки)	130	220
77	Медианная фильтрация	100	140

Наложение шумов на изображение

Для выбора наилучшего метода фильтрации шумов и его параметров необходимо знать статистические свойства шума и параметры изображения, такие как яркость, контрастность, резкость, наличие или отсутствие смазанности. При этом возможно проведение модельного эксперимента, который позволит изучить эффективность того или иного метода фильтрации шумов при вариациях параметров изображения и параметров шума.

Для добавления шума к изображению MATLAB предоставляет функцию J=imnoise(I, type), где I – исходное изображение, type – класс шума:

- ‘gaussian’ – гауссовый белый шум;
- “salt & pepper” –«соль и перец», шум в виде включенных или выключенных пикселов;
- ‘speckle’ –мультипликативный шум.

Функция J=imnoise(I, type, parameters) позволяет дополнительно задать параметры шума, например:

- J=imnoise(I, ‘gaussian’, m, v) добавляет гауссовый белый шум со средним значением m и отклонением v (по умолчанию m=0, v=0.01);
- J=imnoise(I, ‘salt & pepper’, d) параметр d задает плотность шума, по умолчанию d=0.05;
- J=imnoise(I, ‘speckle, v) добавляет мультипликативную компоненту шума, так что J=I + n*I, где n – равномерно распределенный шум со средним значением 0 и среднеквадратичным отклонением v (по умолчанию v=0.04). Этот шум не виден на темных участках изображения, но проявляется на его светлых участках.

```
%Программа производит зашумление изображения m00200011.tif
%тремя различными способами

img1=imread ('m00200011.tif'); subplot(2,2,1);
imshow (img1);
title('Исходное изображение');
```

```
img2=imnoise(img1,'gaussian',0.03);subplot(2,2,2);
imshow (img2);
title('С гауссовым шумом');
img3=imnoise(img1,'salt & pepper',0.02);subplot(2,2,3);
imshow (img3);
title('С шумом "соль и перец"');
img4=imnoise(img1,'speckle',0.03);subplot(2,2,4);
imshow (img4);
title('С мультипликативным шумом');
```



Рисунок 7.4 - Пример исходного и зашумленных изображений

Входное изображение должно быть представлено матрицей классов unit8, unit16 или double. Выходной сигнал имеет тот же класс, что и входной. Файл входного черно-белого или цветного изображения может быть представлен в формате .bmp, .jpg, .tif, .gif и т.д.

Порядок выполнения работы

1. Загрузить изображение.
2. Установить уровень шума 0.005, вид шума – Гауссов, произвести кадрирование изображения, выделив центральную часть лица, и зафиксировать в таблице при выполнении программы максимальные значения коэффициентов корреляции зашумленного изображения и отфильтрованных разными фильтрами изображений. Затем, изменяя параметры фильтров, добиться получения наиболее высоких значений коэффициентов корреляции отфильтрованных сигналов с исходным и вспомогательным.
3. Повторить выполнение п. 2 при шуме salt & pepper с параметрами d=0.01, 0.05 и 0.1 и при мультипликативном шуме speckle с нулевым средним значением и среднеквадратичным отклонением v=0.02, 0.04 и 0.06.
4. При выполнении п. 3 спроектировать на внутреннем языке MATLAB программу цифровой обработки изображений, реализующую уменьшение шумов на изображениях , используя стандартные базовые программы:

- average – усредняющий фильтр;
- medfilt – медианный фильтр;
- gaussian – фильтр Гаусса;
- ordfilt – ранговый фильтр;
- unsharp – фильтр повышения резкости;
- wiener2 – адаптивный фильтр Винера;

Программа должна позволять производить интерактивный выбор вида фильтра и параметров фильтра

Задание

1. Исследовать связь эффективности фильтрации зашумленных изображений с визуальным восприятием изображений и степенью сходства отфильтрованного изображения с исходным незашумленным изображением, найти оптимальные параметры фильтров;
2. Исследовать связь эффективности фильтрации от показателей яркости и контрастности исходного изображения;

3. Спроектировать на внутреннем языке MATLAB программу цифровой фильтрации изображений, содержащей средства интерфейса пользователя, позволяющие выбирать вид и задавать параметры фильтрации (приложение 6).

Содержание отчета

1. Цель работы.
2. Задание к работе.
3. Таблицы с показателями корреляционного сравнения изображений по п. 2 и 3 в соответствии порядка выполнения работы с комментариями.
4. Текст программы по п. 4 в соответствии порядка выполнения работы и экранная форма.
5. Выводы.

Контрольные вопросы

1. Какие основные виды искажений изображений, затрудняющих идентификацию, можно выделить?
2. Перечислите элементарные операции, производимые над изображениями.
3. Какие команды используются для чтения и записи файлов с изображением?
4. Какие фильтры применяются для обработки изображений?
5. Перечисли алгоритм наложения шума на изображение.

Библиографический список

1. Ахмад, Х.М. Ведение в цифровую обработку речевых сигналов: учеб. пособие / Х.М. Фхмад, В.Ф. Жирков; Владимир: Гос. ун-т. – Владимир: Издательство Владимир Гос. ун-т, 2007. – 192 с.

2. Дьяконов, В.П. Энциклопедия Mathcad 2001i и Mathcad 11 / В.П. Дьяконов. – М.:СОЛОН – Пресс, 2004. – 832 с.

3. Зубов, В.С. Программирование на языке TURBO PASKAL (версии 6.0 и 7.0) / В.С. Зубов - М.: Информационно-издательский дом "Филинъ", 1997.-304с.

4. Исследование алгоритмов обработки сигналов в системе MATLAB: метод. Указания к лабораторным работам / Владим. гос. ун-т; сост. Е.К. Левин. - Владимир: Издательство Владимир Гос. ун-т, 2011. – 78 с.

5. Симонович, С.В. Информатика: Базовый курс / С.В.Симонович и др. – Спб.:Питер, 2004. – 640 с.

6. Каганов, В.И. Радиотехника + компьютер + Mathcad / В.И. Каганов. – М.:Горячая линия – Телеком, 2001. – 416 с.

7. Немирко, А.П. Обработка данных в среде MATLAB: методические указания к практическим и лабораторным занятиям / Сост.: А. П. Немирко, Л. А. Манило. - СПб.: Изд-во СПбГЭТУ «ЛЭТИ»,2013. - 35 с

8. Очков, В.Ф. Mathcad 12 для студентов и инженеров / В.Ф. Очков. – СПб.: БХВ – Петербург, 2005. – 464 с.

9. Сергиенко, А.Б. Цифровая обработка сигналов / А.Б. Сергиенко. - СПб. Питер. 2002.-608 с.

10. Тутыгин, В.С. Цифровая обработка сигналов: лаборат. практикум / В.С. Тутыгин. - СПб.: СПбГПУ, 2012. - 58с.

11. Циммерман, Ф. Клиническая электрокардиография / Ф. Циммерман. - М.: «Издательство БИНОМ», 1997. - 448с.

Приложение 1

Пример программного кода, генерируемого автоматически после создания панели и объектов GUI в среде MATLAB

```
%Служебная часть М-файла, создаваемая автоматически после сохранения в файле
%панели интерфейса пользователя <имя>.fig
%В данном примере панель интерфейса пользователя содержит кнопку управления
%PushButton, окно ввода/вывода текста EditText (tag – KP), окно графического
%вывода Axes (tag – axis1)

function varargout = testlab2(varargin)
% TESTLAB2 M-file for testlab2.fig
% TESTLAB2, by itself, creates a new TESTLAB2 or raises the existing
% singleton*.
%
% H = TESTLAB2 returns the handle to a new TESTLAB2 or the handle to
%the existing singleton*.
%
%      TESTLAB2('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in TESTLAB2.M with the given input arguments.
%
%TESTLAB2('Property','Value',...) creates a new TESTLAB2 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before testlab2_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property application
```

```
%       stop.  All inputs are passed to testlab2_OpeningFcn via
varargin.
%
%       *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help testlab2

% Last Modified by GUIDE v2.5 03-Feb-2012 21:54:41

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @testlab2_OpeningFcn, ...
                   'gui_OutputFcn',  @testlab2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

 % --- Executes just before testlab2 is made visible.
function testlab2_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to testlab2 (see VARARGIN)

% Choose default command line output for testlab2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes testlab2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = testlab2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function KP_Callback(hObject, eventdata, handles)
% hObject    handle to KP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP as text
%        str2double(get(hObject,'String')) returns contents of KP as a double
```

```
% --- Executes during object creation, after setting all properties.
function KP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
KP=str2double(get(handles.KP,'string'));
for i=1:1000
   y(i)=sin((2*pi*i*KP)/1000.0);
end
i=1:1000;
plot(i,y);

% --- Executes on button press in Quit.
function Quit_Callback(hObject, eventdata, handles)
% hObject    handle to Quit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
fclose('all');
close ('all');
```

Приложение 2

Пример программного кода в функции Start

```
% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
KP=str2double(get(handles.KP,'string'));
for i=1:1000
   y(i)=sin(2*pi*i*KP./1000.0);%KP – кол-во периодов
end
i=1:1000
plot(i,y);
```

После запуска программы Debug/Run диалоговое окно программы с результатами выполнения будет выглядеть, как показано на рисунке 5.3.

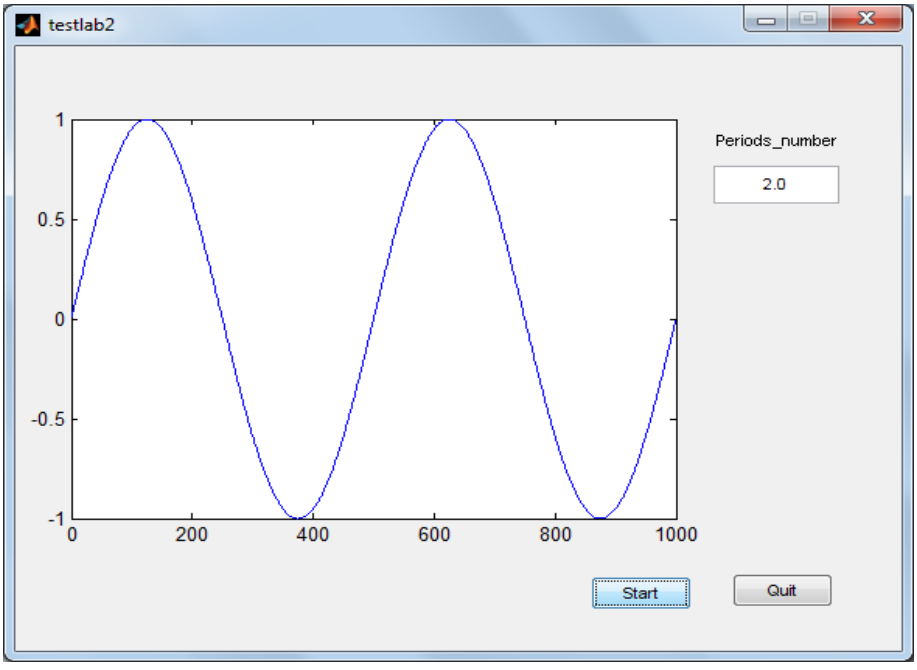


Рисунок 5.3 – Результат выполнения работы программы

Приложение 3

Построение и оформление графиков в MATLAB

```
 i=1:N;
y=A*sin(6.28*KP*i/N); %создание N значений KP периодов модельного sin %сигнала,
figure %создание окна графического вывода
%plot(i,y(1:50));%вывод графика y, кол-во точек – 50.
%plot(i,y) ;% вывод графика y (все N точек)
%plot (i,2*y(1:N),'r-');%отображение графика y линией красного цвета
%semilogx(i,abs(y(1:200))),grid; %то же, что и plot, но в логарифмическом
 %  масштабе по X
%loglog(i,abs(y(1:200))); %  в логарифмическом масштабе по X и Y
grid on; %отображение линий координатной сетки
%title('сигнал до фильтра');%заголовок графика
xlabel('номер отсчета'); % подпись по оси X
ylabel('амплитуда'); % подпись по оси Y
legend('до фильтра');%подпись легенды
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax %(автомасштабирование)
hold on; % «удержание» окна вывода для следующего графика
hold off; %отмена «удержания»
close all; %закрытие всех открытых окон графического вывода
```

Приложение 4

Текст базовой программы

```
%Комбинированное использование ключевых операций ЦОС
%Для повышения точности определения количества периодов и частоты
%"короткого" сигнала используется комбинация
%БПФ, кросскорреляции, сплайн-аппроксимации,передискретизации
kt=1024; % количество отсчетов
f=12; %частота сигнала
dt=2;%шаг дискретности по времени при измерении
k2=500; %декремент затухания огибающей сигнала
shum=0;%шум
kp=4.5;%количество периодов сигнала

%1. ГЕНЕРАЦИЯ МОДЕЛЬНОГО СИГНАЛА
for i=1:kt %обнуление массива сигнала
   y(i)=0;
end
for i=1:kt %генерация модельного сигнала
   if(i>0)&(i<=kt/2)
     y(i)=sin(2*3.14*kp*i/kt)*i*exp(i/k2)/1200;
   end
   if(i>kt/2)&(i<(kt))
     y(i)=sin(2*3.14*kp*i/kt)*(kt-i)*exp((kt-i)/k2)/1200;
   end
    y(i)=y(i)+shum*(2*rand(1)-1);
end
i=1:kt; %отображение модельного сигнала во временной области
figure
plot(i,y);
axis tight;
title('Time domain')
xlabel('Sample number')
```

```
%2. ФУНКЦИОНАЛЬНОЕ ПРЕОБРАЗОВАНИЕ (БПФ)
bpfy=fft(y,kt);%БПФ
bpf=bpfy.*conj(bpfy)/kt;%БПФ
%f=1000*(0:256)/512;
figure
plot(i(1:257),bpf(1:257));
axis tight;
title('Frequency domain')
xlabel('frequency')
%нахождение макс. знач. функции БПФ для массива Y
C=max(bpf);
for i=1:kt %поиск количества периодов, соответствующих максимуму БПФ
   if (bpf(i)==C)
      kpbpf=(i-1);
      break
   end
end
kp_bpf=kpbpf
%3. СОЗДАНИЕ ЭТАЛОНОВ И КРОССКОРРЕЛЯЦИЯ
fr=kpbpf;
   procch=0.1;%область поиска в процентах относит. kp_bpf
shagkor=fr*procch/3;%шаг поиска
      k=0;
  for iii=fr-fr*procch:shagkor:fr+fr*procch %цикл для создания 6 эталонов в окрестности приближенного
            %значения количества периодов, определенных с помощью БПФ.
            k=k+1;
            xkor(k)=iii;
            kor(k)=0;
            for i=1:kt
               x(i)=0;
            end
%Вычисление массивов эталонных сигналов
      for i=1:kt
         if(i>0)&(i<=kt/2)
```

```
            x(i)=sin(2*3.14*iii*i/kt)*i*exp(i/k2)/1200;
          end
          if(i>kt/2)&(i<(kt))
            x(i)=sin(2*3.14*iii*i/kt)*(kt-i)*exp((kt-i)/k2)/1200;
          end
        end
    %вычисление средних значений модельного и эталонных сигналов
          x_sr=mean(x);
          y_sr=mean(y);
          x_sko=0;
          y_sko=0;
    %вычисление СКО и коэф. корреляции модельного и эталонных сигналов
          for i=1:kt
            x_sko=x_sko+(x(i)-x_sr)*(x(i)-x_sr);
            y_sko=y_sko+(y(i)-y_sr)*(y(i)-y_sr);
            kor(k)=kor(k)+(x(i)-x_sr)*(y(i)-y_sr);
          end
          kor(k)=kor(k)/(sqrt(x_sko*y_sko));
      end %конец цикла создания эталонов и вычисления массива коэф. корр.
      %СПЛАЙН-АППРОКСИМАЦИЯ И ПЕРЕДИСКРЕТИЗАЦИЯ
        xx=1:k;
        xi=1:0.1:k;
        r1=sin(xx); %только для тестирования сплайн-аппроксимации
        yint=interp1(xx,kor,xi,'spline');% сплайн-аппроксимация коэф корреляции
        r1=kor;
        %%apr=csaps(xx,r1);
        apr=spaps(xkor,kor,0.000001);%%%%%%%%%%%%%%%
         fnplt(apr)
         hold on
         plot(xkor,r1,'ro')%%%%%%%%%%%%%%%%%
         hold off
```

```
%НАХОЖДЕНИЕ УТОЧНЕННОГО ЗНАЧЕНИЯ КОЛИЧЕСТВА ПЕРИОДОВ СИГНАЛА
        cmax=max(yint); %нахождение максимума коэф. корр.
        for i=1:round((k-1)/0.1+1)
           if (yint(i)==cmax)
              kp_int=fr-fr*procch+(i-1)*shagkor/10      %уточненное значение частоты по MAX функции коэф. корр.
           end
        end
    pause;
    close all;
```

Приложение 5

Текст базовой программы

```
%Программа производит зашумление изображения
%фильтрацию зашумленного изображения с помощью
%различных фильтров, определение коэффициента корреляции
%зашумленного и отфильтрованных изображений с исходным

img=imread ('m00200021.tif');
I=imcrop(img);
figure;
imshow (img);
title('Исходное изображение');
img1=imnoise(img,'gaussian',0,0.005);% 0.005
figure;
imshow (img1);
title('Зашумленное изображение');

%Медианный фильтр 'medfilt2'
hsize=[3 3 ];
F4= medfilt2(img1,hsize);
figure;
imshow(F4);
title('После медианного фильтра');
pause;

%Усредняющий фильтр 'Average'
hsize=[3 3 ];
h= fspecial('average',hsize);
F1=imfilter(img1,h,'replicate');
figure;
imshow (F1);
title('После усредняющего фильтра');
pause;
```

```
ncorr = normxcorr2(I(:,:,1),img1(:,:,1));
figure, surf(ncorr), shading flat;
title('После зашумления');
max_c_noised = max(abs(ncorr(:)))
pause;

 ncorr = normxcorr2(I(:,:,1),F4(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После медианного фильтра');
      max_c_median = max(abs(ncorr(:)))
 pause;

 ncorr = normxcorr2(I(:,:,1),F1(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После усредняющего фильтра');
 max_c_average = max(abs(ncorr(:)))
 pause;

%Фильтр Гаусса 'gaussian'
hsize=[9 9];
sigma=0.99;
h= fspecial('gaussian',hsize,sigma);
F2=imfilter(img1,h,'replicate');
figure;
imshow(F2);title('После фильтра Гаусса');
pause;

 ncorr = normxcorr2(I(:,:,1),F2(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  фильтра Гаусса');
 max_c_gaussian = max(abs(ncorr(:)))
 pause;

%Фильтр Лапласа 'laplacian'
alpha=0.5;
h= fspecial('laplacian',alpha);
F3=imfilter(img1,h,'replicate');
figure;
```

```
imshow(F3);title('После фильтра Лапласа');
pause;

 ncorr = normxcorr2(I(:,:,1),F3(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  фильтра Лапласа');
 max_c_laplasian = max(abs(ncorr(:)))
 pause;

 %Фильтр повышения резкости 'unsharp'
alpha=0.2;
h= fspecial('unsharp',alpha);
F4=imfilter(img1,h);
figure;
imshow(F4);title('После фильтра unsharp');
pause;

 ncorr = normxcorr2(I(:,:,1),F4(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  фильтра unsharp');
 max_c_unsharp = max(abs(ncorr(:)))
 pause;

%Ранговый фильтр
 m=5;n=5;
J = imnoise(img1,'gaussian',0,0.005);
imshow(J);
J6 = ordfilt2(J, 7, ones(4,4));
figure;
imshow(J6);title('После рангового фильтра');
pause;

 ncorr = normxcorr2(I(:,:,1),J6(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  рангового фильтра');
 max_c_ordfilt2 = max(abs(ncorr(:)))
 pause;

 close all;
```

Приложение 6

Текст базовой программы

```
%Программа производит зашумление изображения
%фильтрацию зашумленного изображения с помощью
%различных фильтров, определение коэффициента корреляции
%зашумленного и отфильтрованных изображений с исходным

 img=imread ('m00200021.tif');
 I=imcrop(img);
 figure;
 imshow (img);
 title('Исходное изображение');
 img1=imnoise(img,'gaussian',0,0.005);% 0.005
 figure;
 imshow (img1);
 title('Зашумленное изображение');

%Медианный фильтр 'medfilt2'
hsize=[3 3 ];
F4= medfilt2(img1,hsize);
figure;
imshow(F4);
title('После медианного фильтра');
pause;

%Усредняющий фильтр 'Average'
hsize=[3 3 ];
h= fspecial('average',hsize);
F1=imfilter(img1,h,'replicate');
figure;
imshow (F1);
title('После усредняющего фильтра');
pause;
```

```
ncorr = normxcorr2(I(:,:,1),img1(:,:,1));
figure, surf(ncorr), shading flat;
title('После зашумления');
max_c_noised = max(abs(ncorr(:)))
pause;

 ncorr = normxcorr2(I(:,:,1),F4(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После медианного фильтра');
      max_c_median = max(abs(ncorr(:)))
 pause;

 ncorr = normxcorr2(I(:,:,1),F1(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После усредняющего фильтра');
 max_c_average = max(abs(ncorr(:)))
 pause;

%Фильтр Гаусса 'gaussian'
hsize=[9 9];
sigma=0.99;
h= fspecial('gaussian',hsize,sigma);
F2=imfilter(img1,h,'replicate');
figure;
imshow(F2);title('После фильтра Гаусса');
pause;

 ncorr = normxcorr2(I(:,:,1),F2(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  фильтра Гаусса');
 max_c_gaussian = max(abs(ncorr(:)))
 pause;

%Фильтр Лапласа 'laplacian'
alpha=0.5;
h= fspecial('laplacian',alpha);
F3=imfilter(img1,h,'replicate');
figure;
imshow(F3);title('После фильтра Лапласа');
```

```
pause;

 ncorr = normxcorr2(I(:,:,1),F3(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  фильтра Лапласа');
 max_c_laplasian = max(abs(ncorr(:)))
 pause;

 %Фильтр повышения резкости 'unsharp'
alpha=0.2;
h= fspecial('unsharp',alpha);
F4=imfilter(img1,h);
figure;
imshow(F4);title('После фильтра unsharp');
pause;

 ncorr = normxcorr2(I(:,:,1),F4(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  фильтра unsharp');
 max_c_unsharp = max(abs(ncorr(:)))
 pause;

%Ранговый фильтр
 m=5;n=5;
 J = imnoise(img1,'gaussian',0,0.005);
 imshow(J);
 J6 = ordfilt2(J, 7, ones(4,4));
 figure;
 imshow(J6);title('После рангового фильтра');
 pause;

 ncorr = normxcorr2(I(:,:,1),J6(:,:,1));
 figure, surf(ncorr), shading flat;
 title('После  рангового фильтра');
 max_c_ordfilt2 = max(abs(ncorr(:)))
 pause;

 close all;
```