

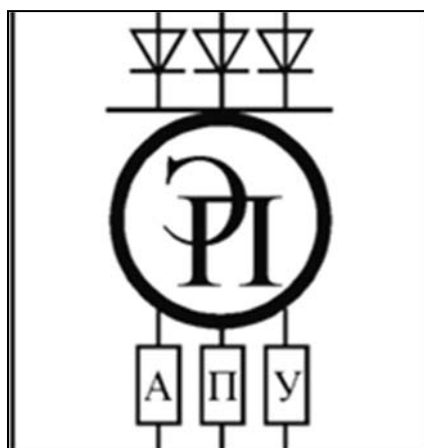
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электропривод и АПУ»

ИНФОРМАТИКА

*Методические рекомендации к лабораторным работам
для студентов специальности 1-53 01 05
«Автоматизированные электроприводы»
очной и заочной форм обучения*

Часть 2



Могилев 2023

УДК 004
ББК 32.97
И94

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой ЭПиАПУ «11» января 2023 г., протокол № 5

Составители: В. Н. Абабурко, А. П. Корнеев

Рецензент В. В. Кутузов

Методические рекомендации к лабораторным работам для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» очной и заочной форм обучения. Методические рекомендации включают цель, описание задания, обобщенный ход выполнения, содержание отчета и контрольные вопросы к каждой из лабораторных работ.

Учебное издание

ИНФОРМАТИКА

Часть 2

Ответственный за выпуск	С. М. Фурманов
Корректор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 81 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019 .

Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

14 Лабораторная работа № 14. Текстовый редактор Microsoft Word.....	4
15 Лабораторная работа № 15. Изучение редактора AutoCAD.....	7
16 Лабораторная работа № 16. Среда компьютерной математики Mathcad.....	10
17 Лабораторная работа № 17. Математическая система MATLAB.....	13
18 Лабораторная работа № 18. Численное решение уравнений	15
19 Лабораторная работа № 19. Методы численного интегрирования обыкновенных дифференциальных уравнений.....	18
20 Лабораторная работа № 20. Интерполяция табличных функций	21
21 Лабораторная работа № 21. Аппроксимация табличных функций	25
22 Лабораторная работа № 22. Оптимизация однопараметрических функций	28
23 Лабораторная работа № 23. Основы программирования сайтов Internet.....	31
24 Лабораторная работа № 24 «Табличный редактор Microsoft Excel.....	35
25 Дополнительная информация об изучаемой дисциплине	39
Список литературы	40

Часть 2

14 Лабораторная работа № 14. Текстовый редактор Microsoft Word

Цель работы:

- изучение текстового редактора Microsoft Word;
- получение практических навыков по созданию документов Microsoft Word на персональном компьютере, соответствующим требованиям ЕСКД.

Задание

Заданием на лабораторную работу является разработка электронного документа отчета, выполненного в соответствии с требованиями ГОСТ 2.105 на основной надписи согласно ГОСТ 2.104.

Текст работы должен содержать следующее:

- 1) содержание с указанием номеров страниц разделов и подразделов, сгенерированного средствами редактора;
- 2) цель лабораторной работы;
- 3) основная часть работы, куда входят следующие подразделы:
 - ответ на теоретический вопрос (согласно варианту задания);
 - вставка изображения и описания заданной формулы;
 - работа с заданной таблицей;
- 4) список использованных источников, выполненный в порядке упоминания в тексте задания в соответствии с ГОСТ 7.1.

14.1 Ход выполнения работы

14.1.1 Загрузка текстового редактора Microsoft Word или его альтернативного варианта после изучения [9, 12].

14.1.2 Создается новый текстовый документ на основе файла шаблона пояснительной записки с настройками текста в соответствии с требованиями ГОСТ 2.105, а также колонтитулами в виде основной надписи по ГОСТ 2.104. Сформированный документ следует сохранить в файл под уникальным именем, включающим фамилию и номер группы.

14.1.3 Настройка параметров текстового документа выполняется следующим образом:

- 1) после перехода в колонтитулы документа производится установка параметров основной надписи:
 - указывается фамилия разработчика работы в позиции *Выполнил*;
 - записывается фамилия преподавателя в позиции *Проверил*;
 - задается шифр работы – инициалы и первый символ фамилии вместе со следующей группой цифр *00.00.000 ПЗ*;
 - в позиции организация указывается номер учебной группы;
 - в обозначении задается: отчет по лабораторной работе № 14;

2) контролируется в параметрах страницы следующее: размер и ориентация формата листа, а также отступы текста от края листа;

3) проверяется состав стилей работы, при необходимости изменяются тип шрифта, его высота, межстрочный интервал и отступы;

4) заполнение полей данных свойств документа производится выбором раздела *«Файл/Свойства/Дополнительные свойства»*.

14.1.4 Формирование структуры документа выполняется записью наименований разделов и подразделов в документе. Первая страница оставляется пока пустой, в ней устанавливается маркер начала новой страницы. Для каждого наименования раздела устанавливается стиль *«Заголовок 1»*, а для наименования подраздела – *«Заголовок 2»*.

14.1.5 В качестве содержания первого раздела переписывается текст цели работы из методических рекомендаций. Высота шрифта должна быть 14 пт. Абзац должен иметь отступ на красную строку размером 1,25 см и полуторный межстрочный интервал. Эти требования распространяются на основной текст документа.

14.1.6 Ответ на теоретическое задание записывается в первом подразделе второго раздела. Текст ответа должен выполняться в обезличенном стиле изложения материала. При копировании в текст данных из других файлов или интернета рекомендуется использовать специальную вставку, которая для вставляемого текста автоматически назначает формат текста того абзаца, в который копируются данные.

14.1.7 Отображение заданной формулы выполняется в теле второго подраздела второго раздела. Заданное математическое выражение создается с помощью редактора формул, входящего в состав текстового редактора. Ниже формулы выполняется пояснение всех использованных в формуле обозначений ее составляющих. При этом следует использовать вставку математических и греческих символов, а также выполнение надстрочного и подстрочного элементов средствами текстового редактора.

14.1.8 Создание заданной таблицы выполняется в третьем подразделе второго раздела документа. При вставке таблицы следует использовать средства *«Вставить таблицу»* или *«Нарисовать таблицу»*. Таблица должна иметь наименование с номером, выполняемым над таблицей (рекомендуется кликнуть по таблице правой кнопкой мыши и в распахнувшемся меню выбрать пункт *«Вставить название»*). Текст данных таблицы может быть выполнен шрифтом высотой 12 пт с одинарным межстрочным интервалом. Таблица заполняется данными реально существующих объектов, заданного типа.

14.1.9 Вставка использованных источников выполняется при ответе на теоретический вопрос, а также при указании источника данных таблицы. При этом рекомендуется использовать встроенный редактор списка литературы. Ссылка в тексте на источник оформляется в виде его номера, заключенного в квадратные скобки. Сам список должен соответствовать требованиям ГОСТ 7.1.

14.1.10 Добавление содержания выполняется на первой странице документа. При этом следует использовать инструмент *«Оглавление»* из меню

(ленты) «Ссылки». В содержании должны отображаться заголовки первого и второго уровней. При этом номер страницы элемента устанавливается автоматически и является ссылкой.

14.1.11 Проверка правописания выполняется после завершения формирования текста документа. При необходимости для фрагмента текста, выполненных на иностранном языке, указывается язык с помощью раздела «Язык» выпадающего меню. Следует максимально использовать возможности текстового редактора при проверке правописания.

14.1.12 Формирование электронного документа выполняется отправкой его на печать на установленный виртуальный принтер. Результатом является pdf- или mdi-файл электронного варианта документа.

14.2 Содержание отчета

Отчет по работе № 14 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- 1) титульный лист отчета;
- 2) содержание, сформированное автоматически и имеющее ссылки;
- 3) раздел 1 «Цель лабораторной работы»;
- 4) раздел 2 «Основная часть», включающая:
 - подраздел 2.1 «Ответ на теоретический вопрос»;
 - подраздел 2.2 «Вставка формулы»;
 - подраздел 2.3 «Работа с таблицей»;
- 5) список использованных источников.

Контрольные вопросы

- 1 Пояснить назначение и функции текстовых редакторов.
- 2 Описать структуру интерфейса текстового редактора.
- 3 Рассказать о способах создания нового документа.
- 4 Показать, каким образом выполняется настройка параметров страниц в текстовых документах.
- 5 Описать простановку номеров страниц в документе.
- 6 Перечислить способы добавления рисунка в документ.
- 7 Пояснить назначение и использование стилей текста.
- 8 Показать способы записи формул в документе.
- 9 Пояснить способы формирования в документе таблиц.
- 10 Рассказать о проверке правописания в редакторе.
- 11 Объяснить, каким образом автоматически сформировать содержание текстового документа.
- 12 Показать, каким образом задается наименование таблиц.
- 13 Перечислить основные параметры абзацев текста.
- 14 Пояснить, как выполняется настройка параметров списков.

- 15 Перечислить способы импорта текстовых данных.
- 16 Показать, каким образом формируется список литературы.
- 17 Рассказать о способах защиты документов от несанкционированного доступа.
- 18 Показать способы настройки колонтитулов документа.
- 19 Пояснить, каким образом реализуется формирование границ и фона документа.
- 20 Показать отображение исправлений в версиях документов.
- 21 Показать, каким образом настраивается отображение документа.

15 Лабораторная работа № 15. Изучение редактора AutoCAD

Цель работы:

- изучение среды графического редактора AutoCAD;
- получение практических навыков по созданию электронных чертежей с изображением схем алгоритмов и электрических схем.

Задание

Заданием является разработка двух чертежей в среде AutoCAD.

- 1 Схема алгоритма заданного численного метода, выполненная согласно требованиям ГОСТ 19.701–90.
- 2 Схема электрическая принципиальная заданного типа операционного усилителя.

15.1 Ход выполнения работы

15.1.1 Загрузка системы компьютерной графики AutoCAD. Рекомендуется сразу переключить интерфейс пользователя AutoCAD на классический вариант, если это не выполнено по умолчанию по [11, 13].

15.1.2 Следует создать новый чертеж на основе прототипа (шаблона) с именем *eskd_a4.dwt* для схемы электрической принципиальной. Пустой чертеж имеет набор блоков для изображения схем электрических и алгоритмов, а также основную надпись в соответствии с ГОСТ 2.104. Следует сразу сохранить чертеж в файле с именем, в котором используется фамилия, номер группы и символы ЕЗ.

15.1.3 Заполнение полей блока основной надписи выполняется в специальном диалоговом окне, вызванном кликом мышью по блоку или командой **ATTREDIT**, где указывается:

- 1) фамилия разработчика работы в позиции «*Выполнил*»;
- 2) фамилия преподавателя в позиции «*Проверил*»;
- 3) шифр работы – инициалы и первый символ фамилии вместе со следующей группой цифр *00.00.000 ЭЗ*;
- 4) в позиции организация указывается номер учебной группы;
- 5) в обозначении задается: Отчет по лабораторной работе № 15. Схема электрическая принципиальная.

15.1.4 Заполняются данные свойств документа выбором пункта «Файл/Свойства».

15.1.5 Контроль параметров электронного чертежа выполняется так:

1) командой **LIMITS** или пунктом «Формат/Границы (*Format/Limits*)» проверяются заданные размеры границ чертежа;

2) командой **UNITS** или выбором пункта «Формат/Единицы *Format/Units*» в окне диалога контролируется установка линейных единиц измерения в миллиметрах, а угловых – в градусах с точностью до единицы;

3) командой **GRID** устанавливаются размеры сетки экрана в 5 мм;

4) командой **SNAP** задается в окне шаг курсора в 2,5 мм;

5) командой **OSNAP** выбираются в диалоговом окне варианты реализации объектной привязки;

6) командой **LAYER** или пунктом «Формат/Слой» определяется состав слоёв чертежа, их наименование, а также тип, цвет и толщина линий, используемых на них для отображения примитивов;

7) рекомендуется установить белый цвет фона рисования для удобства экспорта рисунков из AutoCAD в другие приложения.

15.1.6 Рисование схемы электрической принципиальной выполняется с помощью команд меню «Рисование (*Draw*)» и вставки готовых элементов (конденсаторов, резисторов, диодов и т. п.) из раздела «Вставка/Блок». Изначально рекомендуется отключить объектную привязку, оставив только привязку к координатной сетке, и выполнить изображение корпуса операционного усилителя. Далее, включив объектную привязку клавишей F3, дорисовать корректирующие цепи. При необходимости используются команды редактирования изображений из меню «Редактировать (*Edit*)».

15.1.7 Требуемые надписи на чертеже выполняются с помощью команд однострочного редактора **EDIT** или **DEDIT** или многострочного текста **MEDIT**. Параметры шрифта отображаемого текста можно изменять в окне диалога командой **STYLE** или пунктом «Формат/Стиль текста». Изменение надписей блоков можно выполнять в диалоговом окне, вызываемом кликом по блоку или командой **ATTREDIT**. Среда AutoCAD выполняет проверку правописания в многострочном тексте.

15.1.8 Выполняется создание электронного *pdf*- или *mdi*- чертежа его распечаткой на установленный в системе виртуальный принтер.

15.1.9 Создание нового чертежа схемы алгоритма на основе прототипа *eskd_a4.dwt*. Следует сохранить чертеж в файле с именем, в котором используется фамилия, номер группы и символы **D1**.

15.1.10 Заполнение полей блока основной надписи выполняется аналогично п. 15.1.4 командой **ATTREDIT**, но с отличиями:

1) шифр работы – инициалы и первый символ фамилии вместе с группой цифр *00.00.000 D1*;

2) обозначение: Отчет по лабораторной работе № 15. Схема алгоритма.

15.1.11 Заполнение свойств чертежа и настройка параметров выполняется аналогично пп. 15.1.4 и 15.1.5.

15.1.12 Рисование схемы алгоритма выполняется аналогично п. 15.1.11 с помощью команд меню «*Рисование (Draw)*» и вставки готовых блоков (терминатора, процесса, данных, циклов и т. п.) из раздела «*Вставка/Блок*». Изначально рекомендуется провести центральную вертикальную ось символов. Далее на нее устанавливаются блоки символов схемы. При необходимости используются команды редактирования из меню «*Редактировать (Edit)*».

15.1.13 Надписи на схеме выполняются аналогично п. 15.1.7.

15.1.14 Создание электронного *pdf*- или *mdi*-схемы производится ее печатью на установленный виртуальный принтер.

15.2 Содержание отчета

Отчет по работе № 15 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- текст индивидуального задания;
- чертеж с заданной схемой алгоритма;
- чертеж схемы электрической принципиальной.

Контрольные вопросы

- 1 Указать цели и функции графических редакторов.
- 2 Описать структуру интерфейса пользователя среды AutoCAD.
- 3 Перечислить команды рисования примитивов AutoCAD.
- 4 Назвать основные команды редактирования AutoCAD.
- 5 Пояснить настройку параметров чертежей в AutoCAD.
- 6 Рассказать о создании и вставки блоков в AutoCAD.
- 7 Объяснить назначение и свойства атрибутов блоков.
- 8 Пояснить настройку параметра стиля текста AutoCAD.
- 9 Объяснить назначение и параметры объектной привязки.
- 10 Описать способы экспорта и импорта графических данных из AutoCAD в другие приложения.
- 11 Рассказать о командах установки размеров AutoCAD.
- 12 Пояснить настройку параметров размеров AutoCAD.

16 Лабораторная работа № 16. Среда компьютерной математики Mathcad

Цель работы:

- изучение среды математических расчетов Mathcad;
- получение практических навыков проведения типовых математических расчетов и оформления документов в среде Mathcad.

Задание

Необходимо создать *xmcd*-документ, в котором следует:

- 1) решить заданное нелинейное уравнение;
- 2) построить график функции нелинейного уравнения;
- 3) найти корни заданного алгебраического уравнения;
- 4) решить заданную систему алгебраических уравнений;
- 5) выполнить интерполяцию в указанной точке и аппроксимацию выражения таблично заданной функции;
- 6) найти значения определенного интеграла в указанных пределах, а также первой и второй производных для заданной функции;
- 7) решить обыкновенное дифференциальное уравнение (ОДУ);
- 8) найти локальный минимум и построить график заданной двухпараметрической функции;
- 9) выполнить следующие операции над массивами A и B :
 - вычислить определитель матрицы A ;
 - выделить второй столбец из матрицы A ;
 - найти обратную матрицу A ;
 - выполнить транспонирование матрицы A ;
 - найти скалярное произведение матрицы A на вектор B ;
 - вычислить сумму элементов вектора B ;
 - получить вектор собственных значений матрицы A ;
 - определить максимальное и минимальное значения модуля полученного вектора собственных значений;
 - рассчитать след матрицы A .

16.1 Ход выполнения работы

16.1.1 Загрузка среды компьютерной математики Mathcad выполняется из стартового меню или файловым менеджером для портативной версии по [15]. Следует сразу создать файл документа Mathcad, имя которого должно включать фамилию и группу студента.

16.1.2 Заполнить поля свойств электронного документа с помощью раздела «Файл/Свойства».

16.1.3 Выполняется настройка следующих параметров страниц документа Mathcad с помощью раздела «Файл/Параметры страницы»: формата и ориентации листа, отступы данных от края.

16.1.4 Решение уравнения выполняется следующим образом:

- 1) записывается заданное исходное уравнение в виде функции;
- 2) в виде переменных задаются границы поиска корня;
- 3) корень определяется с помощью функции *root*.

16.1.5 Для построения графика производится следующее:

- 1) объявляется ранжированная переменная аргумента в заданном диапазоне;
- 2) вычисляется значение переменной функции с применением функции, определенной в (1) из п. 16.1.4, аргумент которой берется из (1);
- 3) выводится на экран таблица переменных, найденных в (1) и (2);
- 4) с помощью пункта «Вставка/График/График X-Y» (@) строится декартовский график, где по оси X задается переменная (1), а по Y – (2).

16.1.6 Решение алгебраического уравнения выполняется так:

- 1) создается вектор-столбец заданных коэффициентов полинома;
- 2) с помощью функции *polyroots* с аргументом, заданным в (1), определяется и отображается решение алгебраического уравнения.

16.1.7 Для решения системы линейных алгебраических уравнений (СЛАУ) задается следующая последовательность операций:

- 1) определяется переменная-матрица заданных коэффициентов;
- 2) создается переменная-столбец заданных свободных членов;
- 3) определяется и отображается вектор решения как матричное произведение обратной матрицы переменной (1) на вектор из (2).

16.1.8 Интерполяция и аппроксимация табличной функции производится в следующем порядке:

- 1) объявляется переменная матрица из двух столбцов с заданными таблично данными, где первый столбец – аргумент, второй – функция;
- 2) функцией *cspline* находится вектор сплайна по данным из (1);
- 3) функцией *interp* на основе данных из (2), (1) и заданной точки выполняется интерполяция и отображается ее результат;
- 4) аппроксимация выполняется функцией *regress* на основе данных из (1) для случая параболической функции (второго порядка).

16.1.9 Поиск интеграла и производной реализуются так:

- 1) используется шаблон интеграла, вызываемый символом $\&$, в котором указывается подынтегральная функция (1) из п. 16.1.4;
- 2) задается переменная – точка аргумента для поиска производных;
- 3) для первой производной применяется шаблон, вызываемый символом $\?$, в котором указывается функция (1) из п. 16.1.4 с данными (2);
- 4) для расчета второй и третьей производных вызывается комбинацией клавиш **Ctrl + Shift + ?** специальный шаблон. В нем указывается функция (1) из п. 16.1.4 с данными (2) и число степени.

16.1.10 Решение ОДУ выполняется следующим образом:

- 1) описывается двухпараметрическая функция ОДУ задания;
- 2) объявляются две переменные заданных начальных условий;
- 3) создается переменная конца интегрирования решения;

- 4) указывается переменная для числа точек решения;
- 5) решение ОДУ выполняется функцией *Rkadapt*;
- 6) отображается таблица результатов решения;
- 7) строится график решения аналогично п.16.1.5.

16.1.11 Поиск минимума и построение графика заданной двухпараметрической функции реализуется следующими операциями:

- 1) определяется выражение двухпараметрической функции;
- 2) объявляются две переменные начального поиска;
- 3) функцией Minimize с параметрами (1) и (2) находится решение;
- 4) график для функции (1) строится выбором пункта «Вставка/График/График поверхности» (*Ctrl + 2*).

16.1.12 Матричные операции выполняются над заданными данными с помощью специальных шаблонов с панели инструментов *Матрица*.

16.1.13 Выполняется создание электронной версии формата *pdf* или *mdi* для *Mathcad* документа и экспорт данных в отчет.

16.2 Содержание отчета

Отчет по работе № 16 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- индивидуальное задание с исходными данными;
- текст созданного документа *Mathcad* с решением задания.

Контрольные вопросы

- 1 Пояснить назначение и функции системы *Mathcad*.
- 2 Назвать основные правила построения имен в *Mathcad*.
- 3 Описать состав главного меню окна *Mathcad*.
- 4 Пояснить поиск корней алгебраических уравнений в *Mathcad*.
- 5 Объяснить формат описания функций в среде *Mathcad*.
- 6 Описать решение нелинейного уравнения в среде *Mathcad*.
- 7 Рассказать о способах решения СЛАУ в среде *Mathcad*.
- 8 Пояснить построение графиков функций в среде *Mathcad*.
- 9 Пояснить решение ОДУ в среде *Mathcad*.
- 10 Указать методы интерполяции функции в среде *Mathcad*.
- 11 Описать способы аппроксимации функций в *Mathcad*.
- 12 Пояснить поиск определенного интеграла в среде *Mathcad*.
- 13 Описать расчет производных функций в среде *Mathcad*.
- 14 Перечислить матричные операции в среде *Mathcad*.

17 Лабораторная работа № 17. Математическая система MATLAB

Цель работы:

- изучение среды математической системы MATLAB;
- получение практических навыков проведения в среде MATLAB научно-технических расчетов.

Задание

Следует сформировать *m*-файл, который реализует в среде MATLAB следующие операции:

- 1) для заданной матрицы *A* найти определитель, обратную матрицу, вектор собственных значений, его максимальное и минимальное значение;
- 2) решить СЛАУ, заданную в виде матриц *A* и вектора *B*;
- 3) решить нелинейное уравнение $f(x) = 0$, заданное в виде функции;
- 4) построить график функции $y = f(x)$ на заданном интервале;
- 5) вычислить определенный интеграл для заданной функции;
- 6) найти значение первой и второй производных для функции в заданных точках *a* и *b*;
- 7) решить алгебраическое уравнение по заданному полиному;
- 8) с помощью интерполяции найти значение таблично заданной функции в указанной точке;
- 9) аппроксимировать таблично заданную функцию параболой;
- 10) численно решить заданное ОДУ с построением графика решения;
- 11) построить трехмерный график двухпараметрической функции;
- 12) определить локальный минимум двухпараметрической функции.

17.1 Ход выполнения работы

17.1.1 Запуск среды компьютерной математики MATLAB выполняется в соответствии с [6]. После раскрытия командного MATLAB следует указать текущую папку, в которой будет выполняться работа.

17.1.2 Создание *m*-файла (*script*-файла) для расчетов выполняется в командном окне нажатием клавиш **Ctrl+N**. При этом раскроется окно редактора, в котором отобразится заготовка файла. Следует задать имя *m*-файла, в которое включается фамилия и группу, набранные латиницей.

17.1.3 В начале *m*-файла вызываются команды очистки командного окна *clc* и удаляются данные предыдущих расчетов вызовом *clear all*.

17.1.4 Матричные операции выполняются следующим образом:

- 1) в переменные *A* и *B* задаются значения варианта задания;
- 2) выполняется расчет определителя матрицы функцией *det*;
- 3) обратная матрица *A* вычисляется функцией *inv*;
- 4) след матрицы вычисляется функцией *trace*;
- 5) собственные значения *A* находятся функцией *eig*;

- 6) выделяется диагональ из (4) в вектор функцией *diag*;
- 7) с помощью функции *real* извлекается действительная часть элементов вектора, а функцией *abs* получают модули элементов;
- 8) *max* определяет максимальный элемент, а *min* – минимальный.

17.1.5 Решение СЛАУ выполняется двумя способами:

- 1) классическим выражением $X=A\backslash B$;
- 2) с помощью функции LU-разложения *lu*.

17.1.6 Решение заданного нелинейного уравнения производится вызовом функции *fzero*, где первый параметр – строка уравнения.

17.1.7 Построение декартового графика заданной функции выполняется в следующей последовательности:

- 1) создается новый *m*-файл функции для заданного выражения;
- 2) задается вектор аргумента в указанном диапазоне изменения;
- 3) рассчитывается значение переменной функции на основе (1);
- 4) функцией *plot* строится двухмерный график в декартовой системе координат с включением сетки командой *grid on*.

17.1.8 Поиск определенного интервала и производных функции производится следующим образом:

- 1) значение определенного интервала находится функцией *quad*;
- 2) первая и вторая производные вычисляются на основе данных (2) и (3) из п. 17.1.7 с помощью функции *diff*.

17.1.9 Заданное алгебраическое уравнение решается в порядке:

- 1) определяется вектор с коэффициентами полинома задания;
- 2) с помощью функции *roots* получается вектор решения.

17.1.10 Решение ОДУ выполняется следующим образом:

- 1) создается новый файл-функция с заданным выражением ОДУ;
- 2) функцией *ode45* решается ОДУ на заданном интервале;
- 3) функцией *plot* строится график решения аналогично п.17.1.7.

17.1.11 Интерполяция и аппроксимация таблично заданных функций выполняется в следующем порядке:

- 1) табличная функция задается в виде двух векторов;
- 2) функцией *interp1* с параметром *cubic* (кубическая) выполняется интерполяция в заданной точке;
- 3) аппроксимация для векторов из (1) выполняется функцией *polyfit*.

17.1.12 Поиск минимума и построение трехмерного графика двухпараметрической функции производится в следующем порядке:

- 1) формируется *m*-файл функции с заданным выражением;
- 2) функцией *meshgrid* задается диапазон изменения переменных;
- 3) рассчитывается значение функции на основе (1) и данных (2);
- 4) трехмерный график выполняется с помощью функции *plot3*;
- 5) функцией *fminsearch* находится локальный минимум;
- 6) по полученным в (5) данным определяется значение функции.

17.1.13 Отлаживается созданный *m*-файл запуском клавишей *F5*.

17.1.14 Переносятся данные из командного окна и окон графиков в файл электронного отчета через буфер обмена.

17.2 Содержание отчета

Отчет по работе № 17 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- текст индивидуального задания;
- текст спроектированного *m*-файла решения задания;
- результаты решения задач задания в командном окне и копии изображений полученных графиков.

Контрольные вопросы

- 1 Указать назначение и функции системы MATLAB.
- 2 Описать состав интерфейса среды MATLAB.
- 3 Пояснить решение алгебраических уравнений в MATLAB.
- 4 Описать поиск корня нелинейного уравнения в MATLAB.
- 5 Перечислить способы решения СЛАУ в MATLAB.
- 6 Перечислить функции построения графиков в MATLAB.
- 7 Описать последовательность решения ОДУ в MATLAB.
- 8 Выполнить интерполяцию табличной функции в MATLAB.
- 9 Перечислить основные функции аппроксимации в MATLAB.
- 10 Описать поиск определенного интеграла в MATLAB.
- 11 Объяснить расчет производных функций в MATLAB.
- 12 Описать поиск локального минимума функции в MATLAB.

18 Лабораторная работа № 18. Численное решение уравнений

Цель работы:

- изучение численных методов решения нелинейных уравнений (НУ) и систем нелинейных уравнений (СНУ);
- получение практических навыков программной реализации алгоритмов численного решения нелинейных уравнений на языке C++.

Задание

Разработать схему алгоритма и написать программу-приложение решения нелинейного уравнения в соответствии с одним из следующих методов: серединного деления (дихотомии), касательных (Ньютона), хорд, простых итераций, секущих, равномерного поиска, Мюллера или Ридерса.

18.1 *Ход выполнения работы*

18.1.1 Разработка алгоритма решения НУ выполняется после анализа источников [3, 6–8] в соответствии с требованиями ГОСТ 19.701.

18.1.2 Выполняется запуск среды визуального программирования языка C++, установленной на ПК. В окне IDE среды создается проект приложения с графическим интерфейсом пользователя (GUI).

18.1.3 Параметры окна приложения задаются аналогично п. 11.1.2.

18.1.4 Формирование интерфейса приложения выполняется с использованием следующих визуальных компонентов:

- 1) двух текстовых полей для задания интервала поиска решения;
- 2) текстового поля с указанием точности решения;
- 3) при необходимости добавляются поля текста с описанием значения переменных коэффициентов решаемого уравнения;
- 4) поле для вывода результата решения (корня);
- 5) поле с данными о числе итераций при поиске решения;
- 6) поле с данными о полученной точности;
- 7) текстовые надписи (метки) для всех вышеуказанных полей;
- 8) три кнопки управления приложением: расчета, сведений о программе и выхода из программы.

18.1.5 Настройка параметров компонентов интерфейса выполняется аналогично п. 11.1.3.

18.1.6 Описание данных приложения для решения НУ выполняется в *src*-файле модуля следующим образом:

- 1) при необходимости определяется константа максимально допустимого числа итераций (методы Ньютона, простых итераций и др.);
- 2) объявляются две переменные типа *double* границы решения;
- 3) задается переменная типа *double* для описания точности решения;
- 4) описывается переменные типа *double* корня и точности решения;
- 5) при необходимости объявляются переменные коэффициентов решаемого уравнения типа *double*;
- 6) описывается целочисленная переменная числа итераций;
- 7) определяется однопараметрическая функция, возвращающая результат типа *double*, которая описывает заданное решаемое уравнение.

18.1.7 Программная реализация разработанного алгоритма, который выполняется при нажатии кнопки решения, производится в следующей последовательности:

1) считываются исходные данные из текстовых полей с преобразованием строкового типа в тип *double* для переменных:

- интервала поиска с проверкой корректности их задания;
- значения требуемой точности решения;
- при необходимости коэффициенты выражения НУ (5)

из п. 18.1.6;

2) инициализируется первоначальное значение корня НУ;

3) организуется цикл уточнения значения корня в соответствии со схемой метода. В теле цикла инкрементируется счетчик итераций. Цикл продолжается до тех пор, пока интервал поиска не станет равен заданной точности или число итераций достигнет максимально допустимого;

4) выполняется расчет значения функции для полученного корня;

5) значения корня, полученной точности и числа итераций преобразуются в текст и выводятся в соответствующие поля.

18.1.8 Вывод сообщения о программе выполняется в обработчике события нажатия одноименной кнопки аналогично п. 11.1.11.

18.1.9 Выход из программы выполняется в обработчике события нажатия кнопки «Выход» вызовом метода закрытия формы приложения.

18.1.10 Выполняется отладка и тестирование разработанной программы с использованием среды Mathcad или MATLAB.

18.1.11 Создается загрузочный модуль и выполняется решение заданного НУ с помощью созданной программы.

18.2 Содержание отчета

Отчет по работе № 18 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- заданный метод решения и исходное НУ с интервалом поиска;
- схема алгоритма заданного численного решения НУ;
- тексты проекта созданной программы: *h*- и *сpp*-файлы модуля формы, главный файл проекта, а также размер загрузочного файла;
- результат расчета корня НУ (копия экрана формы с результатами расчета) и тестирование в Mathcad или MATLAB.

Контрольные вопросы

- 1 Назвать методы численного решения нелинейных уравнений.
- 2 Указать исходные данные для численного решения НУ.
- 3 Перечислить характеристики численных методов решения НУ.
- 4 Показать геометрический смысл метода серединного деления.
- 5 Объяснить геометрический смысл метода хорд.
- 6 Показать геометрический смысл метода Ньютона.
- 7 Показать геометрический смысл метода простых итераций.
- 8 Объяснить геометрический смысл метода секущих.
- 9 Показать геометрический смысл метода Мюллера.
- 10 Объяснить суть метода Ридерса.
- 11 Пояснить реализацию процедуры отделения корней.

19 Лабораторная работа № 19. Методы численного интегрирования обыкновенных дифференциальных уравнений

Цель работы:

- изучение численных методов решения обыкновенных дифференциальных уравнений (ОДУ);
- получение практических навыков программной реализации алгоритмов численного интегрирования ОДУ в среде языка C++.

Задание

Разработать схему алгоритма и написать программу-приложение для решения ОДУ, определенного в соответствии с заданным численным методом. Результаты решения ОДУ должны отображаться в таблице и на графике.

19.1 *Ход выполнения работы*

19.1.1 Проанализировав источники [3, 7, 8, 10], разрабатывается заданный алгоритм решения ОДУ в соответствии с ГОСТ 19.701.

19.1.2 Запускается среда визуального программирования языка C++, в которой создается проект приложения с GUI-интерфейсом пользователя.

19.1.3 Параметры окна приложения задаются аналогично п.11.1.2.

19.1.4 Интерфейс главной формы приложения выполняется с использованием ниже перечисленных визуальных компонентов:

- 1) трех текстовых полей задания начальных условий и интервала поиска решения ОДУ;
- 2) текстового поля с указанием точности или шага интегрирования;
- 3) текстового поля с числом точек получаемого решения;
- 4) при необходимости добавляются поля текста с описанием значения переменных коэффициентов решаемого ОДУ;
- 5) полоса отображения процента выполнения решения;
- 6) текстовые надписи (метки) для всех вышеуказанных полей;
- 7) четыре кнопки управления приложением: расчета, просмотра результатов, сведений о программе и выхода из программы.

19.1.5 В проект добавляется новая форма отображения результатов решения ОДУ аналогично п. 12.1.4. На форму помещаются компоненты:

- 1) текстовая таблица для вывода результата решения ОДУ;
- 2) график-диаграмма с настройкой свойств аналогично п. 11.1.5;
- 3) текстовые метки подписи графика и таблицы;
- 4) три кнопки управления окном: сохранения таблицы решения в текстовый файл, сохранения графика и закрытия окна;
- 5) два невидимых компонента окон диалогов ввода имени текстового и графического файлов для сохранения данных решения.

19.1.6 Настройка параметров компонентов интерфейсов для двух форм выполняется аналогично пп. 11.1.3–11.1.5.

19.1.7 В *cpp*-файле модуля главного окна директивой **#include** подключается модуль окна отображения результатов.

19.1.8 Описание глобальных данных приложения выполняется в *cpp*-файле модуля главной формы следующим образом:

1) объявляются три переменные типа **double** для данных начальных условий и границы решения;

2) задаются переменные типа **double** для точности и (или) шага интегрирования;

3) задается целочисленная переменная числа расчетных точек;

4) описываются два указателя X и Y на тип **double** для динамических векторов результатов решения;

5) при необходимости объявляются переменные коэффициентов решаемого ОДУ типа **double**;

6) определяется двухпараметрическая функция, возвращающая результат типа **double**, описывающая решаемое ОДУ.

19.1.9 Программирование разработанного алгоритма выполняется в обработчике нажатия кнопки решения следующим образом:

1) создаются дополнительные локальные переменные, необходимые для реализации алгоритма: число точек интегрирования на шаг сохранения, текущие и предыдущие значения точки решения (или векторы для них), значение текущей точности и др.;

2) берутся исходные данные из текстовых полей главной формы с преобразованием строкового типа в тип **double** для переменных:

- начальных условий и интервала поиска;
- значения точности решения и (или) шага интегрирования;
- число точек сохранения решения;
- при необходимости коэффициенты выражения ОДУ;

3) оператором **new** создаются динамические векторы из указателей;

4) задаются первые значения векторов начальными условиями;

5) иницируется нулем полоса процесса решения, а максимальное значение определяется числом расчетных точек;

6) программируется основной цикл расчета текущей точки решения;

7) вычисляется решение на основе заданного метода;

8) сохраняется значение в элементы вектора;

9) в положение полосы отображения процесса решения устанавливается номер текущей точки;

10) по окончании решения выдается на экран информационное окно с соответствующим сообщением.

19.1.10 Отображение результатов расчета выполняется в обработчике события нажатия соответствующей кнопки, где в теле метода обработчика вызывается метода модального отображения формы результатов.

19.1.11 Вывод информации о программе выполняется в обработчике события нажатия одноименной кнопки по аналогии с п. 11.1.11.

19.1.12 Выход из программы выполняется в обработчике события нажатия кнопки «Выход» следующим образом:

- 1) очищаются два динамических вектора оператором *delete*;
- 2) вызывается метод закрытия главной формы приложения.

19.1.13 В *cpp*-файле модуля формы результатов подключаются директивой *#include* модуль *fstream*, а также делаются доступными глобальные данные. Для этого копируются с главной формы объявления переменных, выполненные в (3) и (4) п.19.1.8 с описателем *extern*.

19.1.14 Программирование события отображения окна результатов выполняется в теле обработчика отображения формы следующим образом:

- 1) в начале тела в свойствах компонента таблицы задается число ее строк на единицу большим, чем число точек решения;
- 2) в первой строке таблицы отображаются подписи столбцов;
- 3) задается цикл *for* для вывода всех рассчитанных точек в поля таблицы, в теле которого выполняются операции:
 - в первом столбце отображается с преобразованием номер точки;
 - во втором столбце выводится текущее значение вектора аргумента с преобразованием вещественных данных в строковый тип;
 - в третьем столбце показывается значение текущей точки вектора функции с преобразованием данных *double* в строковый тип;
- 4) очищается серия данных графика;
- 5) задается цикл *for* для вывода всех точек решения, в теле которого к серии данных графика добавляется текущая точка функции.

19.1.15 Программирование событий формы результатов выполняется следующим образом:

- 1) обработчик нажатия кнопки сохранения таблицы в файл программируется аналогично п. 12.1.10;
- 2) обработчик нажатия кнопки записи графика в файл выполняется по аналогии с п. 12.1.11;
- 3) обработчик нажатия кнопки закрытия окна аналогичен п. 11.1.10.

19.1.16 Производится отладка и тестирование разработанной программы с использованием среды Mathcad или MATLAB.

19.1.17 Создается загрузочный модуль приложения и решается заданное ОДУ в разработанной программе.

19.2 Содержание отчета

Отчет по работе № 19 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;

- задача исследования – исходное ОДУ и численный метод;
- схема алгоритма решения ОДУ заданным методом;
- тексты программы решения ОДУ: главный *сpp*-файл проекты, *h*- и *сpp*-файлы двух модулей, а также размер *exe*-файла;
- таблица и график решения заданного ОДУ;
- результаты проверки решения ОДУ (тест), выполненные в среде Mathcad или MATLAB.

Контрольные вопросы

- 1 Перечислить группы методов для численного решения ОДУ.
- 2 Указать, как определяется точность метода решения ОДУ.
- 3 Пояснить геометрический смысл одношаговых методов численного интегрирования ОДУ.
- 4 Объяснить геометрический смысл многошаговых методов численного интегрирования ОДУ.
- 5 Пояснить область назначения неявных методов численного решения ОДУ.
- 6 Описать алгоритм неявных методов численного решения ОДУ.
- 7 Пояснить выбор величины шага интегрирования при решении.
- 8 Объяснить алгоритм автоматической коррекции величины шага интегрирования в явных одношаговых методах решения ОДУ.
- 9 Пояснить алгоритм методов прогноза и коррекции для ОДУ.

20 Лабораторная работа № 20. Интерполяция табличных функций

Цель работы:

- изучение методов интерполяции таблично заданных функций;
- получение практических навыков визуального программирования задач интерполяции в среде языка C++.

Задание

Следует разработать программу с GUI для интерполяции заданной табличной функции. Необходимо определить с ее помощью значение функции в указанной точке на основе одного из двух методов: линейной интерполяции или полинома Лагранжа.

20.1 Ход выполнения работы

20.1.1 После анализа [3, 7, 8, 10] разрабатывается алгоритм заданного метода интерполяции в соответствии с ГОСТ 19.701.

20.1.2 Загружается среда визуального программирования языка C++, где создается проект приложения с GUI-интерфейсом.

20.1.3 Параметры окна приложения задаются аналогично п.11.1.3.

20.1.4 Формируется интерфейс формы приложения на основе:

- 1) компонентов двух текстовых полей исходных данных: числа узлов таблицы и точки поиска решения;
- 2) установкой двойного переключателя для изменения числа узлов;
- 3) компонента текстового поля отображения решения;
- 4) компонента таблицы строк с данными заданной функции;
- 5) размещения двух невизуальных компонентов диалоговых окон выбора имени файла при вводе и выводе данных таблицы;
- 6) помещения четырех компонентов кнопок: «Загрузить данные», «Сохранить данные», «Интерполяция», «О программе» и «Выход»;
- 7) компонентов текстовых надписей (меток) для всех полей.

20.1.5 Настройка параметров установленных компонентов интерфейса производится аналогично пп. 11.1.3–11.1.5. Дополнительно связывается между собой двойной переключатель и поле числа узлов.

20.1.6 В *cpp*-файле модуля формы подключается модуль *fstream*.

20.1.7 Описание следующих глобальных данных приложения выполняется в *cpp*-файле модуля формы следующим образом:

- 1) объявляются две переменные типа *double* для заданной точки поиска и полученного решения;
- 2) объявляется целочисленная переменная числа узлов таблицы;
- 3) описываются два указателя *X* и *Y* на тип *double* для динамических векторов зависимой и независимой переменных функции.

20.1.8 Следует выполнить описание специального метода перерисовки таблицы данных при изменении числа узлов в порядке:

- 1) в *h*-файле внутри секции *public* класса формы описывается заголовок метода перерисовки таблицы, как *void*-функции без параметров;
- 2) в *cpp*-файл формы копируется заголовок метода перерисовки таблицы с указанием перед именем метода через *::* имени класса формы;
- 3) под заголовком определяется следующий состав тела метода:
 - из поля двойного переключателя считывается значение в переменную числа узлов;
 - оператором *new* создаются два динамических вектора определенного выше размера для указателей *X* и *Y*;
 - задается число строк компонента таблицы на единицу больше, чем число считанное узлов;
 - в первой строке отображаются подписи колонок таблицы;
 - организуется цикл *for* отображения данных таблицы;
 - в теле цикла в первом столбце выводится с преобразованием номер узла, во втором – значение аргумента, в третьем – функции.

20.1.9 Описывается специальный метода чтения данных из строк таблицы следующим образом:

- 1) в *h*-файле модуля внутри секции *public* класса объявляется заголовок метода чтения таблицы, как *void*-функции без параметров;

2) в *cpp*-файл модуля копируется заголовок метода чтения таблицы с указанием перед именем метода через `::` имени класса формы;

3) под заголовком программируется следующий состав тела метода:

- из поля двойного переключателя передается значение числа узлов в глобальную переменную;
- организуется цикл *for* чтения данных из компонента таблицы;
- в теле цикла из первого столбца считывается с преобразованием значения аргумента узла, а из третьего – соответствующее значение функции в соответствующие динамические вектора.

20.1.10 Реализация разработанного в п. 20.1.1 алгоритма выполняется в обработчике нажатия кнопки «*Интерполяция*» следующим образом:

- 1) объявляются локальные переменные;
- 2) вызывается созданный метод чтения значений данных таблицы;
- 3) из поля чтения аргумента определяется точка поиска решения;
- 4) программируется алгоритм интерполяции функции;
- 5) полученный результат после преобразования в строку выводится в текстовое поле решения.

20.1.11 Программируется обработчик нажатия кнопки «*Сохранить данные*» следующим образом:

- 1) вызывается метод чтения данных строковой таблицы;
- 2) внутри условия оператора *if* вызывается окно диалога для ввода имени файла сохранения данных;
- 3) при выборе имени в теле *if* объявляется и открывается для записи файловый поток класса *ofstream* с полученным из диалога именем;
- 4) в отдельную строку файла записывается число узлов;
- 5) в цикле *for* последовательно передаются в созданный поток значения всех точек таблицы функции из динамических векторов;
- 6) закрывается файловый поток встроенным методом *fclose*.

20.1.12 Обработчик события нажатия кнопки «*Загрузить данные*» описывается в следующем порядке:

- 1) внутри условия оператора *if* вызывается окно диалога для ввода имени файла чтения данных таблицы;
- 2) при выборе имени в теле *if* объявляется и открывается для чтения файловый поток класса *ifstream* с взятым из диалога именем;
- 3) из файлового потока считывается число узлов и передается в компонент двойного переключателя;
- 4) вызывается метод перерисовки таблицы;
- 5) в цикле *for* последовательно передаются из файлового потока значения всех точек таблицы в динамические вектора и соответствующие ячейки компонента текстовой таблицы;
- 6) закрывается файловый поток встроенным методом *fclose*.

20.1.13 Вывод справочной информации выполняется в обработчике события нажатия кнопки «*О программе*» по аналогии с п. 11.1.11.

20.1.14 Завершение программы выполняется в обработчике события нажатия кнопки «Выход» так:

- 1) удаляются два динамических вектора оператором *delete*;
- 2) вызывается метод закрытия формы приложения.

20.1.15 Производится отладка и тестирование программы с использованием среды Mathcad или MATLAB.

20.1.16 Создается загрузочный модуль приложения и выполняется интерполяция заданной функции в программе.

20.2 Содержание отчета

Отчет по работе № 20 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- текст индивидуального задания;
- схема алгоритма интерполяции заданным методом;
- тексты проекта: главный *spp*-файл, *h*- и *spp*-файлы модуля формы, а также размер загрузочного файла;
- файл исходных данных заданной функции и копия экрана расчета выполненной интерполяции в указанной точке;
- интерполяция заданной функции, выполненная в среде Mathcad или MATLAB для проверки адекватности разработанной программы.

Контрольные вопросы

- 1 Дать определение понятия «интерполяция».
- 2 Назвать методы интерполяции для инженерных расчетов.
- 3 Пояснить геометрический смысл интерполяции.
- 4 Объяснить отличие интерполяции от аппроксимации.
- 5 Показать вид интерполяционного полинома Лагранжа.
- 6 Записать интерполяционный полином Ньютона.
- 7 Объяснить алгоритм линейной интерполяции функции.
- 8 Изобразить алгоритм интерполяции полиномом Лагранжа.
- 9 Построить алгоритм интерполяции полиномом Ньютона.
- 10 Описать алгоритм квадратичной интерполяции.

21 Лабораторная работа № 21. Аппроксимация табличных функций

Цель работы:

- изучение методов аппроксимации таблично заданных функций;
- получение практических навыков решения задачи линейной аппроксимации в среде визуального программирования языка C++.

Задание

Необходимо написать программу-приложение для аппроксимации табличной функции линейным выражением. Программа должна иметь GUI и определять для заданной табличной функции значения двух коэффициентов прямой, а также рассчитать значения максимальной и средней ошибок аппроксимации в узлах таблицы.

21.1 Ход выполнения работы

21.1.1 Проанализировав [3, 7, 8, 10], проектируется алгоритм линейной аппроксимации заданной табличной функции по ГОСТ 19.701.

21.1.2 Запускается среда визуального программирования языка C++, в которой формируется проект визуального приложения.

21.1.3 Параметры окна приложения задаются аналогично п. 11.1.2.

21.1.4 Создается интерфейс приложения на основе компонентов:

- 1) текстового поля для указания числа узлов таблицы функции;
- 2) двойного переключателя для дискретного изменения числа узлов;
- 3) компонента таблицы строк с данными заданной функции;
- 4) двух текстовых полей отображения коэффициентов решения;
- 5) двух текстовых полей для показа ошибок аппроксимации;
- 6) двух окон диалога выбора имени файла при вводе и выводе;
- 7) пяти кнопок: «Загрузить данные», «Сохранить данные», «Аппроксимация», «Визуализация», «О программе» и «Выход»;
- 8) текстовых надписей (меток) для всех элементов интерфейса.

21.1.5 Установка параметров компонентов интерфейса производится аналогично пп. 11.1.3–11.1.5, также связывается между собой двойной переключатель и поле числа узлов таблицы.

21.1.6 В начале *cpp*-файла модуля формы директивами **#include** подключаются стандартные модули **math** и **fstream**.

21.1.7 В верхней части *cpp*-файла модуля формы объявляются следующие глобальные переменные:

- 1) четыре переменных типа **double** для коэффициентов решения, максимальной и суммарной квадратичной ошибок аппроксимации;
- 2) целочисленная переменная числа узлов таблицы функции;
- 3) два указателя *X* и *Y* на тип **double** для динамических векторов.

21.1.8 Создается метод перерисовки таблицы аналогично п. 20.1.8.

21.1.9 Задается метод чтения данных таблицы аналогично п. 20.1.9.

21.1.10 Созданный в п. 21.1.1 алгоритм программируется в обработчике нажатия кнопки «*Аппроксимация*» следующим образом:

- 1) объявляются шесть локальных переменных промежуточных коэффициентов аппроксимации типа ***double***;
- 2) созданным в п. 21.1.9 методом вводятся данные таблицы;
- 3) инициализируются локальные коэффициенты аппроксимации;
- 4) программируется цикл ***for*** суммирования локальных переменных;
- 5) по полученным данным локальных переменных вычисляются коэффициенты результатов линейной аппроксимации;
- 6) инициализируются нулями ошибки аппроксимации;
- 7) задается цикл ***for*** определения ошибок аппроксимации;
- 8) в теле цикла суммируется квадрат текущей ошибки, а также оператором ***if*** сравнивается модуль текущей ошибки аппроксимации с максимумом, который при необходимости переопределяется;
- 9) полученные коэффициенты и ошибки после преобразования в строковые выводятся в соответствующие текстовые поля решения.

21.1.11 Обработчик события нажатия кнопки «*Сохранить данные*» определяется аналогично п. 20.1.11.

21.1.12 Нажатие кнопки «*Загрузить данные*» программируется аналогично п. 20.1.12.

21.1.13 Отображение справочных данных в обработчике события нажатия кнопки «*О программе*» программируется по аналогии с п. 11.1.11.

21.1.14 Выход из программы производится аналогично п. 20.1.14.

21.1.15 В проект добавляется новая форма визуализации аппроксимации аналогично п. 12.1.4. На форму помещаются компоненты:

- 1) график-диаграмма, в которой добавляются две серии данных (исходной функции и линеаризованной на основе аппроксимации) с настройкой остальных свойств аналогично п. 11.1.5;
- 2) текстовые метки пояснения графика;
- 3) две кнопки для сохранения графика в файл и закрытия окна;
- 4) компонент диалогового окна выбора имени графического файла для сохранения диаграммы.

21.1.16 Выполняется объявление следующих внешних глобальных переменных в *spp*-файле формы визуализации со словом ***extern***:

- 1) две переменных типа ***double*** для коэффициентов решения;
- 2) целочисленная переменная числа узлов таблицы;
- 3) два указателя *X* и *Y* на тип ***double*** динамических векторов.

21.1.17 Программирование события отображения окна визуализации выполняется в теле обработчика отображения формы так:

- 1) очищается две серии данных графика;
- 2) задается цикл ***for*** для всех точек таблицы функции;
- 3) в теле цикла к первой серии данных добавляется текущий узел;

4) далее в теле цикла рассчитывается для точки текущего аргумента значение функции по уравнению линейной аппроксимации, затем эти точки добавляются во вторую серию данных.

21.1.18 Программирование событий формы результатов выполняется следующим образом:

1) обработчик нажатия кнопки записи графика в файл программируется по аналогии с п. 12.1.17;

2) обработчик нажатия кнопки закрытия окна аналогичен п. 12.1.17.

21.1.19 В *spp*-файл главной формы подключается заголовочный файл формы визуализации. В обработчике кнопки «Визуализация» вызывается встроенный метод модального отображения формы визуализации.

21.1.20 Выполняется отладка и тестирование созданного приложения с применением среды Mathcad или MATLAB.

21.1.21 Формируется загрузочный модуль приложения и выполняется аппроксимация заданной функции.

21.2 Содержание отчета

Отчет по работе № 21 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- индивидуальное задание на аппроксимацию;
- алгоритм линейной аппроксимации табличной функции;
- тексты файлов проекта: главный *spp*-файл, *h*- и *spp*-файлы модулей форм, с указанием размера *exe*-файла приложения;
- значение коэффициентов, среднеквадратичная и максимальная ошибки аппроксимации в виде экранной копии и график визуализации;
- тест аппроксимации исходных данных в среде MATLAB или Mathcad для проверки адекватности разработанной программы.

Контрольные вопросы

- 1 Дать определение понятия «аппроксимация».
- 2 Перечислить методы аппроксимации в инженерных расчетах.
- 3 Назвать назначение аппроксимация в инженерных расчетах.
- 4 Записать алгоритм линейной аппроксимации.
- 5 Описать алгоритм аппроксимации степенным рядом.
- 6 Пояснить расчет максимальной ошибки аппроксимации.
- 7 Записать расчет средней и суммарной (среднеквадратичной) ошибок аппроксимации.

22 Лабораторная работа № 22. Оптимизация однопараметрических функций

Цель работы:

- изучение численных методов поиска минимума (максимума) однопараметрических функций;
- получение навыков визуального программирования в среде языка C++ задач оптимизации однопараметрических функций.

Задание

Следует разработать схему алгоритма поиска локального экстремума (минимума или максимума) заданной однопараметрической функции на определенном интервале одним из следующих методов: золотого сечения, чисел Фибоначчи, дихотомии, равномерного поиска, Монте-Карло.

22.1 Ход выполнения работы

22.1.1 На основе [3, 7, 8, 10] рисуется алгоритм поиска экстремума однопараметрической функции заданным методом по ГОСТ 19.701.

22.1.2 Загружается среда визуального программирования языка C++ с проектом визуального приложения.

22.1.3 Настройки окна проекта выполняются аналогично п. 11.1.2.

22.1.4 Интерфейс приложения строится на основе компонентов:

- 1) текстовых полей начала и конца интервала поиска экстремума;
- 2) текстового поля заданной точности поиска решения;
- 3) текстовые поля коэффициентов функции (при необходимости);
- 4) двух текстовых полей отображения координат экстремума;
- 5) компонента многострочного текстового редактора для отображения хода решения поиска экстремума;

6) пяти кнопок: «Решение», «Визуализация», «Сохранить отчет», «О программе» и «Выход»;

7) диалоговое окно выбора имени файла для сохранения отчета;

8) текстовых надписей (меток) для всех компонентов интерфейса.

22.1.5 Настройка параметров компонентов главной формы производится аналогично пп. 11.1.3–11.1.5.

22.1.6 В *cpp*-файле модуля формы подключается модуль *math*.

22.1.7 В *cpp*-файле модуля формы объявляются глобальные данные:

- 1) три переменные типа *double* границ интервала и точности поиска;
- 2) две *double*-переменные координат найденного экстремума;
- 3) целочисленная переменная числа итераций поиска;
- 4) переменные типа *double* для коэффициентов функции;
- 5) однопараметрическая функция с результатом типа *double*;
- 6) указатель на тип однопараметрической функции, который сразу инициализируется оператором *&* выше описанной функцией.

22.1.8 Алгоритм из п. 22.1.1 программируется в обработчике нажатия кнопки «*Решение*» следующим образом:

1) считываются исходные данные из полей формы с преобразованием строковых данных в тип *double* в соответствующие переменные границ интервала и точности поиска;

2) объявляются локальные переменные типа *double* для координат двух промежуточных экстремумов;

3) удаляется весь текст из компонента многострочного редактора;

4) объявляется основной цикл поиска экстремума;

5) в теле созданного цикла инкрементируется счетчик итераций;

6) в теле цикла определяются промежуточные точки экстремума;

7) выбирается наилучшее решение из двух промежуточных точек;

8) определяется текущая точность найденного решения;

9) в многострочный редактор выводятся полученные данные цикла;

10) цикл поиска завершается, если точность решения становится равной или больше заданному значению;

11) наилучший экстремум назначается решением и его координаты выводятся в соответствующие текстовые поля формы.

22.1.9 Обработчик события нажатия кнопки «*Сохранить отчет*» определяется следующим образом:

1) в условии оператора *if* вызывается окно ввода имени файла для сохранения данных;

2) если имя файла задано, то оно передается во встроенный метод записи строк текста многострочного редактора в текстовый файл.

22.1.10 Вывод справочных данных в обработчике события нажатия кнопки «*О программе*» определяется по аналогии с п. 11.1.11.

22.1.11 Выход из программы выполняется аналогично п. 20.1.14.

22.1.12 В проекте создается новая форма визуализации вида функции аналогично п.12.1.4. На форму переносятся следующие компоненты:

1) график-диаграмма с настройкой свойств аналогично п. 11.1.5;

2) текстовая метка для подписи графика;

3) кнопка закрытия окна.

22.1.13 В *сpp*-файле формы визуализации объявляются следующие глобальные переменные:

1) две внешние переменные типа *double* для границ интервала;

2) внешние переменные коэффициентов искомой функции;

3) внешний указатель на тип анализируемой функции.

22.1.14 Тело обработчика отображения формы визуализации программируется следующим образом:

1) определяется локальная константа числа расчетных точек;

2) задаются локальные переменные координат текущей точки;

3) определяется локальная переменная шага расчета как разность между границами интервала, деленная на число расчетных точек без 1;

4) инициализируется начальное значение аргумента;

- 5) очищается серия данных графика;
- 6) организуется цикл *while* расчета функции до тех пор, пока не достигнут аргументом конец интервала поиска;
- 7) в теле цикла вычисляется функция с использованием указателя;
- 8) координаты полученной точки функции добавляются к серии данных графика;
- 9) значение аргумента увеличивается на величину шага расчета.

22.1.15 В обработчике события нажатия кнопки закрытия окна вызывается метод, закрывающий форму.

22.1.16 В *spp*-файл главной формы подключается заголовочный файл формы визуализации. В обработчике кнопки «Визуализация» вызывается встроенный метод модального отображения формы визуализации.

22.1.17 Производится отладка и тестирование приложения с применением ресурсов среды Mathcad или MATLAB.

22.1.18 Создается загрузочный модуль приложения и выполняется с его помощью поиск заданного экстремума функции.

22.2 Содержание отчета

Отчет по работе № 22 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- текст индивидуального задания с исходными данными;
- схема алгоритма численного метода оптимизации;
- тексты проекта: главного *spp*-файла, *h*- и *spp*-файлы модулей, а также указывается размер *exe*-файла приложения;
- расчет экстремума заданной функции в виде экранной копии;
- подтверждающий тест, выполненный среде в Mathcad или MATLAB.

Контрольные вопросы

- 1 Перечислить численные методы поиска экстремума для однопараметрических функций.
- 2 Дать определение понятия «оптимизация».
- 3 Указать область применения методов оптимизации.
- 4 Описать алгоритм метода равномерного поиска экстремума.
- 5 Объяснить алгоритм поиска экстремума методом деления интервала поиска пополам.
- 6 Описать алгоритм минимизации методом золотого сечения.
- 7 Объяснить выполнения расчета вектора чисел Фибоначчи.
- 8 Описать алгоритм минимизации методом чисел Фибоначчи.
- 9 Объяснить алгоритм стохастических методов оптимизации.

23 Лабораторная работа № 23. Основы программирования сайтов Internet

Цель работы:

- изучение языка гипертекстовой разметки электронных документов HTML;
- получение практических навыков создания электронных *html*-документов в среде текстового редактора.

Задание

Следует составить электронный документ формата *HTML*, содержащий описание программы расчета заданного численного метода. В составе *html*-документа должны быть следующие элементы:

- 1) *заголовок* с наименованием численного метода;
- 2) метаданные, включающие сведения: автор документа, дата создания документа, информация для поисковых служб;
- 3) *содержание* в виде списка с внутренними ссылками на нижеперечисленные разделы документа:
- 4) основная часть документа, включающая следующие разделы:
 - раздел 1 «*Описание поставленной задачи*» должен содержать назначение метода и использованные математические формулы;
 - раздел 2 «*Создание схем алгоритмов*» должен описывать используемые данные и иметь рисунок алгоритма численного метода;
 - раздел 3 «*Кодирование программы-приложения*» должен показывать состав форм приложения, указывать среду программирования и использованные компоненты интерфейса, а также показывать программную реализацию представленного алгоритма;
 - раздел 4 «*Тестирование программы*» должен иметь таблицу с требуемыми характеристиками ПК и ссылкой на запуск загрузочного файла описываемой программы;
- 5) список использованных источников, который оформляется в виде нумерованного списка, выполненного в соответствии с требованиями ГОСТ 7.1.

23.1 Ход выполнения работы

23.1.1 Выполняется запуск среды текстового редактора после изучения [2], в котором будет сформирован документ. Можно использовать как специализированный *html*-редактор, так и обычную утилиту операционной системы, например *Блокнот*. Следует сразу сохранить текстовый файл документа с расширением *html*.

23.1.2 Указание типа *html*-документа производится в первой строке файла следующим тегом:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

23.1.3 Создание структуры *html*-документа выполняется в следующей последовательности:

- 1) в начале и конце файла определяется тип *html*-документа с помощью контейнерного тега **<HTML>**;
- 2) заголовок документа оформляется с помощью контейнерного тега **<HEAD>** в начале документа после открывающего тега **<HTML>**;
- 3) внутри заголовка указывается с помощью контейнерного тега **<TITLE>** строка заголовка, отображаемого в браузере;
- 4) основное содержание документа оформляется внутри контейнерного тега **<BODY>**, который следует после закрывающего тега заголовка **</HEAD>**.

23.1.4 Внутри открывающего тега **BODY** определяются следующие параметры документа:

- 1) задается цвет фона документа с помощью атрибута **BGCOLOR**;
- 2) фоновый рисунок определяется атрибутом **BACKGROUND**;
- 3) отступ слева устанавливается атрибутом **LEFTMARGIN**;
- 4) отступ справа назначается атрибутом **RIGHTMARGIN**;
- 5) отступ сверху задается атрибутом **TOPMARGIN**;
- 6) отступ снизу настраивается атрибутом **BOTTOM MARGIN**.

23.1.5 Внутри заголовка (контейнерного тега **HEAD**) выполняется внутри тега **<META NAME="тип_данных" CONTENT="значение">** заполнение следующих метаданных документа:

- 1) сведения об авторе документа задаются установкой атрибутами **NAME="AUTHOR"**, а в значение **CONTENT** помещается имя, фамилия и номер учебной группы;
- 2) адрес электронной почты автора устанавливается в **CONTENT** при **NAME="REPLY-TO"**;
- 3) владелец авторских прав на документ (наименование университета) задается в **CONTENT** при **NAME="COPYRIGHT"**;
- 4) дата создания документа определяется в строке **CONTENT** в формате "месяц, число, год и время" при **NAME="DATE"**;
- 5) информация (до 200 символов) для поисковых служб сети Интернет отображается в **CONTENT** при **NAME="DESCRIPTION"**;
- 6) ключевые слова поиска информации (до 800 символов) указываются в **CONTENT** при **NAME="KEYWORDS"**.

23.1.6 Параметры текста в теле документа определяются внутри заголовка контейнерного тега **** с помощью следующих атрибутов:

- 1) **COLOR** – задает отображаемый цвет символов;
- 2) **FACE** – определяет гарнитуру (наименование) шрифта;
- 3) **SIZE** – устанавливает высоту шрифта.

23.1.7 Формирование заголовков внутри тела документа (контейнера **<BODY>**) выполняется следующим образом:

- 1) внутри контейнерного тега ** ** записываются последовательно каждое наименование разделов и список использованных источников. Это формирует внутренние ссылки на разделы документа,

при этом имя ссылки задается латинскими символами;

2) каждая сформированная выше ссылка отдельно помещается внутрь контейнерного тега заголовка **<H1>**;

3) для улучшения восприятия в конце тела каждого раздела проводятся горизонтальные линии тегом **<HR>**.

23.1.8 В первой строке тела документа начинается определение списка его содержания в следующем порядке:

1) записывается контейнерный тег нумерованного списка: **Содержание**;

2) добавляются в список теги заголовка **<H1>** и внутренней ссылки: **<H1>Содержание</H1>**;

3) после описания ссылки в заголовке списка добавляется пять строк элементов списка контейнерными тегами ****;

4) внутри каждой строки списка записывается ссылка на раздел в формате **Название раздела**;

5) в конец каждого из описанных в п. 23.1.7 разделов добавляется ссылка возврата к содержанию: **К содержанию**.

23.1.9 Оформление текстовой части документа выполняется с помощью контейнерных тегов абзаца **<P>**. Параметры шрифта внутри абзацев могут при необходимости регулироваться согласно п. 23.1.6. Перевод строки в абзаце выполняется тегом **
. Дополнительно можно задавать параметры выравнивания текста внутри абзаца с помощью атрибута **ALIGN в теге **<P>**. Контейнерным тегом **<CITE>** или **<Q>** оформляется цитата или сноска на другой материал в тексте документа. Контейнером **<CODE>** отображается в документе пример фрагмента текста программы в третьем разделе документа. Тег **<VAR>** выделяет в тексте переменные в приведенных примерах фрагментов программ.

23.1.10 Формирование формул в *html*-документе можно выполнять следующими способами:

1) оформлять как отдельный абзац с выравниванием текста по центру (**ALIGN="center"**). Отдельные элементы формул нужно при этом формировать, используя следующие теги-контейнеры:

- **<BIG>** – увеличивает размер шрифта внутреннего текста;
- **<SMALL>** – уменьшает размер шрифта внутреннего текста;
- **<SUB>** – отображает подстрочный текст (нижний индекс);
- **<SUP>** – отображает надстрочный текст (верхний индекс);

2) вставкой изображения с рисунком формулы (рассматривается ниже аналогично рисунку схемы алгоритма);

3) с использованием специализированных языковых средств.

23.1.11 Вставка рисунка со схемой алгоритма численного метода в *html*-документ из внешних графических файлов (формата *jpg* или *gif*) выполняется с помощью тега **** внутри второго раздела. Дополнительно тег может содержать атрибуты:

- 1) **ALT** – текст, отображаемый на месте рисунка;
- 2) **HEIGHT** – высота рисунка в пикселях;

3) **WIDTH** – ширина рисунка в пикселях;

4) **ALIGN** – выравнивание рисунка (top, middle, bottom, texttop).

23.1.12 В четвертом разделе приводится таблица характеристик разработанной программы, которая оформляется в следующем порядке:

1) указывается контейнерный тег **<TABLE>**;

2) внутри **<TABLE>** описывается заголовок (наименование) таблицы в контейнере **<CAPTION>**;

3) далее задается первая строка заголовка таблицы внутри контейнера **<TR>**. Текст каждой из ячеек заголовка оформляется внутри тега-контейнера **<TH>**;

4) формируются строки состава таблицы контейнерами **<TR>**, каждая из которых содержит текст, оформленный в ячейках-контейнерах **<TD>**;

5) дополнительно можно задать как для тега **<TABLE>** так и ее компонентов следующие атрибуты:

- **ALIGN** - выравнивание текста ячеек по горизонтали;
- **BACKGROUND** – задает фоновый рисунок в ячейке;
- **BGCOLOR** – определяет цвет фона ячеек;
- **BORDERCOLOR** – устанавливает цвет рамки;
- **COLSPAN** – объединяет ячейки строки;
- **HEIGHT** – назначает высоту ячеек;
- **NOWRAP** – запрещает перенос строк;
- **ROWSPAN** – объединяет ячейки колонки;
- **VALIGN** – выравнивание текста ячейки по вертикали;
- **WIDTH** – устанавливает ширину ячейки.

23.1.13 Ссылка на внешний загрузочный файл программы выполняется с помощью контейнерного тега **<A>** в формате

**** текст ссылки на файл ****

23.1.14 Список использованных источников оформляется как нумерованный список тегом ****. Каждый источник добавляется как отдельная строка внутри списка контейнером ****.

23.1.15 После сохранения созданный *html*-документ просматривается с помощью установленного на ПК обозревателя сети Интернет (браузера) с выполнением загрузки внешнего исполняемого файла по сформированной ссылке. При необходимости корректируется текст документа. По окончании редактирования фиксируется размер *html*-документа.

23.2 Содержание отчета

Отчет по работе №23 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе. Состав отчета следующий:

- титульный лист отчета;
- текст индивидуального задания;
- текст *html*-документа с указанием размера файла;

– копии окон браузера с созданным документом, отображающим содержание, алгоритм и таблицу данных.

Контрольные вопросы

- 1 Объяснить назначение языка HTML.
- 2 Пояснить структуру *html*-документа.
- 3 Рассказать, каким образом формируется структура тела *html*-документа.
- 4 Перечислить варианты задания параметров отображения текста в *html*-документе.
- 5 Показать вставку в *html*-документ графических изображений.
- 6 Записать пример вызова внешнего файла внутри HTML-документа.
- 7 Назвать способы создания *html*-документов на ПК.
- 8 Объяснить, каким образом оформляется список в *html*-документе.
- 9 Рассказать о формировании таблиц в *html*-документе.
- 10 Показать, каким образом формируются метаданные *html*-документа.
- 11 Описать порядок создания содержания в *html*-документе.
- 12 Пояснить, каким образом настраивается цветовая палитра элементов *html*-документа.

24 Лабораторная работа № 24. Табличный редактор Microsoft Excel

Цель работы:

- изучение программного обеспечения для электронных таблиц (табличного процессора) Microsoft Excel;
- получение практических навыков по созданию документов электронных таблиц (ЭТ) в среде Microsoft Excel.

Задание

Необходимо сформировать документ с электронной таблицей для расчета значений функции в заданном диапазоне значений аргументов, расположенных равномерно от начального до конечного значения. При расчете значений функции следует использовать контроль заданного условия, при выполнении которого используется первое расчетное выражение задания, а при невыполнении – второе. По полученным данным таблицы следует построить в декартовой системе координат график функции на отдельном листе документа. На другом листе документа необходимо проанализировать рассчитанные значения таблицы: найти максимальное и минимальное значения функции, ее среднее значение и среднее отклонение, сумму элементов с использованием библиотеки встроенных функций табличного редактора. Значения коэффициентов, указанных в расчетных выражениях, следует вводить в качестве исходных данных в начале листа таблицы. При отображении таблицы с результатами анализа данных следует выводить текущую дату.

24.1 *Ход выполнения работы*

24.1.1 Запуск табличного процессора, установленного на ПК, выполняется с помощью системного меню или соответствующей пиктограммы на основе [9]. Следует сразу заполнить данные свойств документа ЭТ (наименование, тема, автор, руководитель, организация и группа) и сохранить файл с ЭТ в своей рабочей папке.

24.1.2 Определение структуры документа ЭТ выполняется указанием требуемого числа листов (страниц) данных. В составе электронного документа формируется три листа данных, которым рекомендуется задать следующие наименования: «Таблица», «График» и «Анализ».

24.1.3 Для каждого листа данных выполняется настройка параметров вывода на печать: указывается тип и ориентация формата бумаги, и отступы от краев формата.

24.1.4 Формирование исходных данных выполняется на первом листе ЭТ и включает следующее:

1) выбираются тип и высота шрифта (не менее 12 пт) для отображения исходных данных;

2) определяется требуемая ширина ячеек для нормального отображения подписей исходных данных. При необходимости можно объединять текст нескольких соседних ячеек в единый блок данных;

3) тест с пояснением исходных данных записывается в ячейках слева от самих данных;

4) назначается числовой формат для ячеек исходных данных;

5) записываются заданные исходные данные в соответствующие ячейки ЭТ: начало и конец интервала расчета, а также коэффициенты расчетных формул.

24.1.5 Расчет таблицы функции выполняется следующим образом:

1) в отдельной ячейке выполняется расчет шага изменения аргумента на основе разности конечного и начального значений, деленного на число расчетных точек без единицы;

2) формируется заголовок таблицы расчета функции, где подписываются заданные обозначения переменных, и определяются ее границы;

3) в первую ячейку данных аргумента функции заносится адрес ячейки начала интервала расчета из исходных данных после указания символа равенства (=);

4) во вторую колонку записывается расчетное выражение функции внутри оператора **ЕСЛИ()**, где первым записывается выражение при истинности условия, второе при ложности. Для формирования математического выражения можно использовать мастер вставки функций. Адреса исходных данных следует внутри выражения оформлять как абсолютные с использованием символа \$ (переключение выполняет клавиша *F4*), а адреса значений аргумента (ячейка, стоящая слева) – как относительная;

5) расчет аргумента во второй строке таблицы определяется как сумма данных верхней ячейки с абсолютным адресом ячейки со значением шага

изменения, полученного в (1);

б) выделяется и копируется в буфер обмена расчет функции в первой строке таблицы, далее копируются данные в ячейку второй строки расчета функции (стоящую снизу). При этом автоматически будет изменен адрес ячейки аргумента, использующийся в расчетном выражении;

7) выделяется и копируется в буфер обмена вся вторая строка таблицы, которая дальше копируется в требуемое число нижних строк, до строки последнего расчета табличных данных заданного диапазона. При этом автоматически заполняются требуемыми относительными адресами ячейки таблицы.

24.1.6 Настройка параметров спроектированной таблицы расчета выполняется так:

1) по окончании расчетов сформированная таблица выделяется и оформляется выбранным шаблоном границ;

2) рекомендуется закрепить заголовок таблицы при прокрутке листа. Для этого выделяется первая строка таблицы расчета и на ленте «Вид» выбирается элемент «Закрепить области», в меню которого указывается одноименный пункт.

24.1.7 Отображение диаграммы выполняется в следующей последовательности:

1) выполняется переход на второй лист документа ЭТ;

2) вставляется график типа точечной диаграммы на лист;

3) мышкой задаются размеры графика на весь лист (формата А4);

4) выделив график мышкой в ленте (меню) окна выбирается пункт «Выбрать данные» (или его аналог). В распахнувшемся окне следует в соответствующем поле указать диапазон данных для диаграммы, который задается в формате *верхняя_левая_ячейка:нижняя_правая_ячейка*, в который помещаются расчетные значения таблицы;

5) уточняется наименование набора данных (по умолчанию в MS Excel обозначается как *Ряд1*) и нажимается кнопка **Ок**;

б) удаляются с изображения графика автоматически установленная подпись набора данных (так как он единственный; если на графике изображаются несколько наборов, то их наименования переносятся вниз рисунка диаграммы);

7) при необходимости скорректировать определенный элемент графика он выделяется мышкой, и кликом правой кнопки распахивается всплывающее меню, в котором выбирается пункт формата указанного элемента. В отображенном окне меняется настройка выделенного элемента на требуемое значение.

24.1.8 Анализ таблицы результатов расчета функции выполняется на третьем листе данных следующим образом:

1) формируется таблица с результатами анализа данных, где в первом столбце записываются наименования характеристик, а во втором – расчетные формулы;

2) в первой строке таблицы указывается дата проведения расчета, значение которой получается функцией **СЕГОДНЯ()** ил– и ее аналогом;

3) во второй строке находится максимальное значение диапазона ячеек полученных значений функции с помощью подпрограммы **МАКС** или ее аналога;

4) в третьей строке вычисляется минимальное значение диапазона ячеек значений функции с помощью подпрограммы **МИН** или ее аналога;

5) на пятой строке определяется среднее значение диапазона ячеек второго столбца таблицы функцией **СРЗНАЧ** или ее аналога;

6) в шестой строке отображаются данные среднего отклонения данных ячеек второго столбца рассчитанной таблицы с помощью функции **СРОТКЛ** или ее аналога;

7) в последней строке таблицы выполняется суммирование значений ячеек второго столбца таблицы функцией **СУММ**;

8) для полученной таблицы форматируются размеры столбцов и шрифт по аналогии с п. 24.1.4 и задается отображение границ.

24.1.9 Защита расчетных ячеек ЭТ от случайной модификации выполняется для второго и третьего листа документа с помощью ленты «Рецензирование», на которой активируется пункт «Защитить лист».

24.1.10 Защита документа от несанкционированного доступа производится установкой пароля на открытие разработанного документа с помощью ленты «Файл», на которой выбирается пункт «Сохранить как». В раскрывшемся диалоговом окне в списке «Сервис» выбирается строка «Общие параметры», которая раскрывает окно задания паролей на открытие и изменение документа.

24.1.11 Экспорт данных из таблицы в отчет удобно выполнять с помощью буфера обмена. В отчет передаются данные таблиц расчета функции и анализа, а также рисунок графика функции. Также необходимо сделать копию экрана с отображением формул для таблицы расчета.

24.1.12 После проверки готового документа, каждая страница выводится на виртуальный принтер для получения *pdf*-версии отчета. Фиксация размера и типа файла документа с ЭТ выполняется с помощью файлового менеджера ПК.

24.2 Содержание отчета

Отчет по работе № 24 оформляется индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе.

Состав отчета:

- титульный лист отчета;
- текст варианта индивидуального задания;
- результаты расчета электронной таблицы в виде данных;
- распечатка электронной таблицы в виде формул;
- изображение графика рассчитанной функции;
- результаты анализа таблицы функции.

Контрольные вопросы

- 1 Указать область применения электронных таблиц.
- 2 Описать структуру документа с электронной таблицей.
- 3 Пояснить способы адресации ячеек электронной таблицы.
- 4 Разъяснить способы ввода и редактирования содержимого ячеек ЭТ.
- 5 Рассказать о последовательности построения диаграмм по данным в документе ЭТ.
- 6 Перечислить основные форматы данных ячеек ЭТ.
- 7 Пояснить способы экспорта-импорта данных в ЭТ из других приложений.
- 8 Объяснить способы отображения содержания ячеек документа ЭТ.
- 9 Пояснить, каким образом выполняется защита данных ячеек от несанкционированного изменения содержимого.
- 10 Перечислить основные математические функции табличного процессора.
- 11 Перечислить основные статистические функции табличного процессора.
- 12 Рассказать о защите документов электронных таблиц от несанкционированного доступа.
- 13 Описать назначение элементов интерфейса табличного процессора Microsoft Excel или его аналога.
- 14 Описать способы создания расчетных выражений в среде Microsoft Excel или его аналога.

25 Дополнительная информация об изучаемой дисциплине

Цель изучения дисциплины

Согласно [20] целью преподавания дисциплины является обучение студентов методам решения научных и инженерных задач на персональных компьютерах.

Курс является необходимым для специальности 1-53.01.05 «Автоматизированные электроприводы» в научной и практической деятельности инженера-электрика.

Цель методических рекомендаций

Цель издания методических данных рекомендаций – регламентация лабораторного практикума по дисциплине «Информатика» для студентов всех форм обучения первого курса специальности 1-53 01 05 «Автоматизированные электроприводы» в соответствии с [20] и иных нормативных документов.

Список литературы

- 1 **Бекаревич, Ю. Б.** Самоучитель Access 2010 / Ю. Б. Бекаревич. – Санкт-Петербург: БХВ-Петербург, 2011. – 426 с.
- 2 **Гоше, Х. Д.** HTML5. Для профессионалов: пер. с англ. / Х. Д. Гоше. – 2-е изд. – Санкт-Петербург: Питер, 2015. – 560 с.
- 3 **Гулин, А. В.** Введение в численные методы в задачах и упражнениях: учебное пособие / А. В. Гулин, В. А. Морозова, О. С. Мажорова. – Москва: ИНФРА-М; АРГАМАК-МЕДИА, 2014. – 368 с.
- 4 **Гурвиц, Г. А.** Microsoft Access 2010. Разработка приложений на реальном примере / Г. А. Гурвиц. – Санкт-Петербург: БХВ-Петербург, 2010. – 496 с.
- 5 **Долженков, В. А.** Microsoft Office Excel 2010 / В. А. Долженков. – Санкт-Петербург: БХВ-Петербург, 2011. – 804 с.
- 6 **Дьяконов, В. П.** MATLAB 6.5 SP 1/ 7/ 7 SP1/ 7 SP2+Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики / В. П. Дьяконов, В. В. Круглов. – Москва: СОЛОН-ПРЕСС, 2006. – 456 с.
- 7 **Калиткин, Н. Н.** Численные методы: учебное пособие / Н. Н. Калиткин. – Санкт-Петербург: БХВ-Петербург, 2011. – 586 с.
- 8 **Колдаев, В. Д.** Численные методы и программирование: учебное пособие / В. Д. Колдаев, Л. Г. Гагарина. – Москва: ФОРУМ; ИНФРА-М, 2017. – 336 с.
- 9 **Кравченко, Л. В.** Практикум по Microsoft Office 2007 (Word, Excel, Access), PhotoShop: учебно-методическое пособие / Л. В. Кравченко. – Москва: ФОРУМ; ИНФРА-М, 2015. – 168 с.
- 10 **Пантелеев, А. В.** Численные методы. Практикум / А. В. Пантелеев, И. А. Кудрявцева. – Москва: ИНФРА-М, 2017. – 512 с.
- 11 **Полещук, Н. Н.** AutoCAD 2011 / Н. Н. Полещук. – Санкт-Петербург: БХВ-Петербург, 2011. – 752 с.
- 12 **Радаева, Я. Г.** Word 2010: Способы и методы создания профессионально оформленных документов: учебное пособие / Я. Г. Радаева. – Москва: ФОРУМ; ИНФРА-М, 2013. – 160 с.
- 13 **Соколова, Т. Ю.** AutoCAD 2010: учебный курс / Т. Ю. Соколова. – Санкт-Петербург: Питер, 2010. – 576 с.
- 14 **Харитонов, И. А.** Microsoft Office Access 2007 / И. А. Харитонов, Л. В. Рудикова. – Санкт-Петербург: БХВ-Петербург, 2008. – 1280 с.
- 15 **Шушкевич, Г. Ч.** Компьютерные технологии в математике. Система Mathcad 14: учебное пособие: в 2 ч. / Г. Ч. Шушкевич, С. В. Шушкевич. – Минск: Изд-во Гревцова, 2012. – Ч. 2. – 256 с.
- 16 Информатика. Основы использования системы Mathcad: методические указания к курсовому проектированию для студентов специальности 1-53 01 05 «Автоматизированные электроприводы»: в 7 ч. / Сост. В. Н. Абабурко. – Могилев : Белорус.-Рос. ун-т, 2011. – 40 с.
- 17 Информатика. Основы использования системы MATLAB: методические указания к курсовому проектированию для студентов специальности 1-53 01 05

«Автоматизированные электроприводы»: в 7 ч. / Сост. В. Н. Абабурко. – Могилев: Белорус.-Рос. ун-т, 2011. – Ч. 7. – 40 с.

18 **ГОСТ 2.105–95**. Единая система конструкторской документации. Общие требования к текстовым документам. – Минск: Изд-во стандартов, 1995. – 37 с.

19 **ГОСТ 7.1–2003**. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – Минск: Изд-во стандартов, 2004. – 49 с.

20 Информатика: учебная программа учреждения высшего образования по учебной дисциплине для специальности 1-53 01 05 «Автоматизированные электроприводы». – Могилев : Белорус.-Рос. ун-т, 2022. – 23 с.