

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Томский государственный университет систем управления и
радиоэлектроники»

Кафедра компьютерных систем в управлении и проектировании

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ

Методические указания к лабораторным работам

Томск 2018

Кочергин М.И., Ганджа Т.В.

Пакеты прикладных программ / Методические указания к выполнению лабораторных работ. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2018. – 64 с.

Методическое пособие для студентов вузов технических направлений посвящено изучению математических пакетов Mathcad и Matlab, табличного редактора Excel и системы компьютерной вёрстки TeX.

© Кочергин М.И., Ганджа Т.В., 2018

© ТУСУР, 2018

ОГЛАВЛЕНИЕ

Лабораторная работа 1. Табличный редактор MS Excel.....	4
Лабораторная работа 2. Основы работы в Mathcad.....	7
Лабораторная работа 3. Символьные вычисления в Mathcad	11
Лабораторная работа 4. Программирование и отладка в Mathcad	18
Лабораторная работа 5. Работа с файлами данных в Mathcad	21
Лабораторная работа 6. Моделирование в Mathcad	24
Лабораторная работа 7. Основы программирования в Matlab	27
Лабораторная работа 8. Работа с матрицами в Matlab. Решение систем алгебраических уравнений.....	33
Лабораторная работа 9. Обработка табличных данных в Matlab.....	37
Лабораторная работа 10. Работа с файлами данных в Matlab	42
Лабораторная работа 11. Визуализация данных в Matlab	46
Лабораторная работа 12. Моделирование в Matlab.....	49
Лабораторная работа 13. Построение пользовательского интерфейса в Matlab.....	52
Лабораторная работа 14. Построение блок-схем в MS Visio	55
Лабораторная работа 15. Построение схем и диаграмм в MS Visio	58
Лабораторная работа 16. Набор и вёрстка текста в TEX	60
Лабораторная работа 17. Набор формул и стили в TEX.....	62

Лабораторная работа 1. Табличный редактор MS Excel

1. Цель работы

Рассмотрение процедуры работы с матрицами (массивами) в табличном процессоре *MS Excel* и построения разветвляющихся алгоритмов для обработки табличных данных.

2. Указания к выполнению работы

Работа с массивами. Обычно формула при обработке нескольких аргументов возвращает одно значение; в качестве аргумента формулы может при этом выступать либо ссылка на ячейку, содержащую значение, либо само значение. Для создания ссылки на диапазон ячеек используется формула массива, позволяющая ввести в одну ячейку массив значений.

Этот массив значений называется массивом констант; удобен он тем, что при этом не требуется заполнять значениями вспомогательные ячейки.

Записать массив $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ непосредственно в функцию (в строке функции) можно следующим образом: {1;2;3;4}.

Также массив может быть задан как интервал ячеек, например, A1:C3.

Для работы с матрицами в *MS Excel* используются следующие основные функции:

– МОПРЕД(массив) – возвращает определитель матрицы (матрица хранится в массиве);

– МОБР(массив) – возвращает обратную матрицу для матрицы, хранящейся в массиве;

– МУМНОЖ(массив1;массив2) – возвращает произведение матриц (матрицы хранятся в массивах);

– ТРАНСП(массив) – возвращает транспонированный массив.

– Реализация разветвляющихся алгоритмов. В *MS Excel* предусмотрены функции, позволяющие делать тот или иной расчет в зависимости от выполнения некоторого условия, например:

– СУММЕСЛИ (диапазон; критерий; диапазон_суммирования) – возвращает сумму значений из указанного диапазона ячеек согласно условию.

– СЧЁТЕСЛИ(диапазон; критерий) – возвращает количество ячеек(строк) удовлетворяющих указанному критерию.

3. Содержание работы

1. Работа с массивами.

1.1. Вычисление определителя квадратной матрицы.

1.2. Вычисление обратной матрицы.

1.3. Умножение матриц.

- 1.4. Поиск наибольших элементов в матрице.
- 1.5. Поиск наименьших элементов в матрице.
- 1.7. Транспонирование матриц.
- 1.8. Решение системы линейных алгебраических уравнений матричным методом

2. Реализация разветвляющихся алгоритмов

- 2.1. Использование выражений сравнения в условиях
- 2.2. Использование логических функций в условиях
- 2.3. Использование вложенной функции ЕСЛИ
- 2.4. Построение таблицы функции с ветвлениями
- 2.5. Использование функций СЧЕТЕСЛИ и СУММЕСЛИ

4. Порядок проведения работы

Задание 1. Создайте таблицу в Excel. Введите матрицу согласно варианту. Рассчитайте на этом же листе определитель, обратную матрицу, умножение матрицы на обратную.

Задание 2. Реализуйте функции поиска наибольшего и наименьшего элементов в матрице.

Задание 3. Транспонируйте матрицу.

Задание 4. Решите матричным методом систему линейных алгебраических уравнений

5. Варианты заданий

Варианты к заданиям 1-3 представлены в табл. 1.1

Таблица 1.1 – Варианты к заданиям 1-3

Вариант 1				Вариант 2				Вариант 3				Вариант 4				Вариант 5			
41	32	48	48	22	33	34	33	14	35	22	10	36	33	48	38	45	8	41	10
46	5	49	25	46	2	38	9	3	16	20	25	38	9	18	13	48	13	13	13
7	14	8	41	40	43	38	36	5	48	39	23	14	6	30	26	28	43	47	31
46	28	49	8	48	47	20	2	42	2	40	33	34	25	12	35	7	13	18	24
Вариант 6				Вариант 7				Вариант 8				Вариант 9				Вариант 10			
18	46	20	27	29	9	9	35	12	27	6	41	13	10	44	43	13	10	44	43
42	15	29	39	24	40	31	38	46	50	49	44	41	14	29	32	41	14	29	32
30	38	4	47	1	16	14	23	8	4	1	5	22	8	28	18	22	8	28	18
28	38	3	7	17	27	33	5	42	23	39	20	46	7	8	26	46	7	8	26
Вариант 11				Вариант 12				Вариант 13				Вариант 14				Вариант 15			
21	10	46	17	40	5	29	42	33	28	35	40	25	26	33	18	32	24	12	16
4	12	48	46	20	7	3	1	37	15	10	5	22	26	19	47	30	12	9	47
12	21	25	19	13	48	12	3	33	38	19	47	23	41	41	44	11	43	12	22
7	3	25	6	21	48	18	9	23	10	32	39	16	40	27	28	16	10	22	10
Вариант 16				Вариант 17				Вариант 18				Вариант 19				Вариант 20			
46	13	31	15	5	47	12	27	34	2	5	7	25	45	2	46	41	45	49	3
49	21	36	16	14	37	23	12	20	45	14	37	39	17	38	31	29	2	36	35
22	30	12	22	41	25	49	25	19	46	17	6	36	35	26	31	10	25	26	3
6	14	6	26	2	29	28	32	50	40	34	33	46	10	24	43	12	9	24	4

Варианты к заданию 4 представлены в табл. 1.2.

Таблица 1.2 – Варианты к заданию 4

1	$\begin{cases} 27x_1 + 3,3x_2 + 1,3x_3 = 21 \\ 3,5x_1 - 17x_2 + 2,8x_3 = 17 \\ 4,1x_1 + 5,8x_2 - 17x_3 = 8 \end{cases}$	2	$\begin{cases} 17x_1 + 2,8x_2 + 1,9x_3 = 7 \\ 2,1x_1 + 34x_2 + 1,8x_3 = 11 \\ 4,2x_1 - 1,7x_2 + 13x_3 = 28 \end{cases}$	15	$\begin{cases} 78x_1 + 5,3x_2 + 4,8x_3 = 18 \\ 3,3x_1 + 11x_2 + 1,8x_3 = 23 \\ 4,5x_1 + 3,3x_2 + 28x_3 = 34 \end{cases}$	16	$\begin{cases} 38x_1 + 4,1x_2 - 2,3x_3 = 48 \\ -2,1x_1 + 39x_2 - 5,8x_3 = 33 \\ 1,8x_1 + 1,1x_2 - 21x_3 = 58 \end{cases}$
3	$\begin{cases} 31x_1 + 2,8x_2 + 1,9x_3 = 2 \\ 1,9x_1 + 31x_2 + 2,1x_3 = 21 \\ 7,5x_1 + 3,8x_2 + 48x_3 = 56 \end{cases}$	4	$\begin{cases} 91x_1 + 5,6x_2 + 7,8x_3 = 98 \\ 3,8x_1 + 51x_2 + 2,8x_3 = 67 \\ 4,1x_1 + 5,7x_2 + 12x_3 = 58 \end{cases}$	17	$\begin{cases} 17x_1 - 2,2x_2 + 30x_3 = 18 \\ 2,1x_1 + 19x_2 - 2,3x_3 = 28 \\ 4,2x_1 + 3,9x_2 - 31x_3 = 51 \end{cases}$	18	$\begin{cases} 28x_1 + 3,8x_2 - 32x_3 = 45 \\ 2,5x_1 - 28x_2 + 3,3x_3 = 71 \\ 6,5x_1 - 7,1x_2 + 48x_3 = 63 \end{cases}$
5	$\begin{cases} 33x_1 + 2,1x_2 + 2,8x_3 = 8 \\ 4,1x_1 + 37x_2 + 4,8x_3 = 57 \\ 2,7x_1 + 1,8x_2 + 11x_3 = 32 \end{cases}$	6	$\begin{cases} 76x_1 + 5,8x_2 + 4,7x_3 = 101 \\ 3,8x_1 + 41x_2 + 2,7x_3 = 97 \\ 2,9x_1 + 2,1x_2 + 38x_3 = 78 \end{cases}$	19	$\begin{cases} 33x_1 + 3,7x_2 + 4,2x_3 = 58 \\ 2,7x_1 + 23x_2 - 2,9x_3 = 61 \\ 4,1x_1 + 4,8x_2 - 50x_3 = 70 \end{cases}$	20	$\begin{cases} 71x_1 + 6,8x_2 + 6,1x_3 = 70 \\ 5,0x_1 + 48x_2 + 5,3x_3 = 61 \\ 8,2x_1 + 7,8x_2 + 71x_3 = 58 \end{cases}$
7	$\begin{cases} 32x_1 - 2,5x_2 + 3,7x_3 = 65 \\ 0,5x_1 + 34x_2 + 1,7x_3 = -2,4 \\ 1,6x_1 + 2,3x_2 - 15x_3 = 43 \end{cases}$	8	$\begin{cases} 54x_1 - 2,3x_2 + 3,4x_3 = -35 \\ 4,2x_1 + 17x_2 - 2,3x_3 = 27 \\ 3,4x_1 + 2,4x_2 + 74x_3 = 19 \end{cases}$	21	$\begin{cases} 37x_1 + 3,1x_2 + 4,0x_3 = 50 \\ 4,1x_1 + 45x_2 - 4,8x_3 = 49 \\ -2,1x_1 - 3,7x_2 + 18x_3 = 27 \end{cases}$	22	$\begin{cases} 41x_1 + 5,2x_2 - 5,8x_3 = 70 \\ 3,8x_1 - 31x_2 + 4,0x_3 = 53 \\ 7,8x_1 + 5,3x_2 - 63x_3 = 58 \end{cases}$
9	$\begin{cases} 36x_1 + 1,8x_2 - 4,7x_3 = 38 \\ 2,7x_1 - 36x_2 + 1,9x_3 = 4 \\ 1,5x_1 + 4,5x_2 + 33x_3 = -16 \end{cases}$	10	$\begin{cases} 56x_1 + 2,7x_2 - 1,7x_3 = 19 \\ 3,4x_1 - 36x_2 - 6,7x_3 = -24 \\ 0,8x_1 + 1,3x_2 + 37x_3 = 12 \end{cases}$	23	$\begin{cases} 37x_1 - 2,3x_2 + 4,5x_3 = 24 \\ 2,5x_1 + 47x_2 - 7,8x_3 = 35 \\ 1,6x_1 + 5,3x_2 + 13x_3 = -24 \end{cases}$	24	$\begin{cases} 63x_1 + 5,2x_2 - 0,6x_3 = 15 \\ 3,4x_1 - 23x_2 + 3,4x_3 = 27 \\ 0,8x_1 + 1,4x_2 + 35x_3 = -23 \end{cases}$
11	$\begin{cases} 27x_1 + 0,9x_2 - 1,5x_3 = 35 \\ 4,5x_1 - 28x_2 + 6,7x_3 = 26 \\ 5,1x_1 + 3,7x_2 - 14x_3 = -14 \end{cases}$	12	$\begin{cases} 45x_1 - 3,5x_2 + 7,4x_3 = 25 \\ 3,1x_1 - 6x_2 - 2,3x_3 = -15 \\ 0,8x_1 + 7,4x_2 - 5x_3 = 64 \end{cases}$	25	$\begin{cases} 15x_1 + 2,3x_2 - 3,7x_3 = 45 \\ 2,8x_1 + 34x_2 + 5,8x_3 = -32 \\ 1,2x_1 + 7,3x_2 - 23x_3 = 56 \end{cases}$	26	$\begin{cases} 1,3x_1 + 3,3x_2 + 27x_3 = 21 \\ 2,8x_1 - 17x_2 + 3,5x_3 = 17 \\ 17x_1 + 5,8x_2 - 4,1x_3 = 8 \end{cases}$
13	$\begin{cases} 38x_1 + 6,7x_2 - 1,2x_3 = 52 \\ 6,4x_1 + 13x_2 - 2,7x_3 = 38 \\ 2,4x_1 - 4,5x_2 + 35x_3 = -6 \end{cases}$	14	$\begin{cases} 54x_1 - 6,2x_2 - 0,5x_3 = 5,2 \\ 3,4x_1 + 23x_2 + 0,8x_3 = -8 \\ 2,4x_1 - 1,1x_2 + 38x_3 = 18 \end{cases}$	27	$\begin{cases} 1,9x_1 + 2,8x_2 + 17x_3 = 7 \\ 1,8x_1 + 34x_2 + 2,1x_3 = 11 \\ 13x_1 - 1,7x_2 + 4,2x_3 = 28 \end{cases}$	28	$\begin{cases} 1,9x_1 + 2,8x_2 + 31x_3 = 2 \\ 2,1x_1 + 31x_2 + 1,9x_3 = 21 \\ 48x_1 + 3,8x_2 + 7,5x_3 = 56 \end{cases}$

6. Контрольные вопросы

- Вычисление определителя матрицы в *Excel*.
- Вычисление обратной матрицы в *Excel*.
- Умножение массивов в *Excel*.
- Транспонирование массива в *Excel*.
- Нахождение *K*-го наибольшего элемента массива в *Excel*.
- Вложенная функция ЕСЛИ в *Excel*.
- Построение таблицы функции с ветвлениями в *Excel*.
- Функции СЧЕТЕСЛИ и СУММЕСЛИ в *Excel*.

7. Содержание отчета

Вычисления оформите в *MS Excel* на 2-х листах: «Массивы» и «Ветвление». Отчёт должен содержать краткое описание всех используемых функций; выводы по работе, которые необходимо разместить на отдельном листе; титульный лист.

Лабораторная работа 2. Основы работы в Mathcad

1. Цель работы

Знакомство с интерфейсом математического пакета *Mathcad* и освоение базовых принципов работы в математических пакетах.

2. Указания к выполнению работы

Документ *Mathcad* представляет собой интерактивный документ с областями для вычислений, текста и расположения других объектов. К **основным элементам математических выражений** *Mathcad* относятся типы данных, операторы, функции и управляющие структуры.

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

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, либо в виде чисел с определенным шагом, меняющихся от начального значения до конечного.

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) и двумерные (матрицы).

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 в скобках в левой части, являются параметрами и должны задаваться до определения функции.

Текстовые фрагменты представляют собой фрагменты текста, которые пользователь хотел размещает в своем документе.

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.

3. Содержание работы

1. Изучение интерфейса *Mathcad*

2. Вычисление выражений в документе
3. Изменение формата представления результата и точности
4. Задание и вычисление функций
5. Построение графиков в декартовых координатах
6. Построение графиков поверхности
7. Создание анимационного клипа

4. Порядок выполнения работы

1. Вычислите « $n!$ ». Вычислите логарифм $Ln(n)$. Вычислите e^n .
2. Вычислите выражения:

$$e^n + \sqrt{47 + 56^n + \sin\left(\frac{n}{10}\right)}, \quad e^n + \sqrt{47} + 56^n + \sin\left(\frac{n}{10}\right), \quad \frac{5 \cdot |-6| + n}{8^n}, \quad \text{где вместо } n \text{ поставьте номер варианта.}$$

3. Вычислите $|-n| =$, $n! =$, $\sqrt{n} =$.
4. Определите переменные: $a := 3.4$, $b := 6.22$, $c := 0.149$ и выражения:

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

5. Вычислите: $10x^n - 5y^2$, при $x=n/2$ и $y = -n/3$.
6. Вычислите: $((n+10)^{-1/2}) / ((3 \cdot n)^{1/3} - 81^{1/n})$
7. Вычислите функцию $y = 4x^2 + 5x + 3 \cdot n$ для $x=1, 2, \dots, n+4$.
8. Выполнить следующие операции:

$$\sum_{i:=1..10} i \prod_i (i+n) \int_0^{0.4} x^2 \cdot \lg(x+n) dx = \int_{0.8}^{1.2} \frac{\text{ctg } x}{(\sin x)^2} dx_x$$

$$:= 2 \frac{d}{dx} x^5 \frac{d}{dx} \sin(x)$$

9. Постройте декартовы («X-Y Зависимость») и полярные («Полярные Координаты») графики следующих функций:

$$X(\alpha) := \cos(\alpha) \cdot \sin(\alpha) \quad Y(\alpha) := 1.5 \cos(\alpha)^2 - 1 \quad P(\alpha) := \cos(\alpha).$$

10. Используя команду «Вставка \Rightarrow Матрица», создайте матрицу Q размером 6×6 , заполните ее произвольно и отобразите графически с помощью команды «Вставка \Rightarrow График \Rightarrow Поверхности».

11. Постройте график поверхности («Поверхности») и карту линий уровня («Контурный») для функции двух переменных:

$$X(t, \alpha) := t \cdot \cos(\alpha) \cdot \sin(\alpha) \quad \text{двумя способами:}$$

а) С помощью функции *CreateMesh* (сетка размером 40×40 , диапазон изменения t от -5 до 5 , α - от 0 до 2π).

б) Задав поверхность математически, для этого:

- Определите функцию $X(t, \alpha)$
- Задайте 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 2π с шагом $\pi/20$ $\alpha_j := \pi/20 \cdot j$.

- Определите матрицу $M_{ij} := X(t_i, \alpha_j)$ и отобразите ее графически.

12. Отобразите графически пересечение поверхностей

$$f1(x, y) := \frac{(x+y)^2}{10} \quad f2(x, y) := 5 \cdot \cos\left(\frac{x-y}{3}\right)$$

13. Используя переменную *FRAME* и команду «Вид \Rightarrow Анимация», создайте анимационные клипы с помощью данных приведенных в Таблице 2.1.

5. Варианты заданий

В работе вместо n используйте номер варианта.

Таблица 2.1 – Варианты к заданию 13

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

Окончание табл. 2.1.

№	Переменные и функции	FRAME	Тип графика
4	$r := \text{FRAME}$ $R := 6$ $n := 0 .. 20$ $m := 0 .. 20$ $v_n := \frac{2 \cdot \pi \cdot n}{r + 1}$ $w_m := \frac{2 \cdot \pi \cdot m}{r + 1}$ $x_{mn} := (R + r \cdot \cos(v_n)) \cdot \cos(w_m)$ $y_{mn} := (R + r \cdot \cos(v_n)) \cdot \sin(w_m)$ $z_{mn} := r \cdot \sin(v_n)$	от 0 до 20	График Поверхности (границы на всех осях установить от -11 до 11) В метке для ввода матрицы укажите (x, y, z)

6. Контрольные вопросы

- С помощью какого оператора можно вычислить выражение?
- Как вставить текстовую область в документ MathCAD?
- Чем отличается глобальное и локальное определение переменных?
- Как изменить формат чисел для всего документа? Как изменить формат чисел для отдельного выражения?
- Как вставить встроенную функцию в документ MathCAD?
- Как определить дискретные переменные с произвольным шагом?
- Как определить индексированную переменную?
- Какие виды массивов в MathCAD существуют? Опишите способы создания массивов в MathCAD.
- Какая переменная определяет нижнюю границу индексации элементов массива?
- Как просмотреть содержимое массива, определенного через дискретный аргумент?
- Как построить графики поверхности; полярный; декартов?
- Как построить несколько графиков в одной системе координат?
- Как изменить масштаб графика?
- Как определить координату точки на графике?

7. Содержание отчета

Работу оформите в *Mathcad*, снабдив каждое задание нумерованным заголовком и кратким комментарием, а также добавьте титульный лист в начало работы. Отчёт должен содержать выводы по работе, которые необходимо разместить на отдельном листе, а также титульный лист.

Лабораторная работа 3. Символьные вычисления в Mathcad

1. Цель работы

Знакомство с процедурой аналитических преобразований средствами *Mathcad*.

2. Указания к выполнению работы

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.

Символьные операции можно выполнять двумя способами:

- Непосредственно в командном режиме (используя операции меню **«Символьные операции»**);
- С помощью операторов символьного преобразования (используя палитру инструментов **«Символьное преобразование с ключевыми словами»** – ).

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 («Символьные операции» – «Вычислить»)

Если в документе есть выделенное выражение, то с ним можно выполнять различные операции, представленные ниже («Меню – Символьные операции»):

- **«Вычислить»** – преобразовать выражение с выбором вида преобразований из подменю;
 - **«Аналитически»** [Shift]+F9 – выполнить символьное (аналитическое) преобразование выделенного выражения;

- **«С плавающей точкой»** – вычислить выделенное выражение в вещественных числах;
- **«В комплексных числах»** – 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.;
- **«Развернуть»** – раскрыть выражение [например, для $(X + Y)(X - Y)$ получаем $X^2 - Y^2$];
- **«Факторизовать»** – разложить число или выражение на множители [например, $X^2 - Y^2$ даст $(X + Y)(X - Y)$];
- **«Сборка»** – собрать слагаемые, подобные выделенному выражению, которое может быть отдельной переменной или функцией со своим аргументом (результатом будет выражение, полиномиальное относительно выбранного выражения);
- **«Полиномиальные коэффициенты»** – по заданной переменной найти коэффициенты полинома, аппроксимирующего выражение, в котором эта переменная использована.

Операции с выделенными переменными («Символьные операции» – «Переменная»)

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 Переменная:

- **«Решить»** – найти значения выделенной переменной, при которых содержащее ее выражение становится равным нулю;
- **«Подставить»** – заменить указанную переменную содержимым буфера обмена;
- **«Дифференцировать»** – дифференцировать выражение, содержащее выделенную переменную, по этой переменной (остальные переменные рассматриваются как константы);
- **«Интегрировать»** – интегрировать все выражение, содержащее переменную, по этой переменной;
- **«Разложить в ряд»** – найти несколько членов разложения выражения в ряд Тейлора относительно выделенной переменной;
- **«Преобразовать к дробно-рациональному виду»** – разложить на элементарные дроби выражение, которое рассматривается как рациональная дробь относительно выделенной переменной.

Операции с выделенными матрицами («Символьные операции» – «Матрица»)

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

- «**Транспонировать**» – получить транспонированную матрицу;
- «**Обратить**» – создать обратную матрицу;
- «**Определитель**» – вычислить детерминант (определитель)

матрицы.

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

Операции преобразования («Символьные операции» – «Преобразование»)

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

- «**Фурье**» – выполнить прямое преобразование Фурье относительно выделенной переменной;
- «**Обратное Фурье**» – выполнить обратное преобразование Фурье относительно выделенной переменной;
- «**Лапласа**» – выполнить прямое преобразование Лапласа относительно выделенной переменной (результат - функция переменной s);
- «**Обратное Лапласа**» – выполнить обратное преобразование Лапласа относительно выделенной переменной (результат - функция переменной t);
- «**Z-преобразование**» – 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).

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 (табл. 3.1).

Таблица 3.1 – Пример задания операторов пользователя

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

Отметим, что 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 \div , поскольку этот символ системой не используется.

Ещё одна особенность 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 «Центра ресурсов» («Ресурс Центр \Rightarrow Справочный стол и краткое руководство \Rightarrow Дополнительные математические символы»).

После того как оператор задан, его можно использовать, как функцию и как оператор. Примеры показаны на Рис. 3.1.

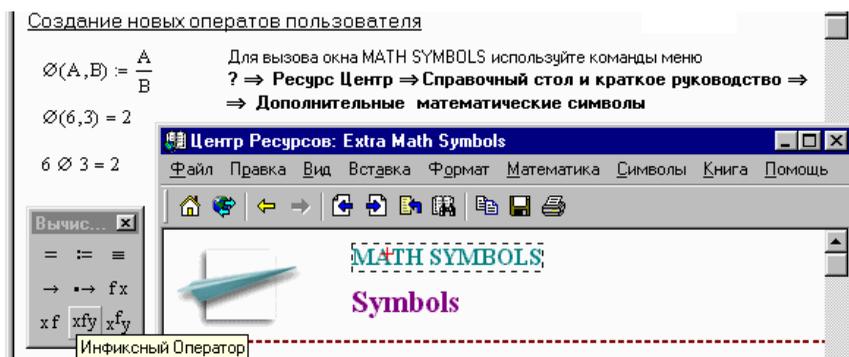


Рис. 3.1 – Создание оператора пользователя

Для применения нового оператора надо вывести его шаблон с помощью панели математических знаков.

3. Содержание работы

1. Представление расчетов с плавающей запятой
2. Вывод чисел в комплексной форме
3. Факторизация выражений
4. Представление выражений в дробно-рациональном виде
5. Разложение выражения в ряд Тейлора
6. Аналитическое интегрирование

7. Аналитическое дифференцирование
8. Матричные аналитические преобразования
9. Вычисление пределов
10. Задание оператора пользователя

4. Порядок выполнения работы

1. Используя операцию «Символы \Rightarrow Расчеты \Rightarrow С плавающей точкой», представьте: а) число $n \cdot \pi$ в 7 позициях; б) число 12,345667 в 3 позициях.

2. Для полинома $g(x)$ (см. Табл. 3.2) выполните следующие действия: а) разложите на множители, б) подставьте выражение $x = y + z$ в $g(x)$, используя операцию «Символьные вычисления \Rightarrow Переменные \Rightarrow Подставить», в) используя операцию «Символьные вычисления \Rightarrow Развернуть», разложите по степеням выражение, полученное в б); г) используя операцию «Символьные вычисления \Rightarrow Сборка», сверните выражение, полученное в 3), по переменной z .

3. Разложите выражения на элементарные дроби используя операцию «Символьные вычисления \Rightarrow Переменные \Rightarrow Преобразование к дробно-рациональному виду»:

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

4. Разложите выражения в ряд с заданной точностью, используя операцию «Символьные вычисления \Rightarrow Переменные \Rightarrow Разложить в ряд»:

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

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

5. Найдите первообразную аналитически заданной функции $f(x)$ (табл. 3.3).

6. Определите символьное значение первой и второй производных $f(x)$ (табл. 3.2).

7. Транспонируйте, инвертируйте матрицу M . Найдите её определитель.

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

8. Вычислите пределы:

$$\lim_{x \rightarrow 1} \frac{x^2 + 2 \cdot x + 5}{x^2 + 1} \quad \lim_{x \rightarrow 0^+} (1 + x)^{\frac{1}{x}} \quad \lim_{x \rightarrow +\infty} x \cdot (\sqrt{x^2 + 1} - x)$$

9. Задайте операторы пользователя:

- Для пересчета единиц электроэнергии (кВт*ч в Дж, эВ в Дж).
- Для пересчета единиц электроэнергии (кВт*ч в Дж, эВ в Дж).
- Для пересчета единиц мощности (эрг/с в Вт, кгс/с в Вт).

5. *Варианты задания*

Таблица 3.2 – Варианты к заданию 2

№ варианта	$g(x)$	№ варианта	$g(x)$
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	$x^4 - 6x^3 + 4x^2 + 10x + 75$		

Таблица 3.3 – Варианты к заданиям 5, 6.

№	$f(x)$	№	$f(x)$	№	$f(x)$
1	$1/(\lg 2x + 1)$	6	$x^2 \arctg(x/3)$	11	$(2x + 3) \sin x$
2	$\cos x / (2x + 5)$	7	$e^{2x} \sin 3x$	12	$\cos 3x / (1 - \cos 3x)$
3	$1/(x \sqrt{x^3 + 4})$	8	$\text{ctg} 2x / (\sin 2x)$	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	$\sqrt{1 + e^{-x}}$

6. *Контрольные вопросы*

- Способы выполнения символьных операций в Mathcad.
- Символьные операции с выделенными выражениями.
- Символьные операции с выделенными переменными.
- Символьные операции с выделенными матрицами.

- Символьные операции преобразования.
- Какие параметры определяет стиль представления результатов вычислений, и где он задается?
- В каких случаях результат символьных преобразований помещается в буфер обмена?
- Как символьно решить уравнение или систему уравнений в Mathcad? Какой знак равенства используется?
- Назовите особенности использования символьного решения уравнений.
- Каким образом можно вычислить предел в Mathcad?
- Для чего необходимо задание операторов пользователя?
- Как задать оператор пользователя?

7. Содержание отчета

Работу оформите в *Mathcad*, снабдив каждое задание нумерованным заголовком и кратким комментарием, а также добавьте титульный лист в начало работы. Отчёт должен содержать выводы по работе, которые необходимо разместить на отдельном листе.

Лабораторная работа 4. Программирование и отладка в

Mathcad

1. Цель работы

Изучение принципов реализации алгоритмов и их отладки в *Mathcad*, знакомство с панелью программирования и её основными операторами для создания исполняемых программ.

2. Указания к выполнению работы

Основные особенности программирования в *Mathcad*:

1. Операторы программы набираются не вручную, а выбираются на Панели программирования (рис. 4.1).

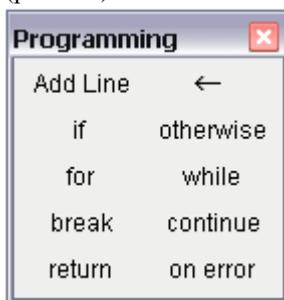


Рис. 4.1 – Панель программирования Mathcad

2. Все программы в *Mathcad* – функции, а не процедуры.

3. Переменные, объявленные ранее (выше или левее) передаются в программу как значения и не перезаписываются.

4. Функцию операторных скобок (начало и конец операторов for, while, if) выполняют вертикальные линии.

5. Внутри программы для присваивания значения переменной используется знак локального присваивания ←.

6. Функция должна заканчиваться идентификатором (именем) возвращаемой переменной (вектора переменных).

7. Программа может содержать рекурсию.

Отладка программ в *Mathcad*

Для отладки программ *Mathcad*, созданных при помощи панели программирования, используются две функции: *trace* и *pause*. Обе выводят информацию в отладочное окно – «Окно трассировки» (*Trace Window*), которое открывается через меню «Инструменты – Отладка – Переключение режима отладки».

Эти функции необходимо включать непосредственно в код программы, используя специальные шаблоны для отображения информации о

значении переменных в окне трассировки. Для того, чтобы информация о переменных появилась в окне трассировки необходимо запустить процесс вычислений. Для этого достаточно выделить вызываемую функцию (вызываемую с аргументами, а не объявляемую) и нажать «Инструменты – Вычислить – Вычислить сейчас» или горячую клавишу F9.

Функция отладки *trace*. Функция *trace* выводит информацию о значении запрашиваемых переменных в Окно трассировки без остановки исполнения программы. У нее может быть любое количество аргументов. Если у функции один аргумент, то именно его значение она выведет в специальном отладочном окне. Если же аргументов несколько, то их используют в следующем виде: `trace("x = {0}, y = {1}", x, y)`. Здесь номера 0, 1 и т.д. означают порядковую позицию соответствующей переменной в списке аргументов функции *trace*, следующих за самым первым аргументом. В результате своей работы функция *trace* с таким набором аргументов выведет строку вида `"x = 0.2, y = 0.03"` (без кавычек), где вместо 0.2 и 0.03 будут стоять те соответствующие реальные значения указанных переменных, которые те принимают в ходе выполнения программы.

Функция отладки *pause*. Функция *pause* останавливает исполнение программы (на том месте, где записана) выводит информацию о значении запрашиваемых переменных в Окно трассировки. Для продолжения исполнения программы требуется нажать на кнопку «Инструменты – Отладка – Продолжить» (*Continue*) или горячая клавиша `<Alt> + <F9>`. Синтаксис использования функции *pause* аналогичен таковому для *trace*

Для очистки окна трассировки необходимо нажать на него правой кнопкой и в контекстном меню выбирать пункт «Очистить всё».

Запустить процедуру отладки можно следующим образом:

1. Разместить в теле программы стоп-функции
2. Открыть окно отладки («Инструменты – Отладка – Переключение режима отладки»)
3. Выделить (нажать для появления синего подчеркивания) область, где вызывается Ваша программа, и запустить вычисления («Инструменты – Вычисления – Вычислить сейчас» или клавишей `<F9>`).

Далее можно двигаться от одной точки останова к другой посредством нажатия комбинации `<Alt> + <F9>` или «Инструменты – Отладка – Продолжить». В обоих случаях место вызова функции должно быть выделено (синим подчеркиванием). Для остановки отладки использовать «Инструменты – Отладка – Прервать».

3. Содержание работы

1 Отладка алгоритма в *Mathcad*

2 Реализация алгоритма с помощью панели программирования

4. **Порядок выполнения работы**

Задание 1. Восстановите работоспособность алгоритма согласно варианту.

Задание 2. Используя «Панель программирования» реализуйте алгоритм согласно варианту.

5. **Варианты заданий**

Варианты к заданию 1 «Отладка программы»:

1. Метод половинного деления
2. Метод полного сканирования
3. Расчет дисперсии
4. Расчет данных для построения гистограммы

Варианты к заданию 2 «Программирование в *Mathcad*»:

1. Посчитать количество положительных или отрицательных (задается при вызове) элементов в матрице любого размера.
2. Поиск наименьшего в матрице любого размера (задается при вызове).
3. Вычисление элемента ряда $1/(n!+n^2)$ с точностью ϵ . (Вычисление должно закончиться, когда рассчитанный элемент ряда будет меньше ϵ , при этом n возрастает на каждой итерации).
4. Заполнить элементы матрицы размера $m \times m$ в шахматном порядке случайными числами из заданного диапазона.
5. Транспонирования матрицы любого размера. Нахождение определителя матрицы.
6. Нахождение обратной матрицы любого размера.

6. **Контрольные вопросы**

- Отладка в *Mathcad*. Используемые функции. Порядок процедуры.
- Использование панели программирования в *Mathcad*. *Addline*. Локальное присваивание. Аргументы и вызов функции.
- Организация циклов в *Mathcad*. Синтаксис.
- Команды панели *Symbolic*
- Техника программирования в *Mathcad*

7. **Содержание отчета**

Работу оформите в *Mathcad*, снабдив каждое задание нумерованным заголовком и кратким комментарием, а также добавьте титульный лист в начало работы. Отчёт должен содержать выводы по работе, которые необходимо разместить на отдельном листе.

Лабораторная работа 5. Работа с файлами данных в Mathcad

1. Цель работы

Изучение принципов работы с файлами данных в *Mathcad*, знакомство с функциями чтения изображений и структурированных данных из файла, а также их записи в файл.

2. Указания к выполнению работы

Работа с изображениями в Mathcad

Для чтения изображения из файла можно использовать следующие функции Mathcad:

READRGB ("file") — чтение цветного изображения;

READBMP ("file") — чтение изображения в оттенках серого;

Например, записать в переменную *MyPics* матрицу изображения из файла *MyPics.bmp* в оттенках серого можно следующим образом:

MyPics:=READBMP("C:\Program Files\Mathcad\MyPics.bmp")

Чтобы просмотреть изображение можно воспользоваться элементом «Изображение» с панели инструментов *Matrix*.

Чтобы инвертировать изображение, надо поменять уровень яркости таким образом, чтобы в каждой точке изображения он был бы равен разности числа 255 и исходного уровня.

Для записи изображения в файл можно использовать следующие функции Mathcad:

WRITERGB("file") — запись цветного изображения;

WRITEBMP ("file") — запись изображения в оттенках серого;

Например, запишем рисунок из переменной *Negativ* обратно в файл.

WRITEBMP("C:\Program Files\Mathcad\MyPics-neg.bmp"):=Negativ

Работа со структурированными файлами (матрицами)

Считывание матрицы с помощью функции READPRN

Функция READPRN считывает файл данных (ASCII-файл) целиком (каждый раз), определяет число строк и столбцов, и создает матрицу из этих данных.

При использовании функции READPRN необходимо учитывать особенности:

1. Все строки в файле данных должны содержать одинаковое количество значений. Строки, не содержащие значений, игнорируются. Если строки в файле имеют различное число значений, *Mathcad*, отмечает функцию READPRN сообщением “ошибка файла”.

2. Функция READPRN игнорирует текст в файле данных.

3. Результатом чтения файла данных является $m \times n$ матрица, где m – число строк, содержащих данные в файле, n – число значений в строке.

4. Чтобы создать матрицу из значений в файле данных, необходимо использовать запись вида $M := \text{READPRN}(\text{file})$.

5. Разделителем значений в строке могут быть пробелы, знаки табуляции или запяты.

Запись структурированных данных (матрицы) в файл осуществляется функцией WRITEPRN . Например, $\text{WRITEPRN}(\text{file}):=M$. На результат записи влияют значения переменных: PRNPRECISION – количество знаков после запятой при записи в файл (по умолчанию 4), PRNCOLWIDTH – ширина столбца в символах (по умолчанию 8). Необходимо помнить следующие особенности использования функции WRITEPRN :

1. Ширина столбца (PRNCOLWIDTH) должна быть такой, чтобы разместились все необходимые цифры (PRNPRECISION) вместе с пробелом, разделяющим отдельные значения

2. Каждый новый вызов функции WRITEPRN записывает новый файл. Для «дозаписи» данных в файл используют функцию APPENDPRN .

3. Если записываемый массив является составным (его элементы являются сами массивами), либо комплекснозначным, то тогда функция WRITEPRN создаст не просто ASCII-файл, а файл специального формата, прочтение которого другими программами не гарантируется.

3. Содержание работы

1. Обработка растровых изображений в *Mathcad*
2. Чтение структурированных данных из файла, их обработка и запись в файл

4. Порядок выполнения работы

Задание 1. Реализуйте средствами *Mathcad* выполнение следующей задачи: «Дано изображение в формате *bmp*. Замените правую половину изображения левой, отражённой по вертикали. Выполните инверсию (негатив) правой (изменённой) части изображения. Сохраните изображение в файл под новым именем».

Задание 2. Реализуйте средствами *Mathcad* чтение структурированного файла (матрица чисел), обработку полученных данных и их запись в новый файл (согласно варианту).

5. Варианты заданий

Варианты к заданию 2:

1. Оценка результатов тестирования по математике.
2. Статистическая обработка результатов тестирования.
3. Обработка результатов экзаменационной сессии.
4. Обработка данных об электропотреблении.

5. Обработка данных о результатах соревнований.
6. Расчёт затрат на оплату услуг электроснабжения.
7. Расчёт затрат на оплату сверхнормативного электропотребления.
8. Обработка данных о перемещении объекта в пространстве.
9. Обработка данных прогноза погоды.
10. Обработка результатов турнира по футболу.
11. Обработка данных о работе приёмной комиссии.
12. Сравнение результатов работы приёмной комиссии.
13. Обработка данных о маршруте движения туристической группы.

6. Контрольные вопросы

- Чтение структурированных данных из файла.
- Запись структурированных данных в файл.
- Чтение изображения из файла. Запись изображения в файл.
- Алгоритм поиска наибольшего элемента в матрице
- Определение количества строк и столбцов в матрице.
- Алгоритм подсчета количества положительных элементов в матрице.
- Алгоритм подсчёта суммы элементов в столбце.

7. Содержание отчета

Работу оформите в *Mathcad*, снабдив каждое задание нумерованным заголовком и кратким комментарием, а также добавьте титульный лист в начало работы. Отчёт должен содержать выводы по работе, которые необходимо разместить на отдельном листе.

Лабораторная работа 6. Моделирование в Mathcad

1. Цель работы

Изучение способов решения дифференциальных уравнений в *Mathcad*, развитие навыков использования математических пакетов для построения и расчёта аналитических моделей физики в дифференциальных уравнениях.

2. Указания к выполнению работы

Во многих вопросах геометрии, физики, механики, естествознания, техники и т.п. играют большую роль дифференциальные уравнения. Так называются уравнения, связывающие между собой независимую переменную x , искомую функцию y и ее производные различных порядков. Для решения дифференциальных уравнений в *Mathcad* можно использовать функцию *odesolve*.

Функция *odesolve*

Встроенная функция *odesolve* предназначена для решения дифференциальных уравнений линейных относительно старшей производной, записанных в общепринятом в математической литературе виде.

Функция *odesolve* методом Рунге-Кутты с фиксированным шагом решает задачу Коши $y(x_0) = y_0, y'(x_0) = y_{0,1}, \dots, y^{(n-1)}(x_0) = y_{0,n-1}$ или простейшую граничную задачу $y^{(k)}(a) = y_{a,k}, y^{(m)}(b) = y_{b,k}, 0 \leq k \leq n-1, 0 \leq m \leq n-1$ для уравнений вида $F(x, y, y', \dots, y^{(n-1)}) = f(x)$.

Обращение к функции имеет вид $Y := \text{odesolve}(x, b, \text{step})$ или $Y := \text{odesolve}(x, b)$, где Y – имя функции, содержащей значения найденного решения, x – переменная интегрирования, b – конец промежутка интегрирования, step – шаг, который используется при интегрировании уравнения методом Рунге-Кутты.

Перед обращением к функции *odesolve* необходимо записать ключевое слово *Given*, затем ввести уравнение и начальные либо граничные условия. При вводе уравнения и условий задачи используется знак символического равенства (логическое равно – $\langle \text{Ctrl} \rangle + \langle = \rangle$), а для записи производных можно использовать как оператор дифференцирования (находится на панели «Математический анализ»), так и знак производной ($\langle \text{Ctrl} \rangle + \langle \text{F7} \rangle$). При этом необходимо обязательно записывать аргумент искомой функции.

Также для решения задачи Коши в *Mathcad* могут использоваться следующие функции:

- $\text{rkfixed}(y, x1, x2, \text{npoints}, D)$ – решение задачи на отрезке методом Рунге-Кутты с постоянным шагом;

- $Rkadapt(y, x1, x2, noints, D)$ – решение задачи на отрезке методом Рунге-Кутты с автоматическим выбором шага;
- $rkadapt(y, x1, x2, acc, noints, D, kmax, save)$ – решения задачи в заданной точке методом Рунге-Кутты с автоматическим выбором шага;
- $Bulstoer(y, x1, x2, noints, D)$ – решение задачи на отрезке методом Булирша-Штера;
- $bulstoer(y, x1, x2, acc, noints, D, kmax, save)$ – решение задачи в заданной точке методом Булирша-Штера;
- $Stiff(r, x1, x2, acc, D, J)$ – решение задачи для жестких систем на отрезке с использованием алгоритма Розенброка;
- $stiff(r, x1, x2, acc, D, J, kmax, save)$ – решения задач для жестких систем на отрезке с использованием алгоритма Розенброка;
- $Stiffb(y, x1, x2, acc, D, J)$ – решение задачи для жестких систем на отрезке

3. Содержание работы

- 1 Решение дифференциальных уравнений в Mathcad
- 2 Построение модели физической задачи в дифференциальных уравнениях и её решение в Mathcad

4. Порядок выполнения работы

- Задание 1. Решите дифференциальное уравнение согласно варианту.
 Задание 2. Составьте модель физической задачи согласно варианту. Реализуйте её в *Mathcad*. Оцените полученные результаты.
 Задание 3. Сделайте выводы о проделанной работе.

5. Варианты заданий

Таблица 6.1 – Варианты задания к заданию 1.

Номер варианта	Уравнение $f(x,y)$	Начальные условия	Интервал для поиска решения	Шаг изменения
1	$\frac{y}{\cos(x) \cdot \ln(y)}$	$y(1)=1$	[1,10]	1
2	$\text{tg}(x)\text{tg}(y)$	$y(0)=0$	[0,5]	0.5
3	$\frac{y}{1+x^2}$	$y(1)=1$	[1,7]	
4	$\sin(3x)-y \cdot \text{tg}(3x)$	$y(0)=1/3$	[0,4]	0,25
5	$\cos(x-2y)-\cos(x+2y)$	$y(0)=\pi/4$	[0,4 π]	$\pi/2$
6	$\sqrt{x^2 + y \cos(x) }$	$y(0)=3,6$	[4,1;6,7]	0,1

Окончание табл. 6.1.

Номер варианта	Уравнение $f(x,y)$	Начальные условия	Интервал для поиска решения	Шаг изменения
7	$\sin(x)+\cos(y^2)$	$y(0)=2,2$	$[0,8;3,2]$	0,1
8	$0,7y+x \cdot \ln(x+y)$	$y(0)=2,5$	$[12,4;14,1]$	0,08
9	$\sqrt{x^2 + e^{xy}}$	$y(0)=1,6$	$[5,2;6,8]$	0,1
10	$y/\ln(y)$	$y(2)=1$	$[2;5]$	0,25
11	$-\frac{\sqrt{1+\cos(2x)}}{\sqrt{1+\sin(y)}}$	$y(\pi/4)=0$	$[\pi/4, 3\pi]$	$\pi/8$
12	$\frac{1}{x^2} - \frac{1}{y}$	$y(1)=0$	$[1;4]$	0.3

Варианты к заданию 2.

2.1 Движение тела, брошенного под углом к горизонту (без учета сопротивления воздуха)

2.2 Колебания математического маятника (без затуханий)

2.3 Моделирование артиллерийской задачи

2.4 Движение подводной лодки (подъём)

2.5 Упругое столкновение шаров

2.6 Движение небесного тела в гравитационном поле

2.7 Движение материального тела в поле тяготения

2.8 Колебания пружинного маятника

2.9 Колебания физического маятника

2.10 Движение брошенного тела, с учетом сопротивления воздуха

6. Контрольные вопросы

- Понятие производной.
- Что называется обыкновенным дифференциальным уравнением?
- Сформулируйте определение задачи Коши для уравнения.
- Что есть решение обыкновенного дифференциального уравнения?
- Какие средства *Mathcad* можно использовать для решения обыкновенных дифференциальных уравнений и их систем?
- Назовите и опишите методы приближенного решения обыкновенных дифференциальных уравнений?

7. Содержание отчета

Работу оформите в *Mathcad*, снабдив каждое задание нумерованным заголовком и кратким комментарием, а также добавьте титульный лист в начало работы. Отчёт должен содержать выводы по работе, которые необходимо разместить на отдельном листе.

Лабораторная работа 7. Основы программирования в Matlab

1. Цель работы

Знакомство с интерфейсом среды *Matlab*, приобретение навыков создания исполняемых программ на языке *Matlab*.

2. Указания к выполнению работы

Matlab (MATrix LABoratory) – интерактивный матрично-ориентированный пакет, предназначенный для выполнения научных и инженерных расчетов.

По умолчанию после запуска пакета *Matlab* на экране появляется комбинированное окно, включающее четыре наиболее важные панели (рис. 1.2):

1 *Command Window* (Окно команд) – самая используемая панель. В ней набирают команды пользователя, подлежащие немедленному исполнению. Здесь же выдаются результаты выполненных команд.

2 *Command History* (История команд) хранит все команды, набираемые пользователем, однако в отличие от содержимого *Command Window* (Окно команд) сюда не попадают сообщения системы и результаты вычислений.

3 *Workspace* (Рабочее пространство) отображает текущий набор переменных, заведенных пользователем в командном окне.

4 *Current Directory* (Текущий каталог) является аналогом известной программы Проводник, но имеет для MATLAB свое особое предназначение. Дело в том, что, кроме работы с математическими выражениями из командного окна, пользователь также может работать с файлами

Для написания текста программ в MATLAB используется редактор М-файлов. При помощи редактора можно создавать собственные функции, вызывать их, в том числе и из командной строки.

Файлы программ сохраняются в MATLAB с расширением *.m. Комментарии в тексте программы начинаются со знака %.

В MATLAB М-файлы делятся на два вида: файл- функции (*functions*) и файл-сценарии (*scripts*). Сценарии содержат определенную последовательность команд, не имеют входных и выходных аргументов. Функции напротив производят определенные действия с входными аргументами и возвращают результат в выходных аргументах.

Слово *function* определяет, что данный файл содержит функцию. Первая строка является заголовком функции, в которой размещается имя функции и список входных и выходных аргументов.

Далее следует тело функции

```
function f=myfunction(x)
f=log(x)/sin(x)
```

В рассмотренной функции с именем *myfunction* входным аргументом является *x*, а результат вычислений – это выходной аргумент *f*. Файл- функцию следует сохранять под тем же именем, которое указано в первой строке файла. Созданную функцию можно вызвать в командной строке или в другой функции.

```
>>myfunction(2)
f = 0.7623
```

Входных и выходных аргументов может быть несколько function [X, Y, Z] = name (x, y, z)

Операторы цикла

Цикл *for*

Этот цикл предназначен для выполнения определенного числа повторяющихся действий. Его структура выглядит следующим образом

```
for count = start : step : final
    команды MATLAB
end
```

В данном примере *count* – счетчик цикла, *start* – начальное значение счетчика, *step* – шаг изменения счетчика, *final* – конечное значение счетчика. Счетчик может принимать не только целые значения, но и вещественные.

Цикл *while*

Цикл *while* используется тогда, когда неизвестно количество необходимых итераций в теле цикла. Он выполняется до тех пор, пока выполняется условие цикла:

```
while условие цикла
    команды MATLAB
end
```

Условный оператор *if*

Условный оператор *if* применяется для реализации разветвленных алгоритмов. Обобщенная его структура такова:

```
if условие
    команды MATLAB
elseif условие
    команды MATLAB
else
    команды MATLAB
end
```

Оператор *switch*

Оператор переключения *switch* позволяет разделить алгоритм программы на равнозначные ветви, каждая из которых выполняется, когда пе-

ременная оператора принимает одно из значений из списка. После выполнения какой-либо из ветвей происходит выход из *switch*. Если подходящих значений для переменной не нашлось, то выполняется ветвь программы, соответствующая *otherwise*.

Пример:

```
switch a
case -1
    disp ('a = -1')
case 0
    disp ('a = 0')
case 1
    disp ('a = 1')
```

Прерывание цикла. Исключительные ситуации

При организации циклических вычислений желательно, чтобы внутри цикла не возникало ошибок. Предотвратить выполнение цикла можно с помощью оператора *break*. Он заканчивает цикл и переходит к оператору *end* и осуществляются вычисления, следующие за *end*.

Если в программе появляются ошибки, приводящие к окончанию программы, то для избежания выхода из программы следует применить оператор *try...catch*. Структура его такова:

try

операторы, выполнение которых может привести к ошибке

catch

операторы, которые следует выполнить при возникновении ошибки в блоке между *try* и *catch*

3. Содержание работы

1. Работа в командном окне Matlab
2. Создание скрипта в Matlab
3. Создание функции в Matlab
4. Создание программы на языке Matlab

4. Порядок выполнения работы

Задание 1. Вычислите указанные арифметические выражения.

Задание 2. Проведите вычисления по заданной формуле при указанных значениях параметров.

Задание 3. Создайте m-функцию для расчёта функции вида $y=f(x)$ в заданной точке. Постройте график этой функции.

Задание 4. Создайте m-скрипт для: 1) для аргумента x , вводимого с клавиатуры, вычислить значения y (через функцию из задания 2); 2) для массива x вычислить значения y и вывести график.

Задание 5. Создайте m-функцию, содержащую простой цикл и решающий задачу согласно варианту.

5. Варианты заданий

Варианты для задания 1 представлены в табл. 7.1.

Таблица 7.1 – Варианты к заданию 1

1. $\frac{\left(12\frac{1}{6} - 6\frac{1}{27} - 5,25\right)13,5 + 0,111}{0,02}$	6. $\frac{\left(58\frac{4}{15} - 56\frac{7}{24}\right) : 0,8 + 2\frac{1}{9} \cdot 0,225}{8,75 \cdot 0,6}$	11. $\frac{\left(4,5 \cdot \frac{2}{3} - 6,75\right) \cdot 0,6}{\left(3,333 \cdot 0,3 + 0,222 \cdot \frac{4}{9}\right) 2\frac{2}{3}}$
2. $\frac{\left(\frac{1}{12} + 2\frac{5}{32} + \frac{1}{24}\right) : 9,6 + 2,13}{0,0004}$	7. $\frac{\left(\frac{0,216}{0,15} + 0,56\right) : 0,5}{\left(7,7 : 24,75 + \frac{2}{15}\right) 4,5}$	12. $\frac{\left(5\frac{4}{45} - 4\frac{1}{6}\right) : 5\frac{8}{15} \cdot 34\frac{2}{7}}{\left(4\frac{2}{3} + 0,75\right) 3\frac{9}{13}}$
3. $\frac{\left(6,6 - 3\frac{3}{14}\right) 5\frac{5}{6}}{(21 - 1,25) : 2,5}$	8. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40}\right) 1,6}$	13. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40}\right) 1,68}$
4. $\frac{2,625 - \frac{2}{3} \cdot \frac{5}{14}}{\left(3\frac{1}{12} + 4,375\right) : 19\frac{8}{9}}$	9. $\frac{\left(\frac{3}{5} + 0,425 - 0,005\right) : 0,12}{30,5 + \frac{1}{6} + 3\frac{1}{3}}$	14. $\frac{(68,023 - 66,028) : 6\frac{1}{9} + \frac{7}{40} \cdot 4,5}{0,042 + 0,086}$
5. $\frac{0,134 + 0,05}{18\frac{1}{6} - 1\frac{11}{14} - \frac{2}{15} \cdot 2\frac{6}{7}}$	10. $\frac{0,725 + 0,42(6)}{0,128 - 6,25 - (0,0345 : 0,12)} \cdot 0,25$	15. $\frac{(21 - 1,965) : (12 \cdot 0,045)}{0,00325 : 0,013} - \frac{4}{0,2 \cdot 0,73}$

Варианты для задания 2 представлены в табл. 7.2.

Таблица 7.2 – Варианты для задания 2

Выражение	Значения параметров	
1. $3m^2 + \sqrt[3]{2n^2} : m$;	a) $m = -\frac{14}{5}, n = tg \frac{\pi}{8}$;	б) $m = 2,2 \cdot 10^{-2}, n = \frac{1}{31}$.
2. $\frac{4}{3} l^3 \sin^2 \frac{\alpha}{2} \sqrt{\cos \alpha}$;	a) $l = 1,7 \cdot 10^3, \alpha = 18^\circ$;	б) $l = \frac{16}{21}, \alpha = \frac{\pi}{5}$
3. $\sqrt{\frac{a\sqrt{b}}{tg \alpha}}$;	a) $a = 1,5, b = 0,8, \alpha = 61^\circ$;	б) $a = 3 \cdot 10^{-2}, b = 0,71, \alpha = \frac{3}{7}\pi$
4. $\frac{3a^2 \sqrt{6,8 \cdot (a-b)}}{4(a+b)^3}$	a) $a = 4,13 \cdot 10^{-1}, b = \frac{1}{261}$;	б) $a = \sin \frac{5\pi}{8}, b = -tg 12^\circ$
5. $\frac{c^3}{6} \cos \frac{\alpha}{2} \sqrt{\sin \alpha}$	a) $c = \lg 2,38, \alpha = \frac{\pi}{5}$;	б) $c = e^{-0,3}, \alpha = 65^\circ$
6. $\sqrt{\frac{n^3}{16,3 \sin \alpha \sin 2\alpha}}$;	a) $n = 3,1516 \cdot 10^{-2}, \alpha = 5^\circ$;	б) $n = e^{3,5}, \alpha = \frac{2\pi}{13}$
7. $5 \sin 35^\circ \sqrt{\frac{S^3 \cos 36^\circ}{\pi^3 tg \alpha}}$;	a) $S = \ln 3, \alpha = 44^\circ$;	б) $S = \frac{18}{25}, \alpha = \frac{7}{12}\pi$

Окончание табл.7.2

Выражение	Значения параметров	
8. $ \lg(1 + \sin \alpha) + \ln(1 - \sin \beta) $	а) $\alpha = \frac{3\pi}{7}, \beta = 83^\circ$	б) $\alpha = \frac{2}{3}\pi, \beta = 16^\circ$
9. $\sqrt[3]{\sin^2(\alpha + \beta) - \sin^2(\alpha - \beta)}$	а) $\alpha = \frac{5}{7}\pi, \beta = 0,3\pi$	б) $\alpha = 12^\circ, \beta = 220^\circ$
10. $(\log_a(b + 1,4))^{-\frac{3}{4}}$;	а) $a = 3,56, b = e^{0,316}$;	б) $a = 2, b = 2,1649 \cdot 10^{-2}$
11. $3\left(p^{-\frac{2}{3}} + q^{-\frac{1}{2}}\right)\sqrt[3]{pq}$;	а) $p = \ln 3, q = \lg 3$;	б) $p = 0,013, q = 1,4 \cdot 10^2$.
12. $\frac{2}{3}m\sqrt{m^3\sqrt{m^4\sqrt{m}}}$;	а) $m = 3,6485 \cdot 10^2$;	б) $m = \frac{24}{37}$
13. $\frac{8}{3}S\sqrt{\frac{S}{\pi}\sin^6\frac{\alpha}{2}}$;	а) $S = e^{111}, \alpha = \frac{7}{11}\pi$	б) $S = 5,403, \alpha = 28^\circ$
14. $2\sqrt{\frac{F}{\pi}\operatorname{tg}\alpha\sin^2\frac{\alpha}{2}}$;	а) $F = \frac{1}{0,03}, \alpha = \frac{5}{7}\pi$	б) $F = \ln 7, \alpha = 1,34^\circ$
15. $\frac{1}{12} \cdot \frac{m^3 \cos \alpha}{(\sin \alpha + \cos \alpha)^3}$;	а) $m = -20,1, \alpha = 20^\circ$;	б) $m = \lg 13,6, \alpha = 1,48$

Варианты для задания 3 представлены в табл. 7.3.

Таблица 7.3 – Варианты к заданию 3

Функция	x_0	Функция	x_0
1. $y = \frac{e^{-x}}{\sqrt{e^{-x} + 1}} - x$	5.5	7. $y = e^{-x^3}\sqrt{x^2} - x - 1$	2.35
2. $y = \frac{x^2 - 1}{\ln(x^2 - 1)} + x$	2.75	8. $y = x^3\sqrt{(1-x)^2} - 1$	8.29
3. $y = \operatorname{sh}x + \sin x - 1$	3.1	9. $y = e^{-x}\sqrt{1+x+x^2} - x^2$	4.56
4. $y = \frac{1}{x(1-\ln x)} - 2$	4.21	10. $y = \sqrt{x} - 1 - \cos(0,5x)$	1.23
5. $y = \frac{2}{3}\sin^2 2x - \frac{3}{4}\cos^2 2x$	6.32	11. $y = x^2 \ln(1+x^2) - x$	7.55
6. $y = \frac{\sin^2 x}{\cos^4 x} - 1$	4.75	12. $y = e^{\sqrt{\sin x}} - 1,5$	3.64

Варианты для задания 5:

1. Нахождение обратной матрицы любого размера.
2. Транспонирование матрицы любого размера. Нахождение определителя матрицы.
3. Заполнить элементы матрицы размера $m \times m$ в шахматном порядке случайными числами из заданного диапазона.
4. Поиск наименьшего в матрице любого размера (задается при вызове).

5. Вычисление элемента ряда $1/(n!+n^2)$ с точностью eps. (Вычисление должно закончиться, когда рассчитанный элемент ряда будет меньше eps, при этом n возрастает на каждой итерации).

6. Посчитать количество положительных или отрицательных (задается при вызове) элементов в матрице любого размера.

6. Контрольные вопросы

- Опишите назначения основных окон в Matlab
- Опишите редактор М-файлов в MATLAB.
- Объясните отличия файла-функции от файла-сценария?
- Как задаются входные и выходные аргументы функции?
- В чем отличие цикла for от цикла while?
- Каков синтаксис оператора ветвления в MATLAB?
- Опишите основные этапы выполненной работы. применяемые функции, операторы.

7. Содержание отчета

Отчёт должен содержать: 1) описание исходных данных варианта; 2) последовательность действий для решения задачи; 3) результаты решения задачи; 4) выводы по работе, 5) приложение – листинги созданных программ с комментариями.

Лабораторная работа 8. Работа с матрицами в Matlab. Решение систем алгебраических уравнений

1. Цель работы

Знакомство с основными операторами и функциями, предназначенными для работы с матрицами; формирование навыков использования программных средств для решения систем уравнений.

2. Указания к выполнению работы

MATLAB работает практически с одним видом объектов – с числовыми прямоугольными матрицами, элементами которых могут быть в общем случае комплексные числа. Все переменные представляют собой матрицы. В некоторых случаях матрицы 1×1 интерпретируются как скаляры, а матрицы с одной строкой или одним столбцом интерпретируются как вектора. В системе MATLAB матрицы могут быть созданы разными способами:

- введены явно с помощью списка элементов;
- сгенерированы встроенными операторами или функциями;
- созданы в m-файлах (см. разд. П1.6);
- загружены из внешнего файла данных.

Явное задание матриц – например, в результате выполнения любого из приведенных далее операторов

```
>> A=[1 2 3; 4 5 6; 7 8 9] <Enter>
```

В результате выполнения команды MATLAB создаст соответствующую матрицу 3×3 .

Элементы внутри строки матрицы могут отделяться друг от друга не только пробелами, но и запятыми. Ввод больших матриц лучше выполнять с помощью m-файлов, в которых легко находить и исправлять ошибки. В пакет MATLAB встроен ряд функций, позволяющих создавать функции специального вида, например, *rand*, *magic*, *zeros* и *ones* позволяют легко сгенерировать матрицы.

- Функция *rand* (n) создает матрицу размером $n \times n$, каждый элемент которой – случайное число с равномерным законом распределения в диапазоне [0, 1].
- Функция *rand*(m,n) создает матрицу размера $m \times n$, каждый элемент которой – случайное число с равномерным законом распределения в диапазоне [0, 1].
- Функция *magic*(n) создает матрицу размером $n \times n$, которая является магическим квадратом (суммы элементов по строкам и столбцам равны).

- Функция `zeros (m,n)` создает нулевую матрицу размера $m \times n$.
- Функция `ones (m,n)` создает матрицу размера $m \times n$, каждый элемент которой равен единице.

Ссылки на отдельные элементы матриц и векторов осуществляются с помощью индексов в круглых скобках. Например, $A(1,3)$ означает элемент матрицы A , стоящий на 1-й строке и 3-м столбце матрицы a , а $x(3)$ означает 3-й элемент вектора x . В качестве индексов векторов и матриц могут использоваться только положительные числа. Ссылаться на элементы матрицы A можно, используя единственный индекс $A(k)$. В этом случае данная матрица рассматривается как один длинный вектор-столбец, сформированный из столбцов исходной матрицы. Например, обратиться ко второму элементу второй строки матрицы a можно, указав $A(2,2)$ или $A(5)$.

Подматрицы и использование двоеточия

Для записи алгоритмов сложной обработки данных в компактной форме в системе MATLAB используются векторы и подматрицы. Например, выражение `1:5` фактически является вектор-строкой `[1 2 3 4 5]`. Или, например, команда

```
>> x=0.2:0.2:1.2
```

создает вектор $x = [0.2, 0.4, 0.6, 0.8, 1.0, 1.2]$.

Двоеточие само по себе означает всю строку или весь столбец, например:

```
>> A (:, 3)
```

Матричные операции

В пакете MATLAB доступны следующие матричные операции: сложение (+); вычитание (-); умножение (*); возведение в степень (^); транспонирование ('); левое деление (\); правое деление (/).

Операции с массивами

Матричные операции сложения и вычитания действуют поэлементно, а остальные приведенные ранее операции – нет, они являются матричными операциями. Следует отметить, что приведенные ранее операции: умножение (*), возведение в степень (^), левое деление (\), правое деление (/) – могут стать поэлементными, если перед ними поставить точку.

3. Содержание работы

- 1 Работа с матрицами в командном окне
- 2 Решение системы линейных алгебраических уравнений встроенными в Matlab функциями
- 3 Решение СЛАУ матричным методом
- 4 Написание программы для решения СЛАУ методами Гаусса, Крамера, простых итераций, Зейделя

4. Порядок выполнения работы

Задание 1. Задайте две матрицы 3-го порядка с помощью функции `rand`. Объедините их в новую матрицу двумя способами с помощью `[]` и функции `cat` сначала по горизонтали, а потом по вертикали.

Задание 2. Задайте квадратную матрицу 5-го порядка с помощью функции `rand`. Удалите две первые строки в этой матрице, а затем последний столбец.

Задание 3. Выделите из квадратной матрицы 3-го порядка главную диагональ, побочную диагональ, (+1)-ю диагональ – на 1-н уровень выше главной, (-1)-ю диагональ – на 1-н уровень ниже главной.

Задание 4. Создайте матрицу 8-го порядка из 1 и 0 в шахматном порядке (`hermat`), покажите графически структуру матрицы (`spy`).

Задание 5. Выполните преобразования матрицы.

Задание 6. Выполните решение СЛАУ встроенными в Matlab функциями (согласно варианту)

Задание 7. Выполните решение СЛАУ матричным методом.

Задание 8. Выполните решение СЛАУ численным методом.

Задание 9. Сравните результаты решения. Сделайте выводы.

5. Варианты задания

Варианты к заданию 5:

Вариант 1. Из матрицы 3-го порядка, заполненной случайными числами, создайте диагональную матрицу: а) 3-го порядка, чтобы на главной диагонали стояли элементы диагонали исходной матрицы; б) 3-го порядка, чтобы на главной диагонали стояли элементы побочной диагонали исходной матрицы; в) 9-го порядка, чтобы на главной диагонали стояли все элементы исходной матрицы; г) 9-го порядка, чтобы на побочной диагонали стояли все элементы исходной матрицы

Вариант 2. Из матрицы 3-го порядка, заполненной случайными числами, создайте диагональную матрицу: а) 3-го порядка, чтобы на главной диагонали стояли элементы диагонали исходной матрицы; б) 3-го порядка, чтобы на главной диагонали стояли элементы побочной диагонали исходной матрицы; в) 9-го порядка, чтобы на главной диагонали стояли все элементы исходной матрицы; г) 9-го порядка, чтобы на побочной диагонали стояли все элементы исходной матрицы.

Вариант 3. Создайте матрицу 5-го порядка, которая заполнена целыми случайными числами в диапазоне от 1 до 10. Преобразуйте исходную матрицу так, чтобы: а) на главной диагонали стояли нули; б) на главной диагонали стояли единицы; в) на главной диагонали, на диагоналях выше и ниже главной стояли пятёрки

Вариант 4. Создайте матрицу 8-го порядка из единиц. Преобразуйте исходную матрицу так, чтобы: а) она состояла из матриц 4-го порядка, содержащих единицы, двойки, тройки и четверки; б) она состояла из матриц 4-го порядка, содержащих $\sin(\pi/6)$, $\sin(\pi/5)$, $\sin(\pi/4)$, $\sin(\pi/3)$.

Вариант 5. Создайте матрицу 6-го порядка, заполненную целыми случайными числами в диапазоне от -9 до 9. Измените матрицу следующим образом: а) Обнулите элементы, значения которых четные. Найдите сумму элементов по столбцам, а затем по строкам; б) Положительные элементы приравняйте к десяти, а отрицательные - минус десяти. Найдите сумму элементов матрицы; в) От положительных элементов найдите \sin ; г) Отрицательные элементы возведите в квадрат.

Вариант 6. Создайте матрицу 3-го порядка, заполненную целыми случайными числами. Измените матрицу следующим образом: а) Преобразуйте матрицу в вектор-столбец. Обнулите все элементы, которые имеют нечетные значения. б) Обнулите все элементы, которые имеют четные значения. Преобразуйте матрицу в вектор-строку.

Варианты к заданиям 6-8 представлены в табл. 1.2 к лабораторной работе №1. При выполнении задания 8 для нечётного варианта используйте метод Гаусса и Зейделя, для чётного – Крамера и простых итераций.

6. Контрольные вопросы

- Создание векторов в системе Matlab.
- Создание матриц в системе Matlab. Специальные функции.
- Индексация векторов и матриц в системе Matlab, удаление, обнуление строк, столбцов.
- Поэлементные и матричные операции в системе Matlab.
- Получение транспонированных массивов.
- Объединение матриц.
- Использование оператора цикла для создания матриц.
- Решите СЛАУ (состоящую из 3-х уравнений) методом Гаусса.
- Что значит решить систему уравнений?
- Суть метода Гаусса для решения систем линейных уравнений.
- Суть метода простой итерации для решения систем уравнений.
- Суть метода Зейделя для решения систем уравнений.
- Суть метода Крамера для решения систем линейных уравнений.

7. Содержание отчета

Отчёт должен содержать: 1) описание исходных данных варианта; 2) последовательность действий для решения задачи; 3) результаты решения задачи; 4) выводы по работе, 5) приложение – листинги созданных программ с комментариями.

Лабораторная работа 9. Обработка табличных данных в Matlab

1. Цель работы

Знакомство со встроенными функциями *Matlab* для обработки табличных данных, развитие навыков создания алгоритмов и программ для обработки данных, включая решение задач аппроксимации и интерполяции.

2. Указания к выполнению работы

К простейшему анализу данных, содержащихся в некотором массиве, относится поиск его элементов с максимальным и минимальным значениями. Для этого определены следующие функции:

- $\max(A)$ возвращает наибольший элемент, если A – вектор; или возвращает вектор-строку, содержащую максимальные элементы каждого столбца, если A – матрица;
- $\max(A, B)$ возвращает массив того же размера, что A и B , каждый элемент которого есть максимальный из соответствующих элементов этих массивов;
- $\max(A, [], \text{dim})$ возвращает наибольший элемент по столбцам или по строкам матрицы в зависимости от значения скаляра dim . Например, $\max(A, [], 1)$ возвращает максимальные элементы каждого столбца матрицы A ;
- $[C, I] = \max(A)$ – кроме максимальных значений, возвращает вектор индексов элементов с этими значениями.

Многие операции статистической обработки данных выполняются быстрее, надежнее и точнее, если данные предварительно отсортированы. Ряд функций служит для выполнения сортировки элементов массива:

- $\text{sort}(A)$ в случае одномерного массива A сортирует и возвращает элементы по возрастанию их значений; в случае двумерного массива происходят сортировка и возврат элементов каждого столбца.
- $[B, \text{INDEX}] = \text{sort}(A)$ наряду с отсортированным массивом возвращает массив индексов INDEX .
- $\text{sort}(A, \text{dim})$ для матриц сортирует элементы по столбцам или по строкам в зависимости от значения переменной dim .
- $\text{sortrows}(A)$ выполняет сортировку строк массива A по возрастанию и возвращает отсортированный массив, который может быть или матрицей, или вектором-столбцом.

К элементарной статистической обработке данных в массиве обычно относят нахождение их среднего значения, медианы (срединного значения) и стандартного отклонения. Для этого определены следующие функции:

- $\text{mean}(A)$ возвращает арифметическое среднее значение элементов массива, если A – вектор; или возвращает вектор-строку, содержащую средние значения элементов каждого столбца, если A – матрица.
- $\text{std}(X)$ возвращает стандартное отклонение элементов массива;
- $\text{corrcoef}(X)$ возвращает матрицу коэффициентов корреляции для входной матрицы, строки которой рассматриваются как наблюдения, а столбцы – как переменные.
- $\text{cov}(x)$ возвращает смещенную дисперсию элементов вектора x . Для матрицы, где каждая строка рассматривается как наблюдение, а каждый столбец – как переменная, $\text{cov}(x)$ возвращает матрицу ковариаций.

Интерполяция и аппроксимация данных

Под аппроксимацией обычно подразумевается описание некоторой, порой не заданной явно, зависимости или совокупности представляющих ее данных с помощью другой, обычно более простой или более единообразной зависимости. Часто данные находятся в виде отдельных узловых точек, координаты которых задаются таблицей данных. График аппроксимирующей функции может не проходить через узловые точки, но приближать их с некоторой (по возможности, малой) среднеквадратической погрешностью. Это характерно для регрессии – реализации метода наименьших квадратов (МНК).

Одна из наиболее известных аппроксимаций – полиномиальная аппроксимация.

- $\text{polyfit}(x, y, n)$ возвращает вектор коэффициентов полинома $p(x)$ степени n , который с наименьшей среднеквадратичной погрешностью аппроксимирует функцию $y(x)$.

В ряде случаев очень удобны сплайновая интерполяция и аппроксимация таблично заданных функций. При ней промежуточные точки ищутся по отрезкам полиномов третьей степени – это кубическая сплайновая интерполяция.

Для одномерной табличной интерполяции используется функция interp1 :

- $y_i = \text{interp1}(x, Y, x_i)$ возвращает вектор y_i , содержащий элементы, соответствующие элементам x_i и полученные интерполяцией векторов x и Y .
- $y_i = \text{interp1}(x, Y, x_i, \text{method})$ позволяет с помощью параметра method задать метод интерполяции:
 - 'nearest' – ступенчатая интерполяция;
 - 'linear' – линейная интерполяция (принята по умолчанию);
 - 'spline' – кубическая сплайн-интерполяция;
 - 'cubic' или 'pchip' – интерполяция многочленами Эрмита;

– 'v5cubic' – кубическая интерполяция MATLAB 5;

Все методы интерполяции требуют, чтобы значения x изменялись монотонно.

3. Содержание работы

1. Обработка данных массивов
 - 1.1. Поиск наименьшего и наибольшего элементов
 - 1.2. Сортировка массивов
2. Экспорт и импорт данных для обработки
3. Интерполяция и аппроксимация табличных функций средствами Matlab
4. Создание программы для интерполяции табличной функции методами Ньютона и Лагранжа
5. Создание программы для аппроксимации табличной функции методом наименьших квадратов

4. Порядок выполнения работы

Задание 1. Сгенерируйте массив случайных чисел длиной 100. Найдите наибольший и наименьший элементы, а также их индексы. Отсортируйте массив по возрастанию и убыванию.

Задание 2. Напишите m-функции для чтения числовых данных из файла и записи данных из переменной в файл в виде таблицы.

Задание 3. Произведите интерполяцию и аппроксимацию табличной функции (согласно варианту) встроенными в *Matlab* функциями. Сравните полученные результаты.

Задание 4. Создайте m-функции для интерполяции значения табличной функции в заданной точке полиномами Лагранжа и Ньютона. Сравните результаты между собой и с результатами, полученными при выполнении задания 3.

Задание 5. Создайте m-функцию для аппроксимации табличной функции методом наименьших квадратов (согласно варианту).

Задание 6. Оцените вид кривой наиболее точно приближающей табличную функцию. Используйте функцию `lsqnonlin` (реализующую метод Левенберга-Марквардта) для аппроксимации табличной функции кривой произвольного вида.

5. Варианты задания

Варианты для заданий 3-5 представлены в таблице 9.2

Таблица 9.2 – Варианты к заданиям 3-5

№	Задание								
1	x	1.20	1.57	1.94	2.31	2.68	3.05	3.42	3.79
	y	2.56	2.06	1.58	1.25	0.91	0.66	0.38	0.21
2	x	1.73	2.56	3.39	4.22	5.05	5.89	6.70	7.53
	y	0.63	1.11	1.42	1.96	2.30	2.89	3.29	3.87
3	x	-4.38	-3.84	-3.23	-2.76	-2.22	-1.67	-1.13	-0.60
	y	2.25	2.83	3.44	4.51	5.29	6.55	8.01	10.04
4	x	1.00	1.64	2.28	2.91	3.56	4.29	4.84	5.48
	y	0.28	0.19	0.15	0.11	0.09	0.08	0.07	0.06
5	x	5.89	3.84	6.19	9.22	7.87	6.29	4.43	8.91
	y	79.31	57.43	60.66	90.55	92.12	71.30	70.50	91.25
6	x	2.91	2.94	6.35	6.58	3.80	6.43	0.57	5.96
	y	82.16	61.02	44.56	82.52	99.19	70.24	63.23	66.48
7	x	1.23	1.79	2.24	2.76	3.20	3.68	4.16	4.64
	y	2.10	2.84	3.21	3.96	4.86	6.06	7.47	9.25
8	x	-4.38	-3.84	-3.23	-2.76	-2.22	-1.67	-1.13	-0.60
	y	1.73	2.56	3.39	4.22	5.05	5.89	6.70	7.53
9	x	2.56	2.06	1.58	1.25	0.91	0.66	0.38	0.21
	y	0.63	1.11	1.42	1.96	2.30	2.89	3.29	3.87
10	x	79.31	57.43	60.66	90.55	92.12	71.30	70.50	91.25
	y	5.89	3.84	6.19	9.22	7.87	6.29	4.43	8.91
11	x	2.10	2.84	3.21	3.96	4.86	6.06	7.47	9.25
	y	1.00	1.64	2.28	2.91	3.56	4.29	4.84	5.48
12	x	0.28	0.19	0.15	0.11	0.09	0.08	0.07	0.06
	y	82.16	61.02	44.56	82.52	99.19	70.24	63.23	66.48

В задании 3 для нечётного варианта использовать ступенчатую и кубическую интерполяцию, для чётного – сплайн-интерполяцию и многочлены Эрмита.

В задании 5 для нечётного варианта использовать линейную приближающую функцию, для чётного – квадратичный трёхчлен.

6. Контрольные вопросы

- Функции для поиска наименьшего и наибольших элементов в массиве.
- Функции для сортировки элементов в массиве.
- Понятие аппроксимации.
- Что такое интерполяция?
- Что такое узлы интерполяции?

- В чем заключается задача отыскания интерполирующего многочлена?
- Как построить интерполяционный многочлен Лагранжа?
- Что такое сплайн? Как происходит процесс интерполирования сплайнами?
- Что такое конечная разность первого порядка? Как она находится?

7. Содержание отчета

Отчёт должен содержать: 1) описание исходных данных варианта; 2) последовательность действий для решения задачи; 3) результаты решения задачи; 4) выводы по работе, 5) приложение – листинги созданных программ с комментариями.

Лабораторная работа 10. Работа с файлами данных в Matlab

1. Цель работы

Изучение способов чтения и записи данных различных форматов в файл средствами Matlab.

2. Указания к выполнению работы

MATLAB имеет программные средства для обработки не только числовой, но и различной другой информации. Прежде всего это строки – цепочки из алфавитных символов. Эти средства широко применяются для создания баз данных, условных выражений, текстовых комментариев и т.д.

Основные функции обработки строк

В основе представления символов в строках лежит их кодирование с помощью сменных таблиц кодов. Такие таблицы ASCII американского кода представления информации в информационных системах ставят в однозначное соответствие каждому символу некоторый код со значением от 0 до 255.

Вектор, содержащий строку символов, в системе MATLAB задается следующим образом:

S = 'Any Characters' – вектор, компонентами которого являются числовые коды, соответствующие символам.

К основным строковым функциям относятся следующие.

- char(X) преобразует массив X положительных целых чисел (числовых кодов от 0 до 65 535) в массив символов системы MATLAB (причем только первые 127 кодов – английский набор ASCII, со 128 до 255 – расширенный набор ASCII) и возвращает его. Примеры:

```
>> X=reshape(32:127,32,3);
```

```
>> S = char(X')
```

```
S =
```

```
!>#%&'()*+,-./0123456789:;<=>?
```

```
@ABCDEFGHIJKLMN O PQRSTU VWXYZ[\]^_
```

```
“abcdefghijklmnopqrstu vwxyz {}~•
```

- double(S) преобразует символы строки S в числовые коды 0–65 535 и возвращает вектор с этими числовыми кодами.

- ischar(S) возвращает логическую единицу, если S является символьной переменной, и логический нуль в противном случае.

Операции над строками

- findstr(str1,str2) обеспечивает поиск начальных индексов более короткой строки внутри более длинной и возвращает вектор этих индексов.

- `lower('str')` возвращает строку символов `str`, в которой символы верхнего регистра переводятся в нижний регистр, а все остальные символы остаются без изменений.

- `upper('str')` возвращает строку символов `str`, в которой все символы нижнего регистра переводятся в верхний регистр, а все остальные символы остаются без изменений.

- `strcat(s1,s2,s3,...)` выполняет горизонтальное объединение соответствующих рядов массивов символов `s1`, `s2`, `s3` и т. д., причем пробелы в конце каждого ряда отбрасываются, и возвращает объединенную строку (ряд) результирующего массива символов, пробелы добавляются заново после анализа строк в полученном массиве. Все входные массивы должны иметь одинаковое число строк

- `strvcat(t1,t2,t3,...)` выполняет вертикальное объединение строк `t1`, `t2`, `t3`, ... в массив символов `S` аналогично `char(t1,t2,t3,...)`.

- `strcmp('str1','str2')` возвращает логическую единицу, если две сравниваемые строки `str1` и `str2` идентичны, и логический нуль в противном случае.

Преобразование символов и строк

- `int2str(X)` округляет элементы массива `X` до целых чисел и возвращает массив символов, содержащих символьные представления округленных целых чисел. Аргумент `X` может быть скаляром, вектором или матрицей.

- `num2str(A)` выполняет преобразование массива `A` в строку символов `str` с точностью до четырех десятичных разрядов и экспоненциальным представлением, если требуется. Обычно используется при выводе графиков совместно с `title`, `xlabel`, `ylabel` или `text`.

`str2double('str')` выполняет преобразование численной строки `s`, которая представлена в ASCII-символах, в число с двойной точностью. При этом `+` и `-` могут быть только в начале строки.

Работа с файлами

При обработке данных возникает необходимость хранения как исходных данных, так и результатов вычислений. Для этого обычно используются файлы. Файл обычно является некоторой совокупностью данных, объединенных одним именем. Тип файла, как правило, определяется его расширением. Перед использованием любого файла он должен быть открыт, а по окончании использования – закрыт. Много файлов может быть открыто и доступно для чтения одновременно. Рассмотрим команды открытия и закрытия файлов.

Команда **`open имя`**, где ***имя*** должно содержать массив символов или символьную переменную, открывает файлы в зависимости от анализа параметра ***имя*** и расширения в имени ***имя***:

- переменная – открывает массив, названный по имени, в редакторе массивов (Array Editor);
- .mat – открывает файл, сохраняет переменные в структуре в рабочей области;
- .fig – открывает его в редакторе дескрипторной графики Property Editor;
- .m – открывает m-файл в редакторе отладчика;
- .mdl – открывает модель в Simulink;
- .p – открывает, если он есть, m-файл с тем же именем;
- .html – открывает HTML-документ в браузере помощи.

Команда или функция delete удаляет файл или объект графики. **delete имя файла** удаляет файл текущей папки. Может быть использована *. Предпочтительно использование с записью в форме функции delete('имя файла'), когда имя файла – строка.

Функция close(H) закрывает только графические окна. Для закрытия файлов необходимо использовать команду **fclose**.

Для записи файлов на диск служит команда **save**.

Операции с двоичными файлами

Двоичными, или бинарными, называют файлы, данные которых представляют собой машинные коды.

fopen(filename, permission) открывает файл с именем filename и параметром, определенным в permission, и возвращает идентификатор fid со значением: 0 – чтение с клавиатуры (permission установлено в 'r'); 1 – вывод на дисплей (permission установлено в 'a'); 2 – вывод сообщения об ошибке (permission установлен в 'a'); –1 – неудача в открытии файла с выводом сообщения message о типе ошибки. Идентификатор fid часто используется в качестве аргумента другими функциями и программами ввода-вывода. Имя файла filename может содержать путь к файлу.

Параметр permission может принимать одно из следующих основных значений:

- 'r' – открытие файла для чтения (по умолчанию);
- 'r+' – открытие файла для чтения и записи;
- 'w' – удаление содержимого существующего файла или создание нового и открытие его для записи;
- 'a' – создание и открытие нового файла или открытие существующего для записи с добавлением в конец файла.

Работа с файлами изображений

Функция A = imread(filename,fmt) читает из файла с именем filename полутоновое или полноцветное изображение и создает A. Если исходное изображение полутоновое, то A – двумерный массив, если исходное изображение полноцветное, то A – трехмерный массив размера m×n×3.

Функция `imwrite(A,filename,fmt)` записывает изображение в файл с именем `filename` в формате `fmt` из массива `A`. `A` может быть матрицей размера $M \times N$ для полутонового изображения, и массивом размера $M \times N \times 3$ для полноцветного изображения.

Работа со звуковыми данными

`sound(Y,FS)` воспроизводит сигнал из вектора `Y` с частотой дискретизации `FS` с помощью колонок, подключенных к звуковой карте компьютера.

Для комплексной демонстрации возможностей работы со звуком служит файл-команда **xpsound**. Эта команда выводит диалоговое окно, которое позволяет выбрать несколько видов звукового сигнала, создать для них массив данных звука и воспроизвести звук (если компьютер оснащен звуковой картой, совместимой с Sound Blaster). Кроме того, имеется возможность графически отобразить временную зависимость звукового сигнала, его частотный спектр и спектрограмму.

Еще один весьма наглядный способ представления массива данных звуковых сигналов – это показ их спектрограммы. Звуковой сигнал при этом делится на множество фрагментов, а спектрограмма дает представление о распределении частот спектра в разные моменты времени.

Демонстрационные примеры можно просмотреть с помощью команды `type xpsound`.

3. Содержание работы

1. Обработка строковых данных
2. Работа с файлами
3. Работа с изображениями
4. Работа со звуковыми файлами

4. Контрольные вопросы

- Опишите процедуру работы с файлами в Matlab
- Опишите команды, используемые для работы с изображениями
- Опишите команды, используемые для работы с аудиофайлами

5. Содержание отчета

Отчёт должен содержать: 1) описание исходных данных варианта; 2) последовательность действий для решения задачи; 3) результаты решения задачи; 4) выводы по работе, 5) приложение – листинги созданных программ с комментариями.

Лабораторная работа 11. Визуализация данных в Matlab

1. Цель работы

Изучение способов визуализации данных в Matlab, настройке их отображения, а также экспорта и импорта.

2. Указания к выполнению работы

Функции одной переменной $y(x)$ находят широкое применение в практике математических и других расчетов, а также в технике компьютерного математического моделирования. Для отображения таких функций используются графики в декартовой (прямоугольной) системе координат. При этом обычно строятся две оси – горизонтальная X и вертикальная Y , и задаются координаты x и y , определяющие узловые точки функции $y(x)$. Эти точки соединяются друг с другом отрезками прямых, то есть при построении графика осуществляется линейная интерполяция для промежуточных точек.

Команда `plot(X,Y)` строит график функции $y(x)$, координаты точек (x,y) которой берутся из векторов одинакового размера Y и X . Если X или Y – матрица, то строится семейство графиков по данным, содержащимся в колонках матрицы.

Команда `plot(Y)` строит график $y(i)$, где значения y берутся из вектора Y , а i представляет собой индекс соответствующего элемента.

Команда `plot(X,Y,S)` аналогична команде `plot(X,Y)`, но тип линии графика можно задавать с помощью строковой константы S (посредством комбинации определённых символов задаётся цвет линии, тип точки, тип линии).

Команда `bar(x,Y)` строит столбцовый график элементов вектора Y (или группы столбцов для матрицы Y) со спецификацией положения столбцов, заданной значениями элементов вектора x , которые должны идти в монотонно возрастающем порядке;

Команда `bar(Y)` строит график значений элементов матрицы Y так же, как указано выше, но фактически для построения графика используется вектор $x=1:m$;

Если данные для построения функции определены с заметной погрешностью, то используют графики функций типа `errorbar` с оценкой погрешности каждой точки. Команда `errorbar(X,Y,L,U)` строит график значений элементов вектора Y в зависимости от данных, содержащихся в векторе X , с указанием нижней и верхней границ значений, заданных в векторах L и U .

Контурные графики

Контурные графики служат для представления на плоскости функции двух переменных вида $z(x,y)$ с помощью линий равного уровня. Они получаются, если трехмерная поверхность пересекается рядом плоскостей, расположенных параллельно друг другу. При этом контурный график представляет собой совокупность спроецированных на плоскость (x,y) линий пересечения поверхности $z(x,y)$ плоскостями. Типичный пример контурных графиков – обычные карты.

Для построения контурных графиков используются команды `contour`.

- `contour(Z)` строит контурный график по данным матрицы Z с автоматическим заданием диапазонов изменения x и y ;
- `contour(X,Y,Z)` строит контурный график по данным матрицы Z с указанием спецификаций для X и Y .

Создание массивов данных для трехмерной графики

Поверхности как объекты трехмерной графики обычно описываются функцией двух переменных $z(x,y)$. Специфика построения трехмерных графиков требует не просто задания ряда значений x и y , то есть векторов x и y . Она требует определения для X и Y двумерных массивов – матриц. Для создания таких массивов служит функция `meshgrid`. В основном она используется совместно с функциями построения графиков трехмерных поверхностей. Функция `meshgrid` записывается в следующих формах:

- `[X,Y] = meshgrid(x,y)` преобразует область, заданную векторами x и y , в массивы X и Y , которые могут быть использованы для вычисления функции двух переменных и построения трехмерных графиков. Строки выходного массива X являются копиями вектора x ; а столбцы Y – копиями вектора y ;

- `[X,Y] = meshgrid(x)` аналогична `[X,Y] = meshgrid(x,x)`;
- `[X,Y,Z] = meshgrid(x,y,z)` возвращает трехмерные массивы, используемые для вычисления функций трех переменных и построения трехмерных графиков.

Функция `ndgrid` является многомерным аналогом функции `meshgrid`.

Графики поверхностей

Команда `plot3(...)` является аналогом команды `plot(...)`, но относится к функции двух переменных $z(x,y)$. Она строит аксонометрическое изображение трехмерных поверхностей и представлена следующими формами:

- `plot3(x,y,z)` строит массив точек, представленных векторами x , y и z , соединяя их отрезками прямых. Эта команда имеет ограниченное применение;
- `plot3(X,Y,Z)`, где X , Y и Z – три матрицы одинакового размера, строит точки с координатами $X(i,:)$, $Y(i,:)$ и $Z(i,:)$ и соединяет их отрезками прямых.

- `plot3(X,Y,Z,S)` обеспечивает построения, аналогичные рассмотренным ранее, но со спецификацией стиля линий и точек, соответствующей спецификации команды `plot`.

Сетчатые 3D-графики с окраской

Команда `mesh(X,Y,Z,C)` выводит в графическое окно сетчатую поверхность $Z(X,Y)$ с цветами узлов поверхности, заданными массивом C .

Движение точки на плоскости

Для отображения движения точки по траектории используется команда `comet`. При этом движущаяся точка напоминает ядро кометы с длинным угасающим хвостом. Команда `comet(X,Y)` отображает движение «кометы» по траектории, заданной парой векторов Y и X и позволяет задавать длину хвоста кометы (отрезка траектории, выделенного цветом) как $p \cdot \text{length}(Y)$, где $\text{length}(Y)$ – размер вектора Y , а $p < 1$. По умолчанию $p = 0.1$ ¹.

Для анимации движения в пространстве используется функция `comet3`.

3. Содержание работы

1. Построение графиков функций
2. Построение нескольких графиков в одних координатных осях
3. Построение поверхностей функции
4. Создание анимированного графика
5. Экспорт и импорт данных для построения графиков

4. Контрольные вопросы

- Команды для построения и оформления одномерных графиков.
- Команды для построения и оформления графиков поверхностей.
- Опишите способы создания анимации в плоскости.

5. Содержание отчета

Отчёт должен содержать: 1) описание исходных данных варианта; 2) последовательность действий для решения задачи; 3) результаты решения задачи; 4) выводы по работе, 5) приложение – листинги созданных программ с комментариями.

Лабораторная работа 12. Моделирование в Matlab

1. Цель работы

Изучение способов решения дифференциальных уравнений в Matlab, развитие навыков использования математических па-кетов для построения и расчёта аналитических моделей физики в дифференциальных уравнениях.

2. Указания к выполнению работы

Анализ поведения многих систем и устройств в динамике, а также решение многих задач в теории колебаний и в поведении упругих оболочек обычно базируется на решении систем обыкновенных дифференциальных уравнений (ОДУ) – ordinary differential equations (ODEs). Как правило их представляют в виде системы из дифференциальных уравнений первого

порядка в форме Коши: $\frac{dy}{dt} = y' = f(y, t)$.

Для решения систем ОДУ в MATLAB реализованы различные решатели: ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb, bvp4c или pdepe – которые реализуют следующие методы решения систем дифференциальных уравнений (приведём только некоторые из них):

- ode45 – одношаговые явные методы Рунге-Кутта 4-го и 5-го порядков в модификации Дорманда и Принца. Это классический метод, рекомендуемый для начальной пробы решения.

- ode23 – одношаговые явные методы Рунге-Кутта 2-го и 4-го порядков в модификации Богацки и Шампина. При умеренной жесткости системы ОДУ и низких требованиях к точности этот метод может дать выигрыш в скорости решения.

- ode113 – многошаговый метод Адамса–Башворта–Мултона переменного порядка класса предиктор–корректор. Это адаптивный метод, который может обеспечить высокую точность решения.

Рассмотрим решение дифференциального уравнения на примере движения тела, брошенного вверх (без учета сопротивления воздуха): $y''(t) = -g$ с начальным условием $y_0 = [0; 10]$.

Для решения задачи составим простой скрипт-файл:

```
y0=[0; 10] %Задание вектора начальных условий
```

```
ts=0:.2:2; % Задание сетки времени
```

```
dydt=@(t,y)[y(2);-9.8]; % Задание анонимной функции правых частей
```

ODE

```
[to,yo]=ode45(dydt,ts,y0) % Расчет решателем ode45
```

После этого остаётся запустить скрипт по имени из командной строки и оценить результат решения.

3. Содержание работы

1. Решение дифференциальных уравнений
2. Построение модели физической задачи

4. Порядок проведения работы

- Задание 1. Решите дифференциальное уравнение согласно варианту.
Задание 2. Составьте модель физической задачи согласно варианту.
Реализуйте её в Matlab как М-функцию. Оцените полученные результаты.
Задание 3. Сделайте выводы о проделанной работе. Оформите отчёт.

5. Контрольные вопросы

- Понятие дифференциального уравнения.
- Что является решением дифференциального уравнения?
- Как решить дифференциальное уравнение в Matlab?
- Составьте аналитическую модель для физической задачи.

6. Варианты заданий

Варианты к заданию 1 представлены в табл 14.1.

Таблица 14.1 – Варианты задания к заданию 1

Номер варианта	Уравнение $f(x,y)$	Начальные условия	Интервал для поиска решения	Шаг изменения
1	$\frac{y}{\cos(x) \cdot \ln(y)}$	$y(1)=1$	[1,10]	1
2	$\text{tg}(x)\text{tg}(y)$	$y(0)=0$	[0,5]	0.5
3	$\frac{y}{1+x^2}$	$y(1)=1$	[1,7]	
4	$\sin(3x)-y \cdot \text{tg}(3x)$	$y(0)=1/3$	[0,4]	0,25
5	$\cos(x-2y)-\cos(x+2y)$	$y(0)=\pi/4$	[0,4 π]	$\pi/2$
6	$\sqrt{x^2 + y \cos(x) }$	$y(0)=3,6$	[4,1;6,7]	0,1
7	$\sin(x)+\cos(y^2)$	$y(0)=2,2$	[0,8;3,2]	0,1
8	$0,7y+x \cdot \ln(x+y)$	$y(0)=2,5$	[12,4;14,1]	0,08
9	$\sqrt{x^2 + e^{xy}}$	$y(0)=1,6$	[5,2;6,8]	0,1
10	$y/\ln(y)$	$y(2)=1$	[2;5]	0,25
11	$-\frac{\sqrt{1+\cos(2x)}}{\sqrt{1+\sin(y)}}$	$y(\pi/4)=0$	[$\pi/4$, 3 π]	$\pi/8$
12	$\frac{1}{x^2} - \frac{1}{y}$	$y(1)=0$	[1;4]	0.3

Задание 2 Построение модели физической задачи

- 2.1 Движение тела, брошенного под углом к горизонту (без учета сопротивления воздуха)
- 2.2 Колебания математического маятника (без затуханий)
- 2.3 Моделирование артиллерийской задачи
- 2.4 Движение подводной лодки (подъём)
- 2.5 Упругое столкновение шаров
- 2.6 Движение небесного тела в гравитационном поле
- 2.7 Движение материального тела в поле тяготения
- 2.8 Колебания пружинного маятника
- 2.9 Колебания физического маятника
- 2.10 Движение брошенного тела, с учетом сопротивления воздуха

7. Содержание отчета

Отчёт должен содержать: 1) описание исходных данных варианта; 2) последовательность действий для решения задачи; 3) результаты решения задачи; 4) выводы по работе, 5) приложение – листинги созданных программ с комментариями.

Лабораторная работа 13. Построение пользовательского интерфейса в Matlab

1. Цель работы

Формирования навыков создания графического интерфейса пользователя (GUI).

2. Указания к выполнению работы

В *Matlab* отмечается 2 подхода к созданию пользовательского графического интерфейса:

1. Все элементы создает пользователь и процедуры

Преимущества:

- полностью контролируемый процесс;
- экономия времени на этапе отладки и модификации GUI;
- в процессе разработки GUI достигается системное понимание функционала.

Недостатки:

- большие затраты времени на этапе построения геометрии GUI.

2. Элементы GUI и процедуры создаются средствами *guide*

Преимущества:

- легко создавать элементы в редакторе
- легко редактировать элементы, структуру всего окна (выравнивать элементы и т.п.)

- система генерирует *guiname.m* исполняемый *guiname.fig*

Недостатки:

- требуется более высокий уровень понимания
- сложность возникает на этапе модификации GUI

Элементы GUI

Figure – родительский объект всех элементов GUI, поле для интерфейса. Построение GUI начинается с команды: `f = figure`. Размер `Figure(position)` привязан к размеру монитора, `[x0,y0,hx,hy] = get(0, 'screensize')`.

Открытие окна инструмента GUIDE

Для открытия окна инструмента GUIDE надо использовать команду

```
>> guide
```

Выбрав мышью на вкладке Create New GUI заготовку Blank GUI и нажав клавишу ОК, можно наблюдать инициализацию инструмента GUIDE. По окончании инициализации появляется основное окно среды GUIDE, содержащее поле окна приложения, вертикальную панель инструментов для добавления элементов интерфейса, горизонтальную панель инструментов и обычное меню (рис. 13.1).

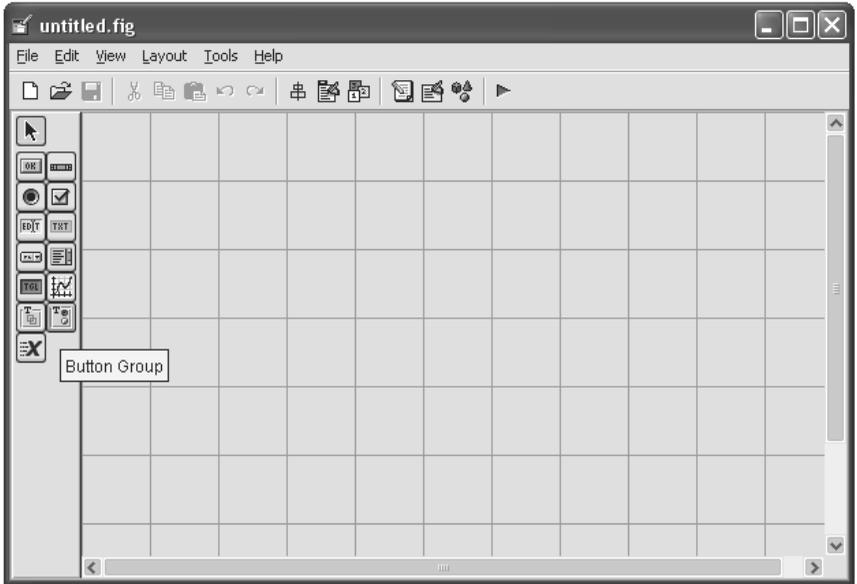


Рис. 13.1 – Пустое окно приложения с GUI

Свойства объектов GUI

Каждый объект (компонент) GUI имеет ряд свойств. Их полный набор задается таблицей из 41 свойства. Создание нового окна с GUI сводится к переносу объектов из панели объектов рис. 13.1 в окно GUI будущего окна с приложением пользователя. Для этого достаточно выбрать нужные объекты и с помощью мыши перенести их в поле окна с приложением пользователя.

После добавления элемента интерфейса необходимо задать его имя (тег), который будет идентифицировать данный объект среди всех остальных объектов. Для задания тега следует перейти к инспектору свойств. Проще всего это сделать двойным щелчком мыши по добавленной кнопке.

3. Содержание работы

1. Создание графического интерфейса для ранее созданной программы
2. Тестирование созданного графического интерфейса

4. Варианты задания

- Построение графика выбранной из списка функции
- Генерация матрицы определённого вида
- Решение СЛАУ выбранным из списка методом

- Интерполяция значения функции в заданной точке выбранным методом
- Аппроксимация табличной функции выбранной приближающей функцией

5. Контрольные вопросы

- Способы создания GUI в Matlab
- Процедура создания GUI через guide
- Общие свойства графических элементов
- Виды графических элементов
- Задание свойств графических элементов

6. Содержание отчета

Отчёт должен содержать: 1) описание исходных данных варианта; 2) последовательность действий для решения задачи; 3) результаты решения задачи; 4) выводы по работе, 5) приложение – листинги созданных программ с комментариями.

Лабораторная работа 14. Построение блок-схем в MS Visio

1. Цель работы

Знакомство с редактором векторной графики Microsoft Visio, приобретение навыков работы в векторных графических редакторах.

2. Указания к выполнению работы

Основные понятия и термины

Образец (трафарет). Объект в наборе фигур Visio. Для создания схемы с помощью Visio используют образцы, которые поставляются вместе с программой или были загружены из Интернета.

Набор элементов. Коллекция образцов.

Фигура. Объект на странице документа Visio. Фигуры создаются путем перетаскивания образцов и наборов элементов на страницу документа; однако фигуры могут также создаваться и другими способами.

Шаблон. Документ Visio, который включает одну или несколько страниц с предустановленными размерами и единицами измерения. Шаблон может также включать один или несколько наборов элементов, фоновые страницы и макеты; страницы в нем могут содержать фигуры или текст. Шаблон может включать и специальное программное обеспечение, которое работает только в нем.

Рабочая область. Коллекция окон программы Visio и их настроек.

В Visio представлено множество шаблонов схем и тысячи фигур, одни из которых простые, а другие – довольно сложные. Каждый шаблон предназначен для конкретных целей, самый простой способ получить сведения о шаблонах и их назначении – воспользоваться окном Категории шаблонов.

Для создания нового документа Visio выполните указанные действия.

1. На вкладке Файл выберите команду «Создать».

2. В появившемся разделе выберите нужный шаблон:

3. Далее можно создать новую схему, используя инструменты на панели инструментов (сверху) и из набора элементов (слева). Для соединения

фигур на листе используется соединительная линия , которая автоматически прикрепляется к точкам соединения  (эти точки есть по умолчанию на всех фигурах, но их можно также добавить дополнительно).

Сохранение файлов Visio

По умолчанию Visio сохраняет файлы в формате *.vsd. В зависимости от того, как планируется использовать схему и организовать совместную работу с ней, могут использоваться разные форматы:

- обычный файл изображения, в том числе в форматах JPG, PNG и BMP;

- веб-страница в формате HTML (файлы изображений и другие файлы ресурсов сохраняются во вложенной папке в том же расположении, в котором сохранен HTML-файл);
- PDF- или XPS-файл;
- чертёж AutoCAD в формате DWG или DXF

3. Содержание работы

1. Построение блок-схемы поиска наибольшего элемента в матрице
2. Построение блок-схемы алгоритма расчёта n -го элемента последовательности Фибоначчи
3. Построение блок-схемы численного алгоритма (по вариантам)

4. Контрольные вопросы

- Основные понятия MS Visio
- Векторные и растровые графические редакторы
- Форматы файлов, поддерживаемые MS Visio
- Элементы для рисования фигур
- Условные графические обозначения элементов блок-схем
- Чтение блок-схемы (на примере выполненной работы)

5. Порядок проведения работы

Задание 1. Создайте документ MS Visio по шаблону «Простая блок-схема» (Basic Flowchart Shapes). Составьте блок-схему поиска наибольшего элемента в матрице. Оформите её согласно ГОСТ 19.701-90.

Задание 2. Составьте блок-схему алгоритма расчёта n -го элемента последовательности Фибоначчи. Оформите её согласно ГОСТ 19.701-90.

Задание 3. Составьте блок-схему численного алгоритма согласно варианту. Оформите её согласно ГОСТ 19.701-90.

6. Варианты заданий

Варианты к заданию 3:

1. Метод простых итераций для решения нелинейного уравнения
2. Метод Гаусса для решения систем линейных уравнений
3. Метод касательных для решения нелинейного уравнения
4. Метод Крамера для решения систем линейных уравнений
5. Метод хорд для решения нелинейного уравнения
6. Метод простых итераций для решения СЛАУ
7. Метод Зейделя для решения систем линейных уравнений
8. Метод Ньютона для решения систем нелинейных уравнений
9. Метод золотого сечения для поиска минимума функции
10. Метод половинного деления для решения уравнений

11. Метод покоординатного спуска для поиска минимума функции
12. Интерполирование полиномом Лагранжа
13. Интерполирование полиномом Ньютона
14. Аппроксимация методом наименьших квадратов (линейная приближающая функция)
15. Аппроксимация методом наименьших квадратов (приближающая функция - квадратичный трёхчлен)

7. Содержание отчета

Отчёт должен быть оформлен в MS Visio и содержать на отдельных листах: 1) титульный лист; 2) построенные схемы и диаграммы; 3) выводы по работе.

Лабораторная работа 15. Построение схем и диаграмм в MS Visio

1. Цель работы

Формирование навыков создания графических документов в редакторах векторной графики.

2. Указания к выполнению работы

Для выполнения каждого задания используйте соответствующий шаблон при создании документа. Каждую схему сохраняйте на отдельном листе документа. При необходимости объединяйте фигуры в группы или структуры.

В версии *Visio* 2010 введены три новых способа установить связи и добавить структуру в схемы.

Контейнеры. Контейнер не только обеспечивает визуальную границу вокруг набора объектов, но также устанавливает логическую связь между объектами в нем. А именно: фигуры знают, когда они являются членами контейнера, и контейнеры знают, какие фигуры в них содержатся.

Ключевое преимущество контейнеров заключается в том, что контейнер и его члены можно перемещать, копировать или удалять как единое целое, но при этом каждая фигура в контейнере сохраняет свою независимость. Другими словами, члены контейнера можно выделять с помощью одного щелчка и обращаться к их данным и другим свойствам.

Контейнер может содержать фигуры, другие контейнеры и списки.

Списки. Список – это специальный тип контейнера, в котором поддерживается упорядоченная связь среди его членов. Каждый объект в списке знает свое порядковое положение; новые объекты добавляются не просто в список, а в конкретное место в списке.

Список может содержать фигуры и контейнеры, но не другие списки.

Выноски. В предыдущих версиях *Visio* выноска была просто фигурой, которая приклеивалась к другой фигуре в качестве комментария. В *Visio* 2010 выноски по-прежнему предоставляют способ добавления комментариев к фигурам, но здесь выноска знает фигуру, к которой она привязана, а фигура может идентифицировать привязанные к ней выноски.

Для визуального объединения набора фигур могут использоваться либо группы, либо контейнеры.

Однако имеются два ключевых отличия в их свойствах, которые позволяют сделать выбор в пользу групп или контейнеров в зависимости от ваших требований.

3. Содержание работы

1. Построение схемы данных и схемы процессов.

2. Построение схемы топологий компьютерных сетей.

3. Построение плана помещения.

4. Контрольные вопросы

– Какие средства контроля за размерами элементов предусмотрены Microsoft Visio?

– Как добавить текст в диаграмме Microsoft Visio?

– Как осуществляется экспорт фрагментов диаграмм Microsoft Visio в текстовый редактор?

– Какие возможности предоставляет программа MS Visio?

– Каким способом можно добавить текст в фигуру, а также независимый текст на диаграмму?

– С помощью какой команды осуществляется форматирование фигур?

– Каким образом можно применить к диаграмме цветовую схему?

– Как добавить текст к коннектору?

5. Порядок проведения работы

Задание 1. Постройте схему некоторого процесса согласно варианту.

Задание 2. Постройте схемы топологий компьютерных сетей: шина, кольцо, звезда, дерево, полносвязная, гибридная.

Задание 3. Постройте план помещения согласно варианту.

6. Содержание отчета

Отчёт должен быть оформлен в MS Visio и содержать на отдельных листах: 1) титульный лист; 2) построенные схемы и диаграммы; 3) выводы по работе.

Лабораторная работа 16. Набор и вёрстка текста в TEX

1. Цель работы

Знакомство с системами вёрстки и набора текстов семейства TEX.

2. Указания к выполнению работы

TEX – система для набора и вёрстки текстов с формулами. С её помощью можно подготовить документы различных типов. TEX представляет собой специализированный язык программирования для создания издательских систем. Каждая издательская система на базе TEX представляет собой пакет макроопределений (макропакет) этого языка.

Основные понятия

Исходный файл – собственно текст документа вместе со спецсимволами и командами, с помощью которых системе передаются указания касательно размещения текста. Такой файл может быть создан любым текстовым редактором и не должен содержать шрифтовых выделений и других элементов оформления.

Спецсимволы { } \$ & # % _ ^ ~ \ имеют особый статус и выполняют специальные функции, поэтому употребление их в тексте в прямом виде приведёт к ошибке компиляции документа. Печатное изображение знаков, соответствующих первым семи из них, можно получить, если в исходном тексте поставить перед соответствующим символом без пробела знак косой черты \. Если символ % употреблен в тексте не в составе комбинации \%, то он является «символом комментария». Знак \$ ограничивает математические формулы. Знаки { } ограничивают группы в исходном файле. Знак ~ обозначает «неразрывный пробел» между словами. Со знака \ начинаются все команды в TEX.

Структура исходного текста имеет следующий вид

```
\documentclass{article}
% Препамбула
\begin{document}
% Основной текст документа
\end{document}
```

Автор и заглавие документа задаются командами \author {имя автора} и \title {заглавие} в препамбуле документа. Для того, чтобы они были созданы в документе, в теле необходимо вызвать команду /maketitle.

Команды \include и \input позволяют автоматически склеивать содержимое файлов, указанных в качестве аргумента.

3. Содержание работы

1. Оформление титульного листа работы

2. Создание текста
3. Создание и компилирование нескольких документов
4. Оформление текста

4. Контрольные вопросы

- Какое расширение имеет исходный TEX-файл?
- Как выглядит заголовок исходного файла?
- Для чего предназначены файлы *.sty?
- Какие символы являются спецсимволами в TEX?
- Как ограничивается группа в TEX?

5. Порядок проведения работы

Задание 1. Создайте документ TEX минимального размера. В тело текста добавьте несколько абзацев текста, транслируйте документ и оцените результат.

Задание 2. Составьте титульный лист к работе определённого типа (согласно варианту), удовлетворяющий требованиям ОС ТУСУР

Задание 3. Создайте в отдельном каталоге копию со всеми ранее созданными файлами. Создайте несколько новых документов (разделов лабораторной работы) и склейте их в едином документе соответствующего типа при помощи соответствующих команд.

6. Варианты заданий

Варианты к заданиям 2, 3:

1. Отчёт о лабораторной работе
2. Отчёт о практике
3. Реферат
4. Курсовая работа
5. Выпускная квалификационная работа
6. Магистерская диссертация

7. Содержание отчета

Отчётом о выполнении данной работы является файл (файлы) формата .tex с исходным текстом, снабженным комментариями.

Лабораторная работа 17. Набор формул и стили в TEX

1. Цель работы

Изучение процедуры набора формул в TEX, развитие умений оформления документации.

2. Указания к выполнению работы

В документе, подготовленном с помощью TEXа, различают математические формулы внутри текста и «выключные» (выделенные в отдельную строку). Формулы внутри текста окружаются знаками \$ (с обеих сторон). Выключные формулы окружаются парами знаков доллара \$\$ и \$\$ с обеих сторон. Формулами считаются как целые формулы, так и отдельные буквы, в том числе греческие, а также верхние и нижние индексы и спецзнаки. Пробелы внутри исходного текста, задающего формулу, игнорируются (надо по-прежнему ставить пробелы, обозначающие конец команды), пустые строки не разрешаются. TEX расставляет пробелы в математических формулах автоматически (например, знак равенства окружается небольшими пробелами). Если надо оставить пробел перед или после внутритекстовой формулы, надо оставить его перед или после ограничивающего ее знака доллара. То же самое относится и к знакам препинания, следующим за внутритекстовой формулой: их также надо ставить после закрывающего формулу знака доллара. Каждая буква в формуле рассматривается как имя переменной и набирается шрифтом «математический курсив» (в отличие от обычного курсива, в нем увеличены расстояния между соседними буквами).

Для набора формул используйте таблицы символов TEX.

Стиль документа задается командой `\documentstyle`. Обязательный аргумент этой команды представляет собой имя так называемого «стилевого файла» с расширением `.sty`: стилом `article` соответствует файл `article.sty`, стилем `report` — файл `report.sty`, и т.д. Стилиевой файл содержит TEXовские команды, которые присваивают значения различным параметрам и задают различные макроопределения (например, в файле `book.sty` содержится определение команды `\chaptername`, ответственной за то, что главы называются Chapter). Действие команды `\documentstyle` начинается с того, что TEX считывает содержимое этого файла; при этом происходят все вышеупомянутые присваивания и определения макросов. После того, как основной стилиевой файл прочитан, начинается просмотр списка стилевых опций. Если в этом списке присутствует стилевая опция, название которой отличается от стандартных, то TEX читает «дополнительный стилиевой файл» с именем, совпадающим с именем опции.

3. *Содержание работы*

- 1 Создание таблицы знаков
- 2 Набор математических формул

4. *Порядок проведения работы*

Задание 1. Создайте таблицы математических символов, греческих букв и операторов согласно варианту.

Задание 2. Наберите несколько математических выражений (не меньше 10), используя введённые буквы. Выражения обязательно должны содержать: стрелки, степени и индексы, дроби, корни, штрихи (хотя бы 1 раз).

5. *Варианты заданий*

Варианты к заданию 1.

- 1: $\leq, \pm, \leftrightarrow, \exists, \in, \partial, \alpha, \beta, \chi, \Omega, \infty$
- 2: $\geq, \times, \rightarrow, \forall, \notin, \Delta, \delta, \varepsilon, \phi, \Xi, \pi$
- 3: $\approx, \div, \leftarrow, \wedge, \subseteq, \Sigma, \varphi, \gamma, \eta, \Psi, \neq$
- 4: $\sim, \oplus, \Leftrightarrow, \neg, \cup, \angle, \kappa, \lambda, \mu, \Lambda, \Pi$

6. *Контрольные вопросы*

1. Какой последовательностью символов задаётся формула $x_{11}=x_1^{y-1}$?
2. Какой последовательностью символов задаётся формула $f'(x)$?
3. Какой последовательностью символов задаётся формула $n \rightarrow \infty$?
4. Что будет выведено на экран в результате вызова кода:
$$\llbracket (x^2)' = 2x, \quad y^{(n)} = ny^{(n-1)}, \quad 3! = 1 \cdot 2 \cdot 3. \rrbracket$$
5. В какие парные знаки заключается нумерованная выключная формула?

7. *Содержание отчета*

Отчётом о выполнении данной работы является файл (файлы) формата .tex с исходным текстом, снабженным комментариями.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Корницкая М.Н. Лабораторный практикум по курсу "Информатика": методические указания к выполнению лабораторных работ для студентов специальности "Мировая экономика" / М.Н. Корницкая, Г.М. Бусыгина, В.В. Соколова; Алт.гос.техн.ун-т им. И.И. Ползунова. – Барнаул: Изд-во АлтГТУ, 2006. – 68с.
2. Сулейманов Р. Р. Численные методы в системе Mathcad. – Башкирский институт развития образования, 2007. – 42 с.
3. Гурский Д.А., Турбина Е.С. Вычисления в Mathcad 12. – СПб.: Питер, 2006. – 544с.
4. Ноздреватых Д.О. Начальные сведения о MathCAD: Учебное пособие. – Томск: ТУСУР, 2016. – 215 с.
5. Основы вычислений и программирования в пакете MathCAD : учеб. пособие / Ю. Е. Воскобойников [и др.] ; под ред. Ю. Е. Воскобойникова ; Новосиб. гос. архитектур.-строит. ун-т (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2012. – 212 с.
6. Ноздреватых, Д. О. Начальные сведения о MATLAB: Учебное пособие [Электронный ресурс] / Ноздреватых Д. О. — Томск: ТУСУР, 2016. — 176 с.
7. Алейников И.А. Практическое использование пакета Mathcad при решении задач: Учебное пособие. – М.: Российский государственный открытый технический университет путей сообщения Министерства путей сообщения Российской Федерации, 2002. – 114 с.
8. Дьяконов В.П. MATLAB. Полный самоучитель. – М.: ДМК Пресс, 2012. – 768 с.
9. Максимова А.П., Малова Н.А. Лабораторный практикум по вычислительной математике. Методические указания по выполнению лабораторных работ. – Чебоксары: Волжский филиал МАДИ (ГТУ), 2008. – 91 с.
10. Берман Н.Д. MS Visio 2010: основы работы. – Хабаровск: Издательство ТОГУ, 2014. – 99 с.
11. Ширяева Е.В., Ширяева И.В. Введение в TeX. Часть I. Набор и вёрстка текста. – Ростов-на-Дону: Южный федеральный университет, 2013. – 64 с.
12. Ширяева Е.В., Ширяева И.В. Введение в TeX. Часть II. Набор формул. – Ростов-на-Дону: Южный федеральный университет, 2013. – 39 с.