

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Российский химико-технологический университет  
имени Д. И. Менделеева

**А. В. Горанский, А. Ф. Егоров, А. Ю. Захарова**

**ПРОГРАММНЫЕ СРЕДСТВА РЕШЕНИЯ  
МАТЕМАТИЧЕСКИХ ЗАДАЧ**

**Лабораторный практикум**

Утверждено Редакционным  
советом университета  
в качестве учебного пособия

Москва  
2017

УДК 66.012-52.001.2:004.42(75)

ББК 22.18:22.19

Г67

Рецензенты:

Доктор технических наук, профессор Тверского государственного  
технического университета

*В. Н. Богатилов*

Кандидат технических наук, доцент Российского химико-технологического  
университета им. Д. И. Менделеева

*А. В. Женса*

**Горанский А. В.**

Г67 Программные средства решения математических задач. Лабораторный  
практикум: учеб. пособие / А. В. Горанский, А. Ф. Егоров, А. Ю. Захарова. –  
М.: РХТУ им. Д. И. Менделеева, 2017. – 72 с.

ISBN 978-5-7237-1580-6

Содержатся методические рекомендации по использованию универсальных  
программных средств для решения математических задач. Дано описание  
лабораторных работ для решения систем линейных и нелинейных уравнений, поиска  
экстремумов функций и решения других математических задач с использованием  
программных средств MS Excel, MathCAD, MATLAB и языка программирования R.

Предназначено для студентов, изучающих дисциплину «Универсальные  
программные средства решения математических задач», относящуюся к базовой части  
программы бакалавриата по направлению подготовки 18.03.02 Энерго- и  
ресурсосберегающие процессы в химической технологии, нефтехимии и  
биотехнологии в соответствии с федеральными государственными образовательными  
стандартами высшего образования, и выполняющих лабораторные, учебные курсовые  
научно-исследовательские работы и выпускные квалификационные работы.

УДК 66.012-52.001.2:004.42(75)

ББК 22.18:22.19

Учебное издание

ГОРАНСКИЙ Артем Витальевич

ЕГОРОВ Александр Федорович

ЗАХАРОВА Анастасия Юрьевна

**ПРОГРАММНЫЕ СРЕДСТВА РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ**

**Лабораторный практикум**

Редактор Е. В. Копасова

Подписано в печать 25.12.2017 г. Форма 60x84 1/16

Усл. печ. л. 4,2. Уч.-изд. л. 4,4. Тираж 100 экз.

Российский химико-технологический университет имени Д. И. Менделеева

Издательский центр

Адрес университета и издательского центра: 125047, Москва, Миусская пл., 9

ISBN 978-5-7237-1580-6 © Российский химико-технологический  
университет им. Д. И. Менделеева, 2017

© Горанский А. В., Егоров А. Ф., Захарова А. Ю.,  
2017

## Оглавление

Введение.....	5
1. Лабораторная работа 1. Использование табличного процессора MS Excel и редактора MS Word для обработки данных и оформления результатов расчетов.....	6
1.1. Функциональные возможности MS Excel .....	6
1.2. Решение системы уравнений графическим способом.....	8
1.3. Решение системы уравнений с помощью надстройки «Поиск решения» .....	10
1.4. Генерация случайных чисел и их сортировка .....	12
1.5. Требования к отчету по лабораторной работе .....	14
2. Лабораторная работа 2. Использование систем компьютерной алгебры на примере MathCAD (часть 1).....	15
2.1. Основы работы с системами компьютерной математики на примере Mathcad .....	15
2.2. Расчет значений функции и представление результатов в виде вектор-строки и вектор-столбца .....	21
2.3. Построение двумерных графиков функции в декартовых координатах .....	22
2.4. Задание функции двух переменных и матриц.....	24
2.5. Нахождение частных производных и их значений.....	24
2.6. Построение трехмерного графика функции двух переменных.....	26
2.7. Требования к отчету по лабораторной работе .....	27
3. Лабораторная работа 3. Использование систем компьютерной алгебры на примере MathCAD (часть 2).....	27
3.1. Определение порядка химической реакции с использованием MathCAD .....	27
3.2. Решение системы линейных уравнений с помощью встроенной функции и матричного метода.....	30
3.3. Определение детерминанта матрицы.....	32
3.4. Нахождение экстремума функции.....	33
3.5. Требования к отчету по лабораторной работе .....	35
4. Лабораторная работа 4. Использование пакета прикладных программ MatLAB для решения математических и общепромышленных задач.....	35
4.1. Пакет прикладных программ MatLAB.....	35

4.2. Работа с переменными и массивами .....	37
4.3. Работа с функцией в *.m-файлах .....	39
4.4. Работа с данными из внешних файлов.....	41
4.5. Решение системы линейных алгебраических уравнений .....	44
4.6. Реализация метода наименьших квадратов в MATLAB .....	46
4.7. Определение кислотности раствора в MATLAB .....	49
4.8. Требования к отчету по лабораторной работе .....	51
5. Лабораторная работа 5. Язык программирования R-Language для решения математических задач .....	51
5.1. Язык программирования R.....	51
5.2. Работа с матрицами в визуальной среде для языка R. Сортировка массива .....	53
5.3. Решение системы уравнений методом Гаусса .....	54
5.4. Нахождение экстремумов функции .....	55
5.5. Построение графика функции.....	56
5.6. Решение системы нелинейных уравнений .....	57
5.7. Вычисление определенного интеграла .....	58
5.8. Работа с условными операторами .....	59
5.9. Работа с индексированными матрицами .....	60
5.10. Экспорт данных из визуальной среды для языка R.....	61
5.11. Решение системы нелинейных уравнений с использованием языка R.....	61
6. Обзор специализированных программных комплексов и языков программирования, используемых в инженерной практике .....	63
6.1. Система компьютерной алгебры Mathematica .....	63
6.2. Язык программирования Python .....	66
6.3. Язык программирования Julia.....	67
6.4. Язык программирования Fortran.....	67
6.5. Пакет прикладных программ STATISTICA .....	68
Заключение .....	71
Библиографический список.....	72

## Введение

В данном практикуме рассматриваются лабораторные работы по курсу «Универсальные программные средства решения математических задач», предназначенные для закрепления теоретических и практических знаний в области применения пакетов прикладных программ для математических вычислений.

Учебное пособие содержит пять лабораторных работ, направленных на освоение студентами основных функциональных возможностей:

- табличных редакторов на примере MS Excel;
- компьютерной математики на примере PTC Mathcad;
- пакета прикладных программ MatLab и его модулей;
- языка программирования R.

В лабораторном практикуме, помимо практических заданий для лабораторных работ, представлены теоретические и практические аспекты компьютерного моделирования [1] и применения современных специализированных компьютерных пакетов программ и языков программирования (Mathematica, Python, Julia, Fortran, STATISTICA) для автоматизированного численного и символьного вычисления, визуализации вычислений, математического моделирования, статистической обработки данных и разработки алгоритмов. Представлены функциональные возможности пакетов прикладных программ для математических вычислений и математического моделирования.

# **1. Лабораторная работа 1. Использование табличного процессора MS Excel и редактора MS Word для обработки данных и оформления результатов расчетов**

**Цель работы:** Получение практических навыков работы с табличным процессором MS Excel. Реализация простых расчетов, диаграмм и графиков.

## **1.1. Функциональные возможности MS Excel**

В зависимости от версии MS Excel (2007, 2010, 2013, 2016) может существенно различаться по функциональным возможностям, но обобщенно этот табличный процессор позволяет формировать и выводить на печать документы, представленные в табличном виде, выполнять расчеты на основании исходных данных. Более подробно основные задачи, решаемые с помощью табличного редактора Excel, перечислены ниже [2]:

- Создание, редактирование и оформление разнообразных табличных документов;
- Выполнение расчетов на основании исходных данных с применением специально предназначенного инструментария;
- Сохранение созданного документа в различных форматах (Excel, веб-страница, шаблон, текстовый документ и др.);
- Ввод и обработка табличных данных с помощью встроенных механизмов формул, функций, макросов и др.;
- Анализ и управление данными (автоматический расчет итоговых и промежуточных данных, их структуризация и консолидация, использование сводных таблиц и отчетов и др.);
- Вставка в документ различных объектов (рисунок, функция, примечание, объект и др.);
- Выполнение операций по ручному и автоматическому форматированию документа и его составных элементов. Автоматическая проверка орфографии документа, оперативное устранение допущенных некорректностей с использованием механизма автозамены;

- Импорт необходимых данных из различных источников (включая базы данных OLAP) и последующая их обработка. Поддержка XML-формата;
- Настройка параметров представления табличного документа, а также настройка пользовательского интерфейса;
- Доступ к данным других программ (например, Word, PowerPoint, Access и др.);
- Формирование самых разнообразных отчетов – аналитических, сводных, графических, в виде диаграмм и др.;
- Выполнение стандартных функций Microsoft Office – печать документа, поиск данных и их замена, проверка наличия ошибок и др.;
- Создание приложений с применением языка программирования VBA;
- Создание рисунков с использованием специально предназначенной функции;
- Защита табличного документа от неквалифицированного и несанкционированного доступа;

Специфика использования табличного редактора Excel во многом определяется потребностями конкретного пользователя, поэтому перечень задач может меняться в зависимости от конкретной ситуации.

Документ MS Excel называют рабочей книгой (рис. 1.1). Табличное пространство состоит из строк (1–65536) и столбцов (до 256), каждый рабочий лист имеет название, которое отображается внизу. В ячейки можно вводить данные различных типов, включая:

- текст,
- число – в различных форматах,
- дата/время,
- формула – может содержать константы, ссылки на ячейки, встроенные функции Excel, соединенные операторами ( +, -, \*, /, ^ ).

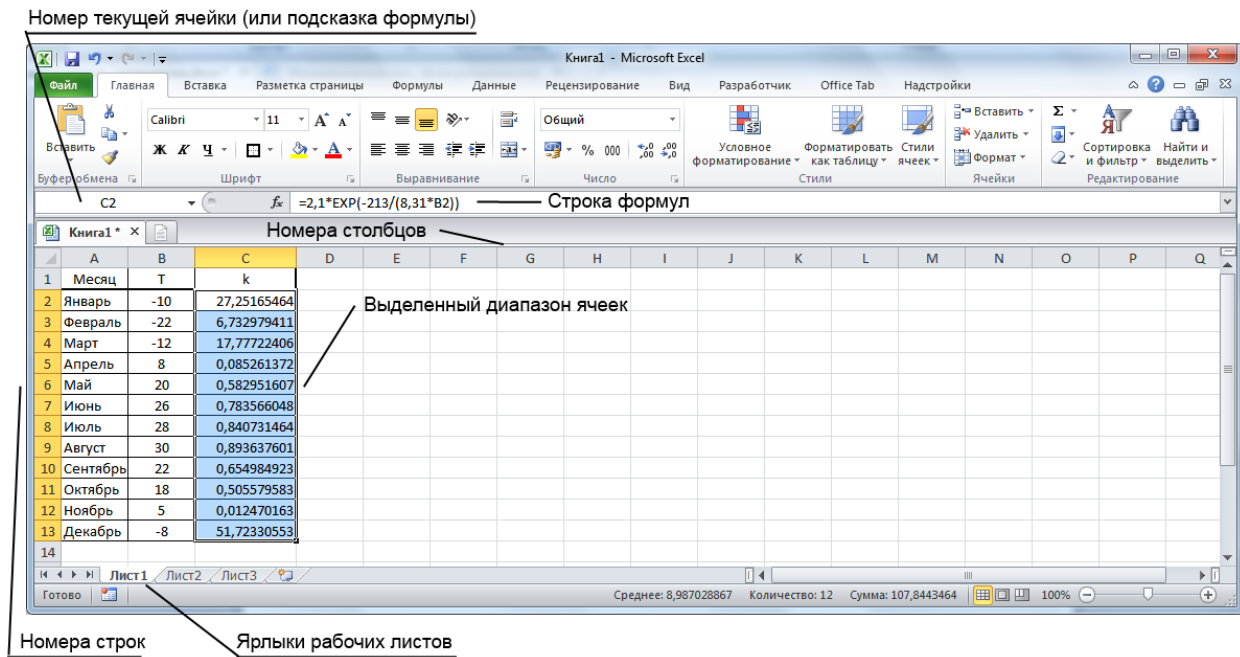


Рис. 1.1. Интерфейс программы Microsoft Excel

## 1.2. Решение системы уравнений графическим способом

**Задание:** Решить графическим способом в MS Excel систему уравнений:

$$\begin{cases} y - x^2 - 5x + 6 = 0 \\ y = 2 \cdot |x - 2| \end{cases}$$

**Решение:**

1. Необходимо преобразовать уравнения системы к виду  $y = f(x)$ .

$$\begin{cases} y = x^2 + 5x - 6 \\ y = 2 \cdot |x - 2| \end{cases}$$

2. Задать начальные значения  $x$ .

3. Найти значение первой функции при заданных  $x$ .

4. Найти значение второй функции при тех же  $x$  (рис. 1.2).

5. Далее необходимо выделить блок с данными и построить графики функций, используя точечный тип диаграммы.

6. Решением системы являются точки пересечения графиков функций.



	A	B	C
1	<b>x</b>	<b>y<sub>1</sub></b>	<b>y<sub>2</sub></b>
2		=A2*A2+5*A2-6	=2*ABS(A2-2)
3	-15	144	34
4	-13	98	30
5	-11	60	26
6	-9	30	22
7	-7	8	18
8	-5	-6	14
9	-3	-12	10
10	-1	-10	6
11	1	0	2
12	3	18	2
13	5	44	6
14	7	78	10
15	9	120	14
16	11	170	18
17	13	228	22

Рис. 1.2. Определение значений функций при заданных  $x$

7. Для нахождения координат точек пересечения с заданной точностью построить новый график на том отрезке, где находится решение, с шагом, равным значению точности (рис. 1.3).

### Графическое решение системы

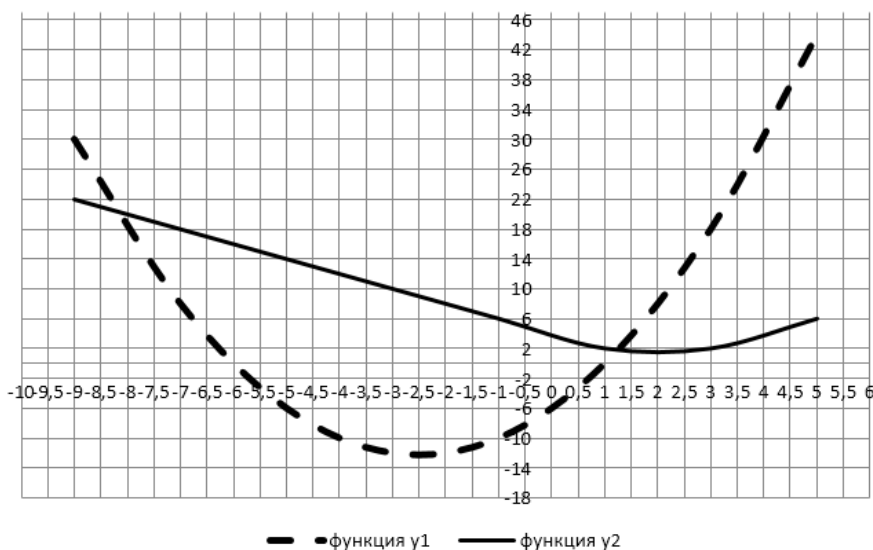


Рис. 1.3. Графическое решение системы

Ответами являются точки пересечения двух графиков:  $(-8, 2)$ ;  $(1,25, 2)$ .

### 1.3. Решение системы уравнений с помощью надстройки «Поиск решения»

*Задание:* Решить систему уравнения, воспользовавшись командой «Поиск решения»:

$$\begin{cases} x - y = 2 \\ 3x - y^2 = 3 \end{cases}$$

«Поиск решения» – это надстройка для Microsoft Excel, которую можно использовать для анализа «что если». С ее помощью можно найти оптимальное значение (максимум или минимум) формулы, содержащейся в одной ячейке, называемой целевой, с учетом ограничений на значения в других ячейках с формулами на листе. Надстройка «Поиск решения» работает с группой ячеек, называемых ячейками переменных решения или просто ячейками переменных, которые используются при расчете формул в целевых ячейках и ячейках ограничения. Надстройка «Поиск решения» изменяет значения в ячейках переменных решения согласно пределам ячеек ограничения и выводит нужный результат в целевой ячейке.

*Решение:*

1. Необходимо записать исходные данные (рис. 1.4) (в ячейку A1 ввести текст «Заданная система», в ячейку B1 записать первое уравнение, в ячейку B2 – второе уравнение, в ячейку A3 ввести текст «X=», в ячейку A4 «Y=», в ячейку A5 «уравнение 1», в ячейку A6 «уравнение 2». Ячейки B3 и B4 оставляем пустыми для рассчитанных значений X и Y соответственно).

2. В ячейку B5 нужно вписать уравнение 1, используя правило записи арифметических выражений, следующим образом: в левой части вместо X указывать ячейку B3, вместо Y ячейку B4, правую часть отбросить. Таким же образом переписать левую часть второго уравнения в ячейку B6.

4. Далее нужно выбрать команду «Данные» → «Поиск решения».

5. Установить целевую ячейку – ту ячейку, в которой содержится формула, например, B5 и задать значение, равное значению правой части первого уравнения.

6. В поле «изменяя ячейки переменных» указать ячейки, в которых хотим увидеть ответ (B3 и B4).

	A	B
1	Заданная система	$x-y=2$
2		$3x-y^2=3$
3	$x=$	
4	$y=$	
5	Уравнение 1	$=B3-B4$
6	Уравнение 2	$=3*B3-B4*B4$

Рис. 1.4. Задание исходных данных

7. Ввести ограничение  $B6 = 3$ . Для этого кликнуть на кнопку «Добавить» и в полученном окне установить реквизиты следующим образом: в поле «Ссылка на ячейку» указать ячейку, в которой записана левая часть другого уравнения, в центральном поле выбрать знак «=», в третьем – ввести число, равное значению правой части. Закрывать окно «Добавить ограничение», щелкнув кнопкой «Ок» (рис. 1.5).

Рис. 1.5. Параметры поиска решения

8. Решить систему уравнений, кликнув на кнопку «Найти решение». Ответом являются значения  $x=1,21$ ,  $y=-0,79$  (рис. 1.6).

	А	В
1	Заданная система	$x-y=2$
2		$3x-y^2=3$
3	$x=$	1,208711978
4	$y=$	-0,791288022
5	Уравнение 1	2
6	Уравнение 2	2,999999201

Рис. 1.6. Решение системы уравнений командой «Поиск решения»

#### 1.4. Генерация случайных чисел и их сортировка

*Задание:* Сгенерировать выборку 20 случайных целых чисел (диапазон от -10 до 10) и отсортировать по убыванию в MS Excel.

**Использование функции СЛЧИС.** Представленная в Excel функция СЛЧИС генерирует равномерное распределенное случайное вещественное число в промежутке между 0 и 1. Другими словами, любое число от 0 до 1 имеет равную вероятность быть возвращенным этой функцией. Если вам нужны случайные числа с большими значениями, используйте простую формулу умножения. Следующая формула, например, генерирует равномерное случайное число между 0 и 1000: =СЛЧИС()\*1000.

Чтобы ограничить случайное число целыми числами, используйте функцию ОКРУГЛ: =ОКРУГЛ((СЛЧИС()\*1000);0).

**Использование функции СЛУЧМЕЖДУ.** Для генерации равномерных случайных чисел между любыми двумя числами вы можете использовать функцию СЛУЧМЕЖДУ. Следующая формула, например, генерирует случайное число между -10 и 10: =СЛУЧМЕЖДУ(-10;10).

В версиях, предшествующих Excel 2007, функция СЛУЧМЕЖДУ доступна только при установке дополнительного пакета анализа. Для совместимости с предыдущими версиями (и чтобы избежать использования этой надстройки) используйте такую формулу, где А представляет нижний, а В – верхний предел: = СЛЧИС()\*(В-А)+А. Чтобы сгенерировать

случайное число между 40 и 50, используйте следующую формулу:  
 $=\text{СЛЧИС}()*(50-40)+40$ .

*Решение:*

1. Для генерации выборки случайных целых чисел необходимо в меню выбрать команду «Данные» → «Анализ данных» → «Генерация случайных чисел».

Далее в поле «Число переменных» нужно ввести «1». В поле число случайных чисел – «20». Распределение – «Равномерное». Параметры между «-10» и «10». В параметрах ввода указывается выходной интервал, в котором хранятся случайные числа. В данном случае это первая ячейка для ряда чисел (рис. 1.7).

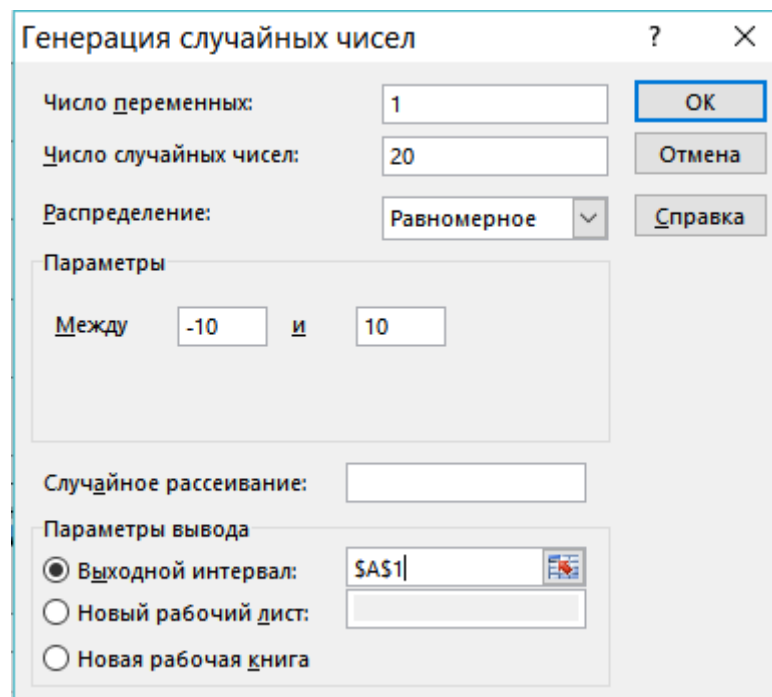


Рис. 1.7. Генерация случайных чисел

2. Для сортировки ряда случайных чисел по убыванию нужно выделить диапазон ячеек и в меню выбрать команду «Данные» → «Сортировка». В появившемся окне указать параметры сортировки (рис. 1.8).

Ряд из 20 случайных чисел, в заданном диапазоне от -10 до 10, отсортированных по убыванию, приведен на рис. 1.9.

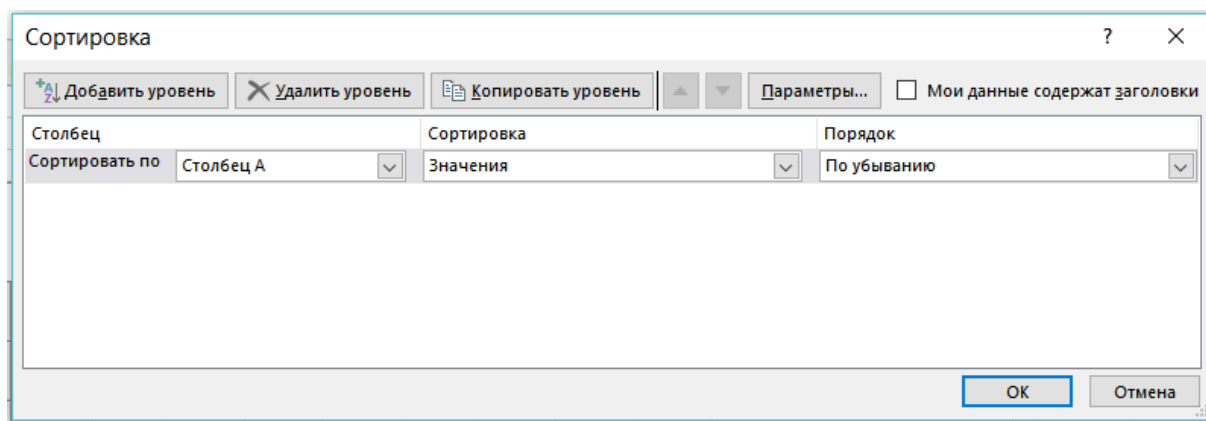


Рис. 1.8. Параметры сортировки чисел

	A
1	-9
2	-9
3	-8
4	-7
5	-6
6	-5
7	-4
8	-2
9	-1
10	-1
11	0
12	0
13	0
14	2
15	5
16	6
17	6
18	6
19	7
20	7

Рис. 1.9. Ряд случайных чисел

### 1.5. Требования к отчету по лабораторной работе

Отчет по лабораторной работе выполняется на листах белой бумаги формата А4 в печатном виде.

Отчет формируется в следующем порядке:

1. Титульный лист.
2. Содержание.

3. Цель работы.
4. Задание.
5. Теоретическая часть. Краткие теоретические положения о функциональных возможностях MS Excel.
6. Практическая (расчетная) часть.
7. Выводы по результатам выполнения работы.
8. Библиографический список.
9. Приложение.

При оформлении отчета используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страниц ставятся по центру в верхнем колонтитуле.

## **2. Лабораторная работа 2. Использование систем компьютерной алгебры на примере MathCAD (часть 1)**

**Цель работы:** Получение навыков решения расчетно оформительских задач в вычислительной среде MathCAD.

### **2.1. Основы работы с системами компьютерной математики на примере Mathcad**

Рабочий документ Mathcad представляет собой совокупность областей. Область может располагаться в любом месте документа.

Для создания новой области следует просто выполнить щелчок мышью на свободном месте документа.

Чтобы сделать видимыми границы областей, следует выбрать в меню **Вид** подменю **Границы**, в результате чего фоновым цветом будут отображены только имеющиеся области (рис. 2.1).

- Визир предназначен для указания позиции в рабочем документе, в которой должна появиться очередная область.
- Маркер ввода в Mathcad аналогичен подобному средству в любом текстовом редакторе.

- Выделяющая рамка используется, во-первых, при копировании, перемещении и удалении частей математического выражения и, во-вторых, для вставки и удаления операторов.

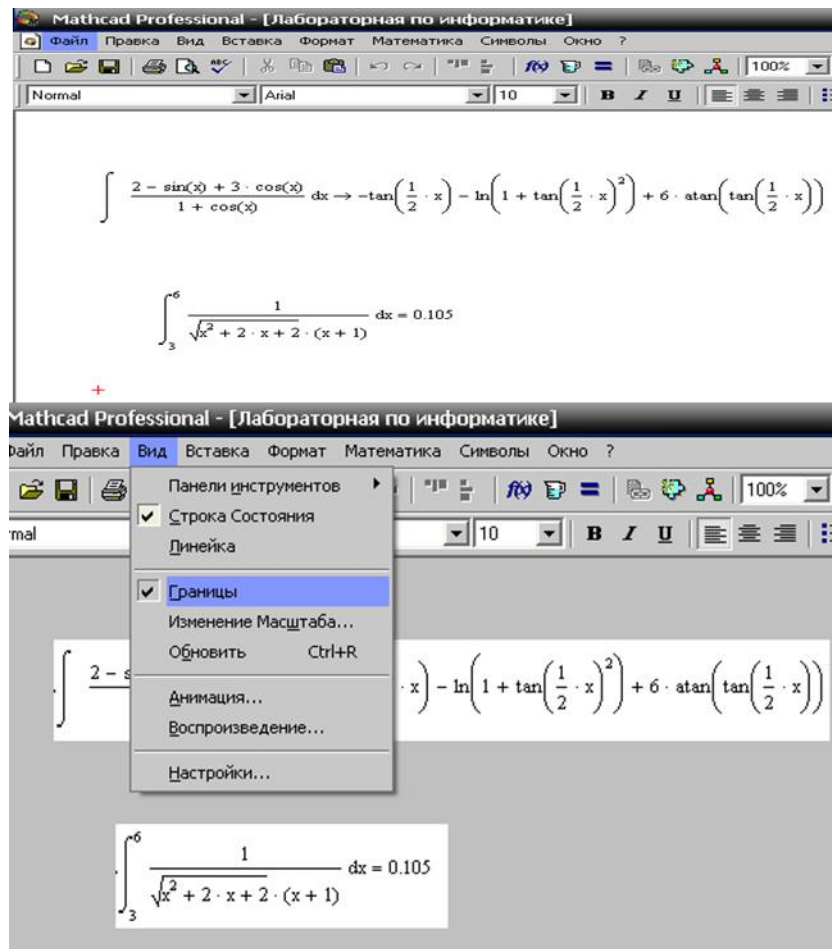


Рис. 2.1. Структура рабочего документа Mathcad

В других системах компьютерной математики (СКМ) интерфейс преимущественно представляется в виде командной строки, где сначала вводятся исходные данные, а результат вычислений выводится в следующую строку.

### Построение простых математических моделей

Первое, что необходимо сделать – переместить визир в требуемое место.



Ввод выражений осуществляется как путем нажатия клавиш и их комбинаций, так и выбором соответствующих кнопок на различных палитрах символов (рис. 2.2).

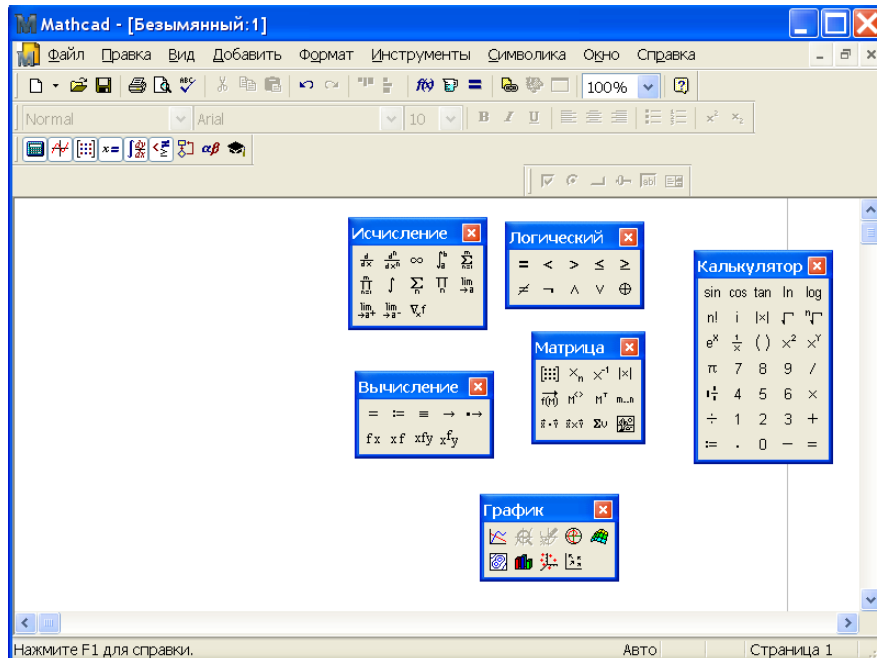


Рис. 2.2. Палитры инструментов Mathcad


Так, например, чтобы получить на экране выражение вида  $\frac{3}{4+5^2}$  нужно набрать на клавиатуре  $3/(4+5^2)$  или можно после ввода 5 нажать на арифметической палитре кнопку  $x^y$ , а затем ввести 2 в поле ввода.

**Изменение имени или числа.** Здесь редактор выражений ведет себя аналогично текстовому редактору. Для выполнения этой операции следует предварительно щелкнуть мышью на требуемом объекте, это поместит на него маркер ввода.

Замена и удаление оператора. Наиболее важный шаг в этом случае – расположить выделяющую рамку так, чтобы нужный оператор оказался в выделенном выражении наивысшим. Так, чтобы заменить в числителе дроби  $\frac{x - 3 \cdot a^2}{-4 + \sqrt{y + 1}}$  знак минус на знак плюс, следует полностью заключить числитель в выделяющую рамку, после чего нажать [Del] – это удалит наивысший оператор (в данном случае вычитание), заменив его пустым

полем в рамке (заметим, что повторное нажатие [Del] удалило бы числитель целиком),  $\frac{x \square 3 \cdot a^2}{-4 + \sqrt{y + 1}}$  и нажать [+], в результате чего на месте указанного пустого поля появится знак плюс.

**Вставка оператора** между двумя соседними символами выполняется путем установки между ними маркера ввода и нажатия соответствующей клавиши или кнопки на палитре символов.

Для вставки **операторов, требующих только одного операнда необходимо** выделить все выражение, которое должно быть в данном случае операндом, и нажать соответствующую комбинацию клавиш или кнопку на палитре символов. Подобным образом выполняется вставка скобок – выделив выражение, заключаемое в скобки, следует нажать .

Для **вычисления выражения** следует после его ввода нажать [=] или выбрать кнопку на арифметической палитре. В результате значение выражения вычисляется и отображается на рабочем листе. Если выражение использует переменные или функции, то они должны быть определены в рабочем документе заранее.

Текстовые области в Mathcad являются областями изменяемого размера, при этом информация, содержащаяся в них, при выполнении вычислений игнорируется. Как правило, текстовые области используются для отображения комментариев к математическим выражениям и графикам.

Для **создания текстовой области** необходимо щелчком мыши поместить визир в позицию, с которой предполагается начать текст, и выбрать команду **Текстовая область** меню **Вставка**. В результате появляется текстовая рамка с находящимся внутри маркером ввода. Теперь можно начинать вводить требуемый текст (рис. 2.3).

Для **редактирования** вводимого текста в Mathcad используются те же принципы, что и в большинстве текстовых редакторов.

**Выравнивание областей.** Эта операция также применяется к выделенным областям и вызывается выбором подменю **Выровнять области** меню **Формат** или кнопками на **Панели инструментов**. Возможно выравни-

нивание по нижней (вертикальное) и верхней (горизонтальное) границам (рис. 2.4).

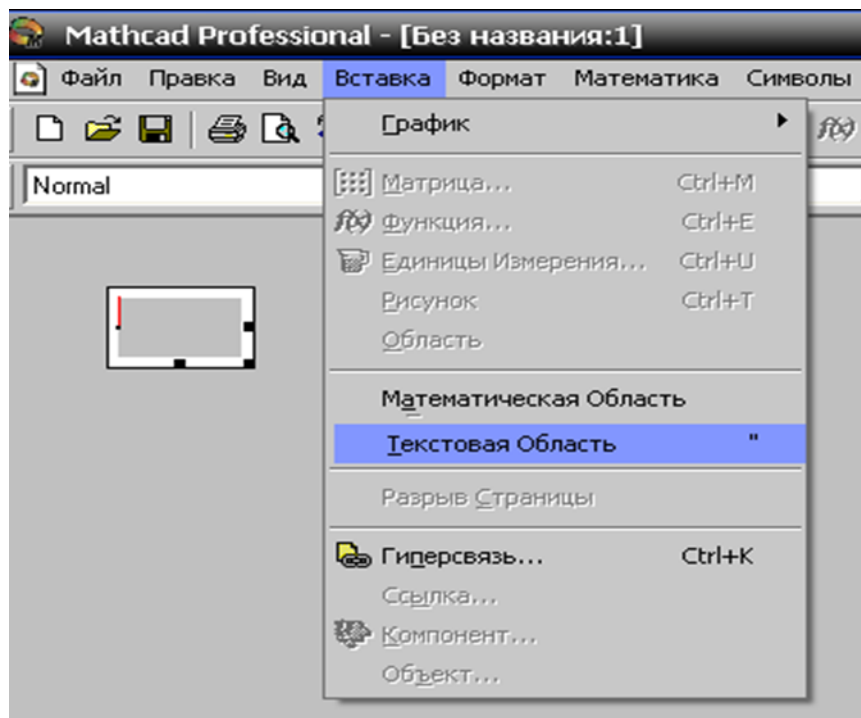


Рис. 2.3. Вставка текстовой области

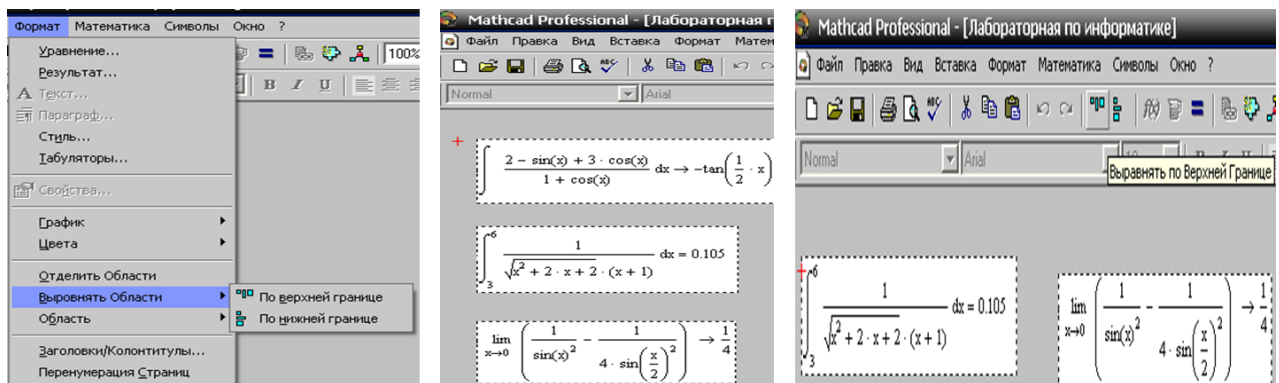


Рис. 2.4. Выравнивание областей по вертикали и по горизонтали

### Определение переменных:

- ввести имя переменной, подлежащей определению;
- нажать [:=] или выбрать кнопку := на арифметической палитре – результатом будет появление после имени переменной знака определения := ;
- ввести выражение, определяющее переменную.

**Определение функций** аналогично определению переменных. Основное различие состоит в том, что имя функции включает список формальных аргументов-параметров, заключенный в скобки. Заметим, что эти **аргументы должны быть именами**, а не более сложными выражениями.



- **Имена переменных и функций** могут содержать как латинские, так и греческие буквы. Для ввода греческой буквы следует ввести ее латинский эквивалент с учетом регистра или воспользоваться палитрой **Греческий алфавит**.

- Одна и та же переменная или функция может быть определена **неоднократно**.

- Все символы в имени должны быть введены с использованием одного и того же шрифта с одними и теми же параметрами.

- Если в определении функции используется имя переменной, которой **нет в списке аргументов**, то такая переменная должна быть **определена выше** определения функции.

- **Переменные и функции** могут быть определены **глобально**.

Кнопка  вместо кнопки  на **Панели вычисления**.

В Mathcad и других СКМ существуют predefined (встроенные) переменные (рис. 2.5) и функции. Некоторые из встроенных переменных (точность, количество максимальных итераций цикла и т.д.) можно изменять, другие (число пи, основание натурального логарифма и др.) изменить нельзя – константы, или это доступно не во всех программах.

Список всех **встроенных функций** с кратким описанием каждой функции вызывается командой **Функция** меню **Вставка** (рис. 2.6). Двойной щелчок мышью на выбранной функции или нажатие кнопки **Insert** вставляет ее в рабочий лист.

Переменная	Значение по умолчанию	Определение и использование
$\pi$	3.14159...	Число $\pi$ . В численных расчетах используется значение $\pi$ с учетом 15 значащих цифр. В символьных вычислениях $\pi$ сохраняет свое точное значение.
$e$	2.71828...	Основание натурального логарифма. В численных расчетах используется значение $e$ с учетом 15 значащих цифр. В символьных вычислениях $e$ сохраняет свое точное значение.
$\infty$	$10^{307}$	В численных расчетах – заданное большое число. В символьных вычислениях – бесконечность. Для ввода символа $\infty$ следует нажать [Ctrl]+[G]
TOL	$10^3$	Допустимая погрешность для численных расчетов
ORIGIN	0	Определяет индекс первого элемента массива

Рис. 2.5. Встроенные переменные Mathcad (имена зарезервированы), последние две переменные можно переопределить

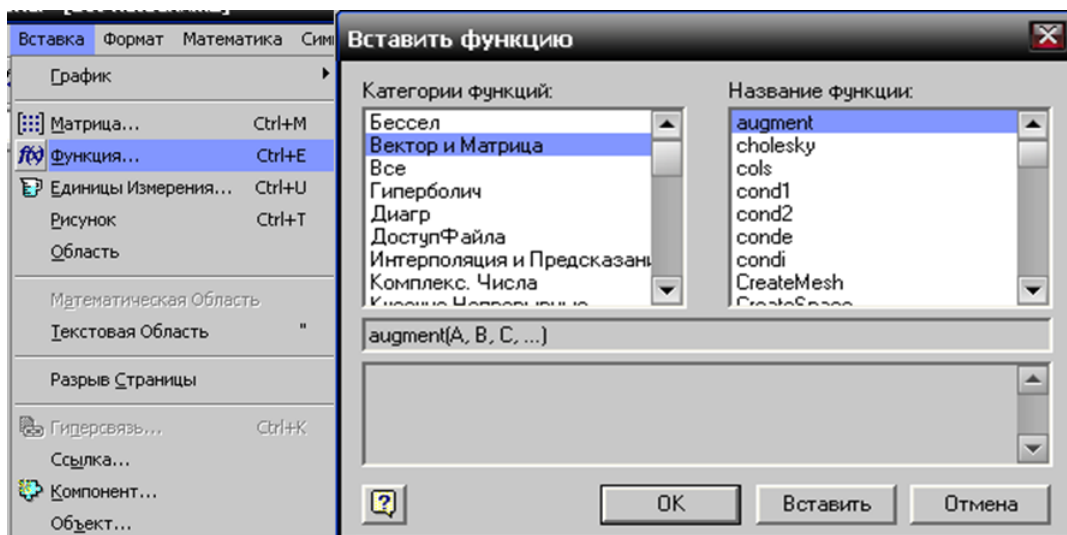


Рис. 2.6. Встроенные функции Mathcad

## 2.2. Расчет значений функции и представление результатов в виде вектор-строки и вектор-столбца

Задание: Даны функции:

$$1) y = \frac{d}{dx} e^{1+0,2x} \cdot \sin(x);$$

$$2) z = 0,01x^3$$

1. Ввести блоки текста как заглавие задачи и шаги ее исполнения.
2. Задать функции 1) и 2) как функции пользователя.

3. Ввести диапазон аргумента и шаг как отдельные переменные.
4. Вычислить вектор-строку значений функции 1) в заданном интервале и с заданным шагом.
5. Вычислить вектор-столбец значений функции 2) в заданном интервале и с заданным шагом.

*Решение:* Данные функции задаются как функции пользователя с помощью знака «:=». Вводится диапазон аргумента. Затем вычисляется вектор-строка значений функции  $y(x)$  и вектор-столбец значений функции  $z(x)$  в заданном интервале и с заданным шагом с помощью знака «= $\Rightarrow$ » (рис. 2.7).

**Задание №1. Расчет значений функции**

$$a := -10 \quad b := 10 \quad h := 1 \quad y(x) := \frac{d}{dx} \left( e^{1+0.2x \cdot \sin(x)} \right) \quad z(x) := 0.01x^3$$

$$x := 0..|a| + |b|$$

$$k_{0,x} := \frac{d}{dx} e^{1+0.2(x-a) \cdot \sin(x-a)} \quad k2_x := 0.01(x-a)^3$$

k =		0	1	2	3	4	5
	0	-1.636	-0.057	1.438	19.803	25.297	...

	0
0	10
1	13.31
2	17.28
3	21.97
4	27.44
5	33.75
6	40.96
k2 = 7	49.13
8	58.32
9	68.59
10	80
11	92.61
12	106.48
13	121.67
14	138.24
15	...

Рис. 2.7. Расчет значений функций

### 2.3. Построение двумерных графиков функции в декартовых координатах

*Задание:* Даны функции:

$$1) y = \frac{d}{dx} e^{1+0,2x \cdot \sin(x)} ;$$

$$2) z = 0,01x^3$$

1. Вывести графики двух функций одновременно на одном поле в декартовых координатах.

2. Настроить отображение маркеров и стиля линий для каждой функции.

3. Изучить прочие возможности по оформлению графика 2D функций.

*Решение:* Задается функция как функция пользователя (описано выше).

На панели **Графики** выбирается «двумерный график». В рабочем окне вводится обозначение функции с ее аргументом в скобках. Далее указывается сам аргумент (рис. 2.8, внизу). Если нужно отобразить два графика на одном поле, то в ячейке слева перечисляются обозначения функций и аргументов в скобках через запятую.

#### Задание №2. Двумерный график функции в декартовых координатах

$$y(x) := \frac{d}{dx} \left( e^{1+0,2x \cdot \sin(x)} \right) \quad z(x) := 0,01x^3$$

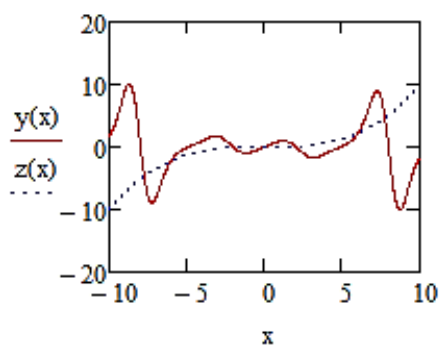


Рис. 2.8. Решение уравнений графическим методом

Разные кривые должны быть изображены различными цветами. Для форматирования графика нужно дважды щелкнуть на области графика. Для управления отображением построенных линий служит вкладка **Следы** (Traces) в открывшемся диалоговом окне. Текущий формат каждой линии

приведен в списке. Под списком расположены элементы управления, позволяющие изменять формат. Поле **Метка легенды** (Legend Label) задает описание линии, которое отображается только при сбросе флажка «Скрыть описание» (Hide Legend). Список **Символ** (Symbol) позволяет выбрать маркеры для отдельных точек, список **Линия** (Line) задает тип линии, список **Цвет** (Color) – цвет. Список **Тип** (Type) определяет способ связи отдельных точек, а список **Размер** (Width) – толщину линии [3].

Для создания матрицы на панели Toolbar (панель матричных инструментов) необходимо выбрать кнопку с функцией «Определение размеров матрицы» и вписать в ячейки нужные значения.

## 2.4. Задание функции двух переменных и матриц

*Задание:* Дана функция:

$$y = \left( \frac{\sin(x)}{x} \right) |\cos(t)|$$

1. Ввести блоки текста, как заглавие задачи и шаги ее исполнения.
2. Задать функцию, как функцию пользователя.
3. Ввести диапазоны изменения аргументов как матрицу 2x2 (аргументы – в колонках, границы диапазонов – в строках матрицы).

*Решение:* Данная функция задается как функция пользователя. Далее вводятся диапазоны изменения аргументов как матрица 2x2 (рис. 2.9).

### Задание №3. Функции двух переменных и матриц

$$y(x, t) := \left( \frac{\sin(x)}{x} \right) \cdot |\cos(t)|$$

$$\begin{pmatrix} -2\pi & -2\pi \\ 2\pi & 2\pi \end{pmatrix}$$

Рис. 2.9. Функция пользователя и диапазон изменения аргументов как матрица

## 2.5. Нахождение частных производных и их значений

*Задание:* Дана функция:




$$y = \left( \frac{\sin(x)}{x} \right) |\cos(t)|$$

1. Ввести блоки текста как заглавие задачи и шаги ее исполнения.
2. Определить две новые функции – как частные производные от данной по каждому из аргументов.
3. Вычислить значения частных производных в углах диапазона - сформировать четыре вектор-строки по два значения в каждом.

*Решение:* Задается функция, как функция пользователя.

Для того, чтобы найти производную, нужно: щелкнуть по свободно-му месту в рабочем документе, щелкнуть в панели Calculus по кнопке «d/dx», ввести с клавиатуры в помеченных позициях имя или выражение функции и аргумента.

Заключить все выражение в выделяющую рамку и щелкнуть по стро-ке **Symbolically** в пункте **Evaluate** меню **Symbolics** (или щелкнуть по кнопке «» в панели символьных операций **Symbolic**) [4].

Для нахождения значений производных, нужно ввести диапазон из-менений переменной (указано выше) и вывести значения, написав обозна-чение функции и используя знак «=>» (рис. 2.10).

**Задание №4. Частные производные**

$$y(x, t) := \left( \frac{\sin(x)}{x} \right) \cdot |\cos(t)|$$

$$\frac{d}{dt} y(x, t) \rightarrow -\frac{\text{signum}(\cos(t), 0) \cdot \sin(t) \cdot \sin(x)}{x}$$

$$\frac{d}{dx} y(x, t) \rightarrow \frac{|\cos(t)| \cdot \cos(x)}{x} - \frac{|\cos(t)| \cdot \sin(x)}{x^2}$$

$$x := -2\pi, 2\pi .. 2\pi \quad t := -2\pi, 2\pi .. 2\pi$$

$$\frac{d}{dx} y(x, t) =$$

-0.159
-0.159
0.159
0.159

$$\frac{d}{dt} y(x, t) =$$

0
0
0
0

$$(-0.159 \ -0.159 \ 0.159 \ 0.159)$$

Рис. 2.10. Частные производные и их значения

## 2.6. Построение трехмерного графика функции двух переменных

*Задание:* Дана функция:

$$y = \left( \frac{\sin(x)}{x} \right) |\cos(t)|$$

1. Ввести блоки текста, как заглавие задачи и шаги ее исполнения.
2. Вывести контурный график функции (изолиниями).
3. Вывести сетчатый график функции (с заливкой).
4. Изучить прочие возможности по оформлению графика 3D функции.

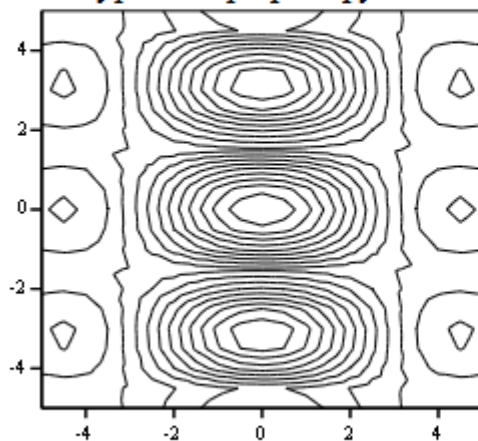
*Решение:* Задается функция, как функция пользователя. На палитре **График**, выбирается «Линии уровня» и вводится необходимая функция. В результате получается контурный график функции.

Для построения сетчатого графика функции на палитре **График** выбирается «График поверхности» и так же указывается необходимая функция (рис. 2.11).

### Задание №5. Трёхмерный график функции двух переменных

$$y(x, t) := \left( \frac{\sin(x)}{x} \right) \cdot |\cos(t)|$$

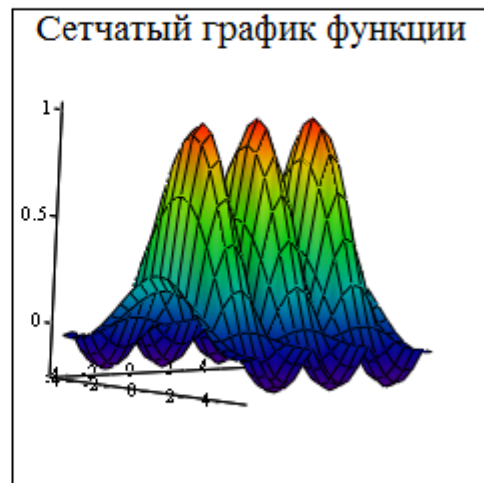
Контурный график функции



у

а

Сетчатый график функции



у

б

Рис. 2.11. Графики функции двух переменных:

а – контурный график функции;

б – сетчатый график функции

## 2.7. Требования к отчету по лабораторной работе

Отчет по лабораторной работе выполняется на листах белой бумаги формата А4 в печатном виде.

Отчет формируется в следующем порядке:

1. Титульный лист.
2. Содержание.
3. Цель работы.
4. Задание.
5. Теоретическая часть. Основы работы с системами компьютерной математики на примере Mathcad.
6. Практическая (расчетная) часть.
7. Выводы по результатам выполнения работы.
8. Библиографический список.
9. Приложение.

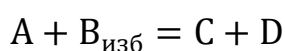
При оформлении отчета используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страниц ставятся по центру в верхнем колонтитуле.

## 3. Лабораторная работа 3. Использование систем компьютерной алгебры на примере MathCAD (часть 2)

**Цель работы:** Получение навыков решения расчетно-оформительских задач в вычислительной среде MathCAD.

### 3.1. Определение порядка химической реакции с использованием MathCAD

*Задание:* При проведении некоторой реакции:



замерялась концентрация компонента А. Были получены следующие данные:

Время, мин	0	3	5	7	10	15	25
C <sub>A</sub> , ммоль/л	10	7,40	6,34	5,50	4,64	3,63	2,54

Необходимо, используя MathCAD, определить порядок реакции.

Для этого нужно:

1) вычислить скорость реакции в каждый из моментов времени, воспользовавшись соотношением:

$$w_t = -\frac{\Delta C}{\Delta t} = \frac{C_{t-1} - C_t}{t - t_{-1}}$$

2) построить график зависимости  $\ln w$  от  $\ln C_A$  (где  $C_A$  есть среднее между соседними отсчетами), и функцией *slope* вычислить тангенс угла наклона этого графика (что является порядком реакции).

3) функцией *expfit()* вычислить коэффициенты  $a, b, c$  экспоненциальной регрессии вида  $f(x) = a \cdot e^{bx} + c$ , наилучшим образом приближающей экспериментальные данные.

4) построить на одном поле график зависимости  $C_A$  от  $t$ , и график функции регрессии  $f(x)$ .

*Решение:* На основе исходных данных составляются матрицы для времени и концентрации. По формуле вычисляется скорость реакции в каждый из моментов времени. Для этого из соответствующих матриц берутся порядковые элементы согласно формуле (рис. 3.1).

Затем составляются новые матрицы и матрицы значений натурального логарифма. По зависимостям натурального логарифма строится график (рис. 3.2). На панели инструментов есть соответствующая функция. И с помощью функции *slope* вычисляется тангенс угла наклона этого графика (что является порядком реакции). *Slope (vx, vy)* – Возвращает скаляр: наклон линии регрессии в смысле наименьших квадратов для данных из  $vx$  и  $vy$  [5].

Вычисляются коэффициенты  $a, b, c$  экспоненциальной регрессии вида  $f(x) = a \cdot e^{bx} + c$  с помощью функции *expfit (vx, vy, vg)* (возвращает вектор, содержащий коэффициенты  $(a, b$  и  $c)$  аппроксимирующего выражения вида  $a \cdot e^{bx} + c$ , график которого лучшим образом приближается к

точкам, чьи координаты хранятся в векторах  $ix$  и  $iy$  (вектор  $vg$  содержит первое приближение к решению)) [6].

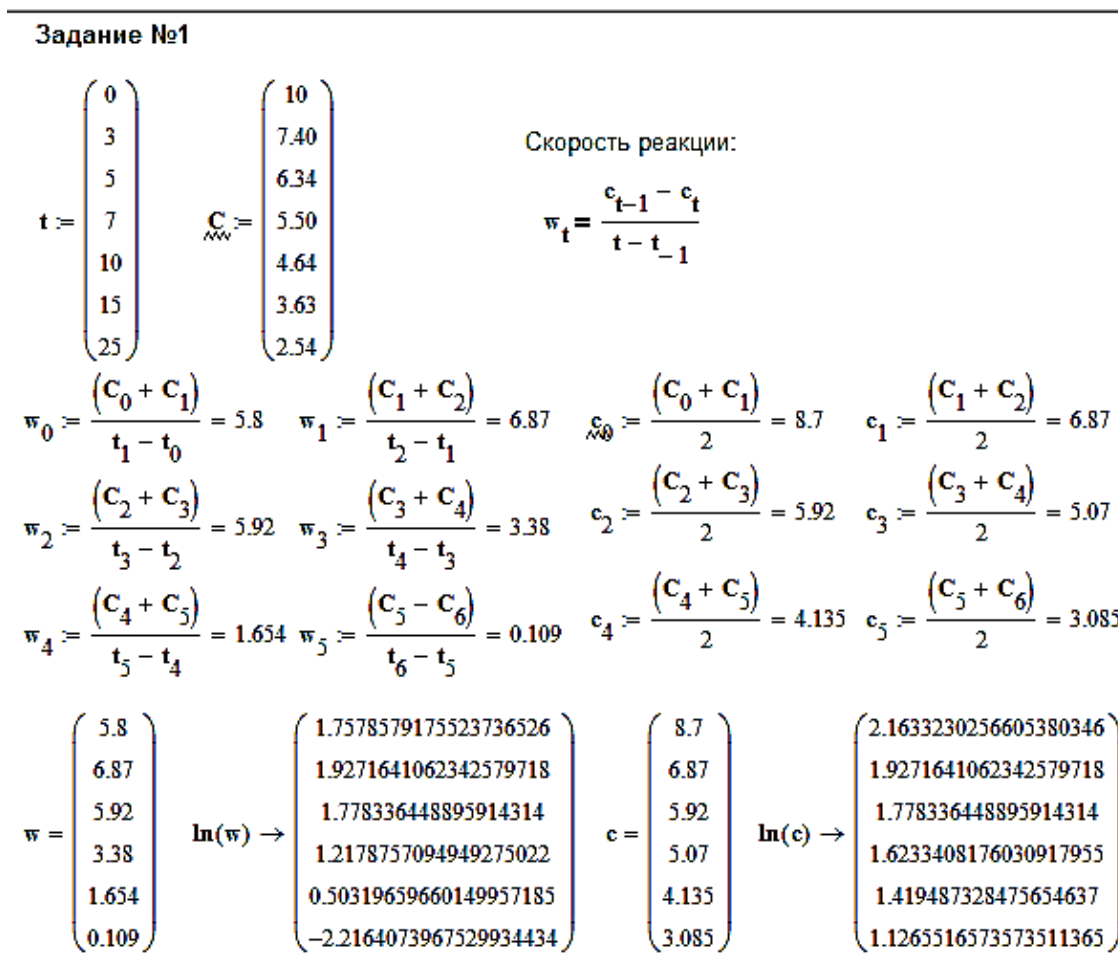


Рис. 3.1. Скорость реакции в каждый из моментов времени

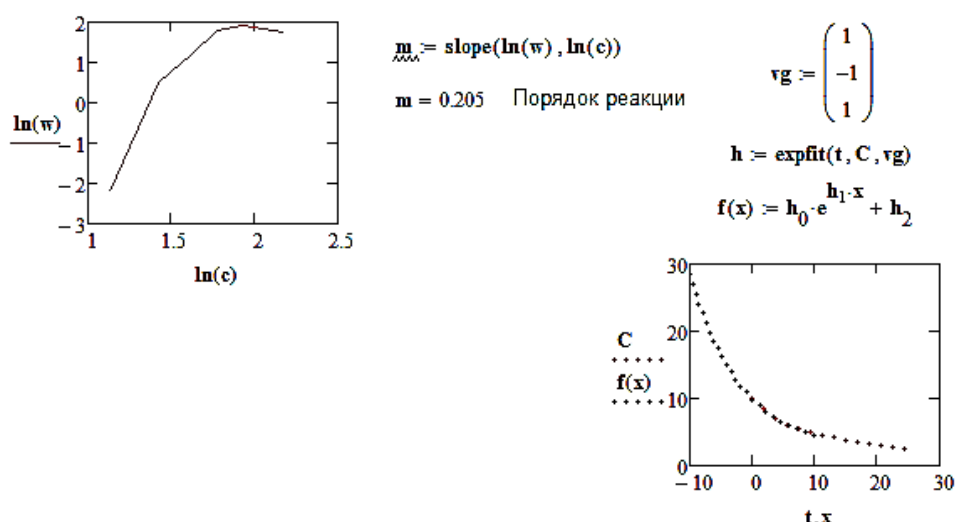


Рис. 3.2. График зависимости  $\ln(w)$  от  $\ln(c)$  и график зависимости концентрации от времени с функцией регрессии  $f(x)$

### 3.2. Решение системы линейных уравнений с помощью встроенной функции и матричного метода

*Задание:* Решить систему уравнений двумя способами (встроенной функцией и с использованием матричного метода):

$$\begin{cases} 4x - y + 2 = 18 \\ -2x + 5y = 10 \end{cases}$$

Вычислить интеграл и проверить результат дифференцированием. Постараться добиться полного совпадения подынтегрального выражения с результатом дифференцирования.

$$\int_0^{+\infty} \frac{dx}{x^2 + 6x + 7}$$

*Решение:*

1. Необходимо задать начальные приближения для всех неизвестных, входящих в систему уравнений. Mathcad решает уравнения при помощи итерационных методов. На основе начального приближения строится последовательность, сходящаяся к искомому решению.

2. Ввести ключевое слово *Given*. Оно указывает Mathcad, что далее следует система уравнений. При печати слова *Given* можно использовать любой шрифт, прописные и строчные буквы. Убедитесь, что при этом Вы не находитесь в текстовой области или параграфе.

3. Ввести уравнения в любом порядке ниже ключевого слова *Given*. Удостоверьтесь, что между левыми и правыми частями уравнений стоит символ =. Используйте [Ctrl]= для печати символа =. Между левыми и правыми частями неравенств может стоять любой из символов <, >, ≤, и ≥.

4. Ввести любое выражение, которое включает функцию *Find*. При печати слова *Find* можно использовать шрифт любого размера, произвольный стиль, прописные и строчные буквы.

Функция *Find* возвращает найденное решение следующим образом:

1. Если функция *Find* имеет только один аргумент, то она возвращает решение уравнения, расположенного между ключевым словом *Given* и функцией *Find*.

2. Если функция *Find* имеет более одного аргумента, то она возвращает ответ в виде вектора. Например, *Find* (*z1*, *z2*) возвращает вектор, содержащий значения *z1* и *z2*, являющиеся решением системы уравнений.

Ключевое слово *Given*, уравнения и неравенства, которые следуют за ним, и какое-либо выражение, содержащее функцию *Find*, называются *блоком решения уравнений* [7].

Матричный метод:

Создается матрица *A* и вектор-столбец *b*:

$$A = \begin{pmatrix} 4 & -1 \\ -2 & 5 \end{pmatrix}, b = \begin{pmatrix} 16 \\ 10 \end{pmatrix}.$$

Находим обратную матрицу  $A^{-1}$  и умножаем ее на вектор-столбец *b*:  $\vec{x} = A^{-1} \cdot \vec{b}$ . Получаем вектор решения [8] (рис. 3.3).

---

### Задание №2

Использование встроенной функции

*Given*

$$4 \cdot x - y + 2 = 18$$

$$-2 \cdot x + 5 \cdot y = 10$$

$$\text{Find}(x,y) \rightarrow \begin{pmatrix} 5 \\ 4 \end{pmatrix}$$

Использование матричного метода

$$A := \begin{pmatrix} 4 & -1 \\ -2 & 5 \end{pmatrix} \quad b := \begin{pmatrix} 16 \\ 10 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 0.278 & 0.056 \\ 0.111 & 0.222 \end{pmatrix}$$

$$X := A^{-1} \cdot b = \begin{pmatrix} 5 \\ 4 \end{pmatrix}$$

Ответ: *x*=5, *y*=4

Рис. 3.3. Решение уравнений с использованием встроенной функции и матричного метода

Заданный определенный интеграл вычисляется с помощью встроенной функции на панели инструментов. Для проверки берется неопределен-

ный интеграл от заданной подынтегральной функции. Затем производится дифференцирование. Полное совпадение с подынтегральной функцией приведено на рис. 3.4.

Вычислить интеграл:

$$\int_0^{\infty} \frac{1}{x^2 + 6 \cdot x + 7} dx \rightarrow \frac{\sqrt{2} \cdot (\ln(\sqrt{2} + 3) - \ln(3 - \sqrt{2}))}{4}$$

$$\int \frac{1}{x^2 + 6 \cdot x + 7} dx \rightarrow \frac{\sqrt{2} \cdot \operatorname{atanh}\left[\frac{\sqrt{2} \cdot (x + 3)}{2}\right]}{2}$$

Проверка:

$$\frac{d}{dx} \left[ \frac{\sqrt{2} \cdot \operatorname{atanh}\left[\frac{\sqrt{2} \cdot (x + 3)}{2}\right]}{2} \right] \rightarrow \frac{1}{2 \cdot \left[ \frac{(x + 3)^2}{2} - 1 \right]} \rightarrow \frac{1}{(x + 3)^2 - 2}$$

Рис. 3.4. Решение уравнений с использованием встроенной функции и матричного метода

### 3.3. Определение детерминанта матрицы

*Задание:*

Найти детерминант матрицы  $R = \begin{bmatrix} 7 & 21 & 48 \\ 3 & 9 & 36 \\ 2 & 12 & 24 \end{bmatrix}$

1) привести заданную матрицу  $R$  к ступенчатому виду методом Гаусса;

2) транспонировать и найти ранг транспонированной матрицы.

*Решение:* Найти детерминант матрицы можно с помощью встроенной функции на панели инструментов:  $|M|$ . Функция  $rref(M)$  выполняет элементарные операции над строками матрицы  $M$  – приводит ее к ступенчатому виду с единичной матрицей в первых столбцах, т.е. выполняет прямой и обратный ход Гауссова исключения. Транспонировать можно с помощью встроенной функции на панель инструментов  $M^T$ . Осуществить проверку



правильности нахождения ранга матриц с помощью встроенной функции MathCad  $rank()$  (рис. 3.5).

### Задание №3

$$M := \begin{pmatrix} 7 & 21 & 48 \\ 3 & 9 & 36 \\ 2 & 12 & 24 \end{pmatrix}$$

$$|M| = -648$$

$$\text{rref}(M) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M^T = \begin{pmatrix} 7 & 3 & 2 \\ 21 & 9 & 12 \\ 48 & 36 & 24 \end{pmatrix}$$

$$\text{rank}(M^T) = 3$$

Рис. 3.5. Действия над матрицей

### 3.4. Нахождение экстремума функции

*Задание:* Найти экстремумы функции:  $z=4x^2-6xy-34x+5y^2+42y+7$ .

*Решение:* Определяется функция двух переменных. Далее оценивается по графику положение экстремумов.

Для поиска экстремумов в Mathcad существует две функции:

- *Minimize* ( $y,x$ ) – для поиска значения  $x$ , соответствующего локальному минимуму функции  $y(x)$ ;

- *Maximize* ( $y,x$ ) – для поиска значения  $x$ , соответствующего локальному максимуму функции  $y(x)$ .

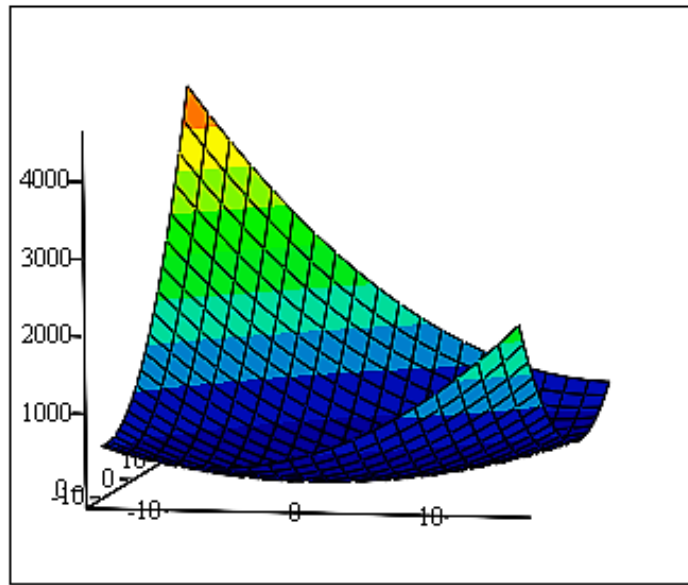
Так как  $y(x)$  может иметь несколько локальных экстремумов, а функции *Minimize* ( $y,x$ ) и *Maximize* ( $y,x$ ) позволяет найти только одно значение, то в Mathcad дополнительно задается начальное приближение переменной  $x$ . В результате находится значение экстремума функции  $y(x)$ , ближайшее к заданному начальному приближению переменной  $x$ .

При поиске экстремумов посредством записи неравенства может быть задана некоторая область, в границах которой производится отыска-

ние экстремума. В этом случае записи неравенства предшествует введение ключевого слова *given*. На рис. 3.6, 3.7 представлен пример нахождения экстремумов функции в Mathcad [9].

**Задание №4**

$$z(x,y) := 4 \cdot x^2 - 6 \cdot x \cdot y - 34 \cdot x + 5 \cdot y^2 + 42 \cdot y + 7$$



z

Рис. 3.6. Функция и ее график поверхности

$$x := 1 \quad y := 1$$

Given

$$-10 \leq x \leq 10$$

$$-10 \leq y \leq 10$$

$$P := \text{Minimize}(z, x, y) = \begin{pmatrix} 2 \\ -3 \end{pmatrix}$$

Given

$$-50 \leq x \leq 50$$

$$-50 \leq y \leq 50$$

$$p := \text{Maximize}(z, x, y) = \begin{pmatrix} -50 \\ 50 \end{pmatrix}$$

Рис. 3.7. Экстремумы функции

### 3.5. Требования к отчету по лабораторной работе

Отчет по лабораторной работе выполняется на листах белой бумаги формата А4 в печатном виде.

Отчет формируется в следующем порядке:

1. Титульный лист.
2. Содержание.
3. Цель работы.
4. Задание.
5. Теоретическая часть.
6. Практическая (расчетная) часть.
7. Выводы по результатам выполнения работы.
8. Библиографический список.
9. Приложение.

При оформлении отчета используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страниц ставятся по центру в верхнем колонтитуле.

## 4. Лабораторная работа 4. Использование пакета прикладных программ MatLAB для решения математических и инженерных задач

**Цель работы:** Изучение интерфейса и основных принципов ведения вычислений в среде MatLAB.

Решение вычислительных задач в среде MatLAB.

Отработка навыков решения расчетно-оформительских задач в среде MatLAB.

### 4.1. Пакет прикладных программ MatLAB

MATLAB (сокращение от англ. Matrix Laboratory, в русском языке произносится как Матлаб) – пакет прикладных программ для решения задач технических вычислений и одноименный язык программирования, применяемый в этом пакете. Пакет используют более миллиона инженер-

ных и научных работников, он работает на большинстве современных операционных систем, включая Linux, Mac OS, Solaris (начиная с версии R2010b поддержка Solaris прекращена) и Microsoft Windows.

MATLAB как язык программирования был разработан Кливом Моулером (англ. Cleve Moler) в конце 1970-х годов, когда он был деканом факультета компьютерных наук в Университете Нью-Мексико. Целью разработки служила задача дать студентам факультета возможность использования программных библиотек Linpack и EISPACK без необходимости изучения Фортрана. Вскоре новый язык распространился среди других университетов и был с большим интересом встречен учеными, работающими в области прикладной математики. До сих пор в Интернете можно найти версию 1982 года, написанную на Фортране, распространяемую с открытым исходным кодом. Инженер Джон Литтл (англ. John N. (Jack) Little) познакомился с этим языком во время визита Клива Моулера в Стэнфордский университет в 1983 году. Поняв, что новый язык обладает большим коммерческим потенциалом, он объединился с Кливом Моулером и Стивом Бангертом (англ. Steve Bangert). Совместными усилиями они переписали MATLAB на С и основали в 1984 году компанию The MathWorks для дальнейшего развития. Эти переписанные на С библиотеки долгое время были известны под именем JASCRAS. Первоначально MATLAB предназначался для проектирования систем управления (основная специальность Джона Литтла), но быстро завоевал популярность во многих других научных и инженерных областях. Он также широко использовался и в образовании, в частности, для преподавания линейной алгебры и численных методов.

Язык MATLAB является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования.

Программы, написанные в MATLAB, бывают двух типов – функции и скрипты. Функции имеют входные и выходные аргументы, а также собственное рабочее пространство для хранения промежуточных результатов

вычислений и переменных. Скрипты же используют общее рабочее пространство. Как скрипты, так и функции сохраняются в виде текстовых файлов и компилируются в машинный код динамически. Существует также возможность сохранять так называемые pre-parsed программы – функции и скрипты, обработанные в вид, удобный для машинного исполнения. В общем случае такие программы выполняются быстрее обычных, особенно если функция содержит команды построения графиков.

Основная особенность языка MATLAB – его широкие возможности по работе с матрицами, которые создатели языка выразили в лозунге «думай векторно» (англ. Think vectorized).

## 4.2. Работа с переменными и массивами

*Задание:*

1. Определить переменную  $M$  как массив  $3 \times 4$  элементов

-47	17	-32
-35	-15	1
18	-50	-30
-37	35	0

2. Извлечь вторую строку в отдельный массив, присвоив его в новую переменную.
3. Вычислить максимальное значение и его порядковый номер в новом массиве.
4. Извлечь третий столбец исходного массива в отдельную переменную.
5. Вычислить треть суммы элементов этого массива, присвоить в переменную  $k$ .
6. Подсчитать количество элементов в  $M$ , больших по абсолютному значению чем  $k$ .

*Решение:* В исполняемый файл MatLAB (формата \*.m) построчно записываются команды (рис. 4.1):

1. Определяется переменная  $M$  как массив  $3 \times 4$  элементов (строка 2).
2. Извлекается вторая строка в отдельный массив (строка 4).
3. Далее вычисляется максимальное значение и номер его по счету в этом массиве (строки 6 и 7 соответственно).

4. Затем извлекается третий столбец, вычисляется треть суммы элементов этого массива (присваивается в переменную k) и вычисляется количество элементов в M, больших по абсолютному значению чем k (строки 9, 11, 13).

```
Lab3_ch1_Task1.m* x +
1
2 - M=[-47 17 -32; -35 -15 1; 18 -50 -30; -37 35 0]
3
4 - M2=M(2,1:3)
5
6 - max(M2)
7 - N=find(M2==max(M2(:)))
8
9 - M3=M(1:4,3)
10
11 - k=(sum(M3)/3)
12
13 - n=sum(abs(M(:))>k)
```

Рис. 4.1. Команды, записанные в \*.m-файл

В результате выполнения этого файла в командном окне будет следующий результат (рис. 4.2, 4.3):

```
>> Lab3_ch1_Task1

M =

    -47     17    -32
    -35    -15     1
     18    -50    -30
    -37     35     0

M2 =

    -35    -15     1

ans =

     1

N =

     3
```

Рис. 4.2. Командное окно после запуска \*.m-файла (пункты 1–3)

```
M3 =  
  
-32  
  1  
-30  
  0  
  
k =  
  
-20.3333  
  
n =  
  
12
```

Рис. 4.3. Командное окно после запуска \*.m-файла (пункт 4)

### 4.3. Работа с функцией в \*.m-файлах

*Задание:* Дана функция:

$$y = \begin{cases} \sqrt{1 + |x|}, & x \leq 0 \\ \frac{1 + 3x}{\sqrt[3]{1 + x + 2}}, & x > 0 \end{cases}$$

1. Создать \*.m-файл с определением функции по своему варианту.
2. Вычислить минимальные и максимальные значения функции на интервале  $[-5 \ 5]$ .
3. Построить график функции на этом интервале.
4. Нанести подписи осей графика и заголовок всего графика на изображение.

*Решение:* Создается \*.m-файл с определением данной функции, причем имя файла всегда должно совпадать с именем функции.

В коде функции (рис. 4.4) представлено вычисление значений функции для переменной меньшей либо равной нулю (строка 4) и больше нуля (строка 6).

```

Lab3_ch1_Task2ghg.m  task2f.m  +
1  function y=task2f(x)
2  for i=1:length(x)
3      if x(i)<=0
4          y(i)=sqrt(1+abs(x(i)));
5      else
6          y(i)=(1+3*x(i))/(power((1+x(i)),(1/3))+2);
7      end
8  end
9  end

```

Рис. 4.4. \*.m-файл с определением функции

Для построения графика функции на интервале  $[-5, 5]$  с шагом 0,1 создается переменная (массив)  $x$ , содержащая все точки интервала. Для этого через «:» перечисляются [левая граница интервала]:[шаг]:[правая граница интервала]. Переменная  $y1$  содержит массив значений функции, полученных в этих точках. Далее с помощью встроенной функции *plot()*, строится график с включенной сеткой (*grid on*). Максимум и минимум функции вычисляются с помощью встроенной функций *fminbnd*. В примере ниже найден только минимум. Обязательно нужно подписать оси графика (*xlabel*, *ylabel*) и заголовок (*title*) (рис. 4.5).

```

x=-5:0.01:5;
y1=task2f(x);
plot(x,y1,'b'), grid on;
[x,y1]=fminbnd(@task2f,-5,5)
set(0,'DefaultAxesFontSize',14,'DefaultAxesFontName','Times New Roman')
title('График функции','Times New Roman')
xlabel('Ось x')
ylabel('Ось y')

```

Рис. 4.5. \*.m-файл с командами

Окно с построенным графиком функции представлено на рис. 4.6.

Минимальное и максимальное значение функции будет выведено в командном окне.



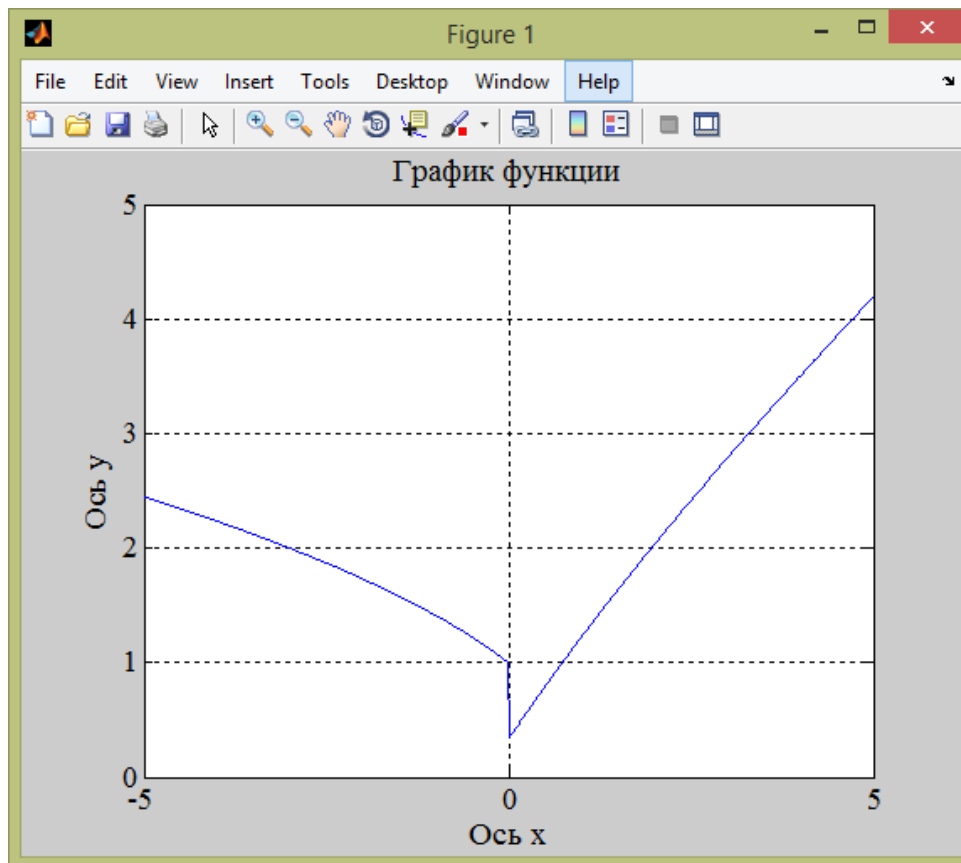


Рис. 4.6. График функции

#### 4.4. Работа с данными из внешних файлов

*Задание:* Дана функция:

$$y = \begin{cases} \sqrt{1 + |x|}, & x \leq 0 \\ \frac{1 + 3x}{\sqrt[3]{1 + x} + 2}, & x > 0 \end{cases}$$

1. Протабулировать (вычислить значения функции в точках) функцию на интервале  $[-15, 15]$  с шагом 0,01.
2. Не выводя результаты на экран, записать данные в текстовой файл с тремя колонками: №,  $x$ ,  $f(x)$ . Где № – номер точки по счету (всего должно получиться 3000 строк).
3. Считать получившийся файл в новую переменную как массив символов.

*Решение:* Для вычисления значений заданной функции в точках на интервале  $[-15, 15]$  с шагом 0,01 в \*.m-файле создается переменная (массив)  $x$ , содержащая все точки интервала. Для этого через «:»

перечисляются [левая граница интервала]:[шаг]:[правая граница интервала].

Далее вводится заданная функция. Здесь (строка 2) указана ссылка на функцию, созданную в предыдущем задании.

Данные записываются в текстовый файл с тремя колонками: №,  $x$ ,  $f(x)$  при помощи функции *fprintf*. Для этого используется параметр функции, позволяющий указать формат записи в файл.

*FileID* – это <идентификатор файла>. Он указывает на файл, с которым предполагается работать. Чтобы получить идентификатор, используется функция:

<идентификатор файла> = *fopen*(<имя файла>, <режим работы>);

где параметр <режим работы> может принимать значения, приведенные в табл. 4.1.

Таблица 4.1

#### Режимы работы с файлами в MatLab

Параметр <Режим работы>	Описание
'r'	чтение
'w'	запись (стирает предыдущее содержимое файла)
'a'	добавление (создает файл, если его нет)
'r+'	чтение и запись (не создает файл, если его нет)
'w+'	чтение и запись (очищает прежнее содержимое или создает файл, если его нет)
'a+'	чтение и добавление (создает файл, если его нет)
'b'	дополнительный параметр, означающий работу с бинарными файлами, например, 'wb', 'rb', 'rb+', 'ab' и т.п.

Далее с помощью функции *fileread* считывается получившийся файл в новую переменную как массив символов (рис. 4.7, 4.8).

```

Lab3_ch1_Task3.m  x  +
1 -   x=-15:0.01:15;
2 -   y=task2f(x);
3 -   fileID = fopen('funcc2.txt','w');
4 -   for i=1:length(x)
5 -       fprintf(fileID,'%6.0f %6.4f %6.4f\r\n',i ,x(i),y(i));
6 -   end
7 -   fileContents = fileread('funcc2.txt')

```

Рис. 4.7. \*m-файл с заданием

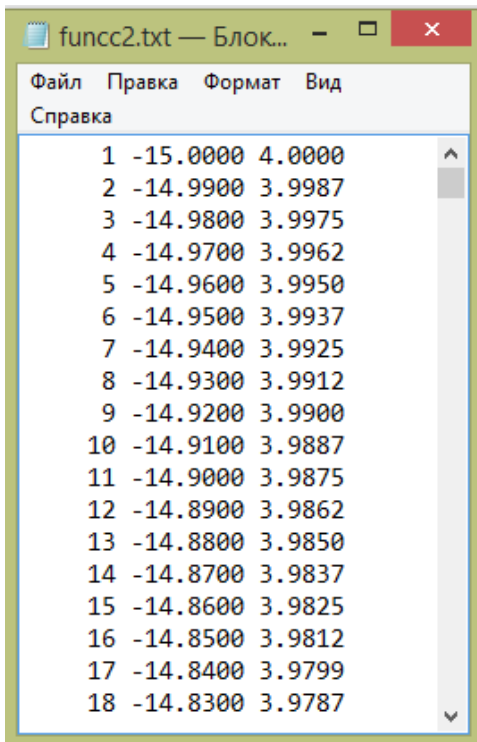


Рис. 4.8. Данные в текстовом файле

После выполнения считывания, в командном окне выводится переменная *fileContents* (рис. 4.9).

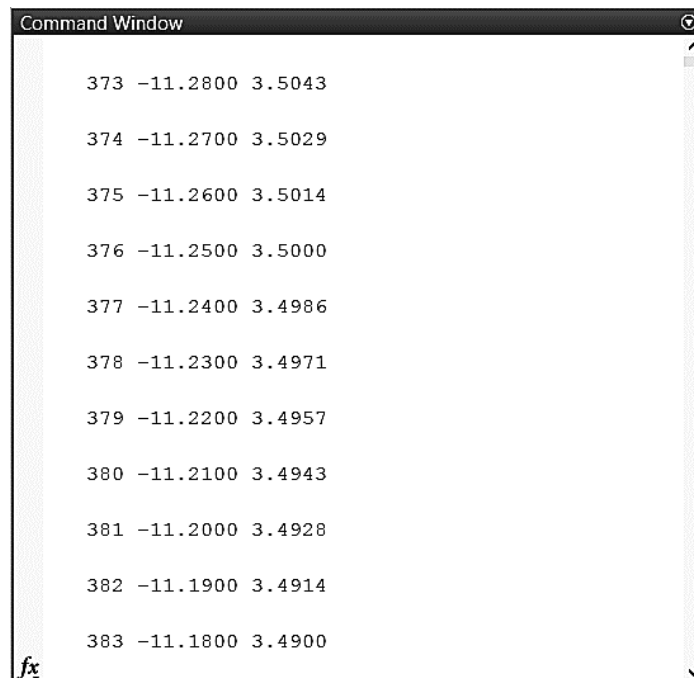


Рис. 4.9. Массив символов из файла func2.txt

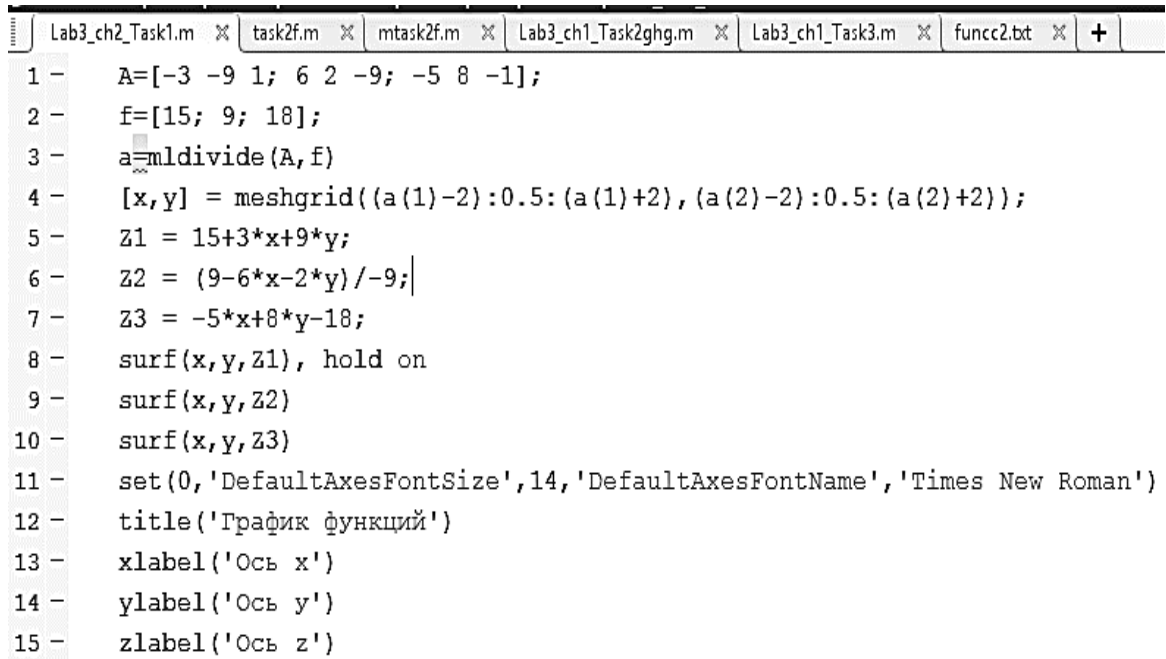
## 4.5. Решение системы линейных алгебраических уравнений

Задание: Дана система линейных уравнений:

$$\begin{cases} -3 \cdot x - 9 \cdot y + z = 15 \\ 6 \cdot x + 2 \cdot y - 9 \cdot z = 9 \\ -5 \cdot x + 8 \cdot y - z = 18 \end{cases}$$

1. Решить систему линейных алгебраических уравнений.
2. На одном графике изобразить все три плоскости в окрестностях точки пересечения.

Решение: Система линейных алгебраических уравнений решается с помощью встроенной функции *mldivide* (рис. 4.10). Для того чтобы построить графики поверхностей, используется функция *meshgrid*.



```
Lab3_ch2_Task1.m × task2f.m × mtask2f.m × Lab3_ch1_Task2ghg.m × Lab3_ch1_Task3.m × func2.txt × +
1 - A=[-3 -9 1; 6 2 -9; -5 8 -1];
2 - f=[15; 9; 18];
3 - a=mldivide(A,f)
4 - [x,y] = meshgrid((a(1)-2):0.5:(a(1)+2), (a(2)-2):0.5:(a(2)+2));
5 - z1 = 15+3*x+9*y;
6 - z2 = (9-6*x-2*y)/-9;
7 - z3 = -5*x+8*y-18;
8 - surf(x,y,z1), hold on
9 - surf(x,y,z2)
10 - surf(x,y,z3)
11 - set(0,'DefaultAxesFontSize',14,'DefaultAxesFontName','Times New Roman')
12 - title('График функций')
13 - xlabel('Ось x')
14 - ylabel('Ось y')
15 - zlabel('Ось z')
```

Рис. 4.10. \*m-файл с заданием

Решение системы приведено на рис. 4.11.

$$\begin{cases} x = -4,0311 \\ y = -0,7512 \\ z = -3,8543 \end{cases}$$

```

Command Window
>> Lab3_ch2_Task1

a =

    -4.0311
    -0.7512
    -3.8543

```

Рис. 4.11. Решение системы

На одном графике (рис. 4.12) изображены все три плоскости в окрестностях найденной ранее точки пересечения.

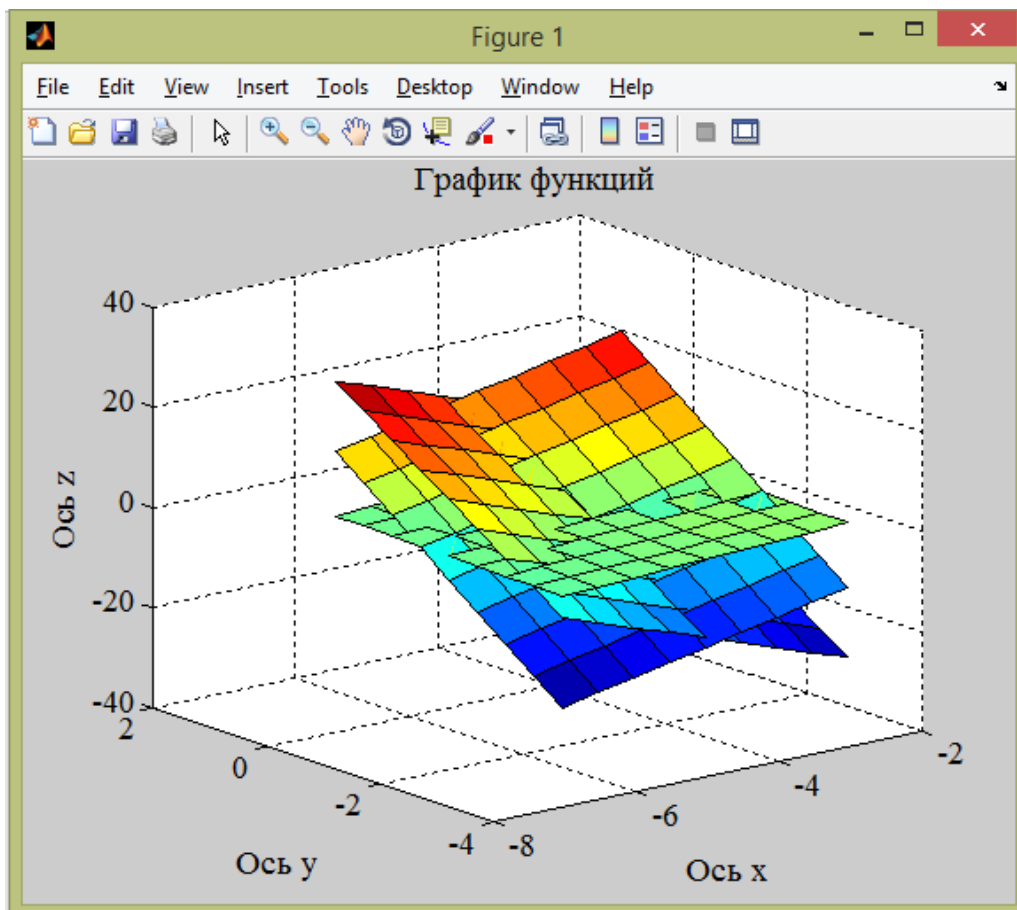


Рис. 4.12. График трех плоскостей

*Проверка:*

$$\begin{cases}
 -3 \cdot (-4,0311) - 9 \cdot (-0,7512) - 3,8543 = 15 \\
 6 \cdot (-4,0311) + 2 \cdot (-0,7512) - 9 \cdot (-3,8543) = 9 \\
 -5 \cdot (-4,0311) + 8 \cdot (-0,7512) + 3,8543 = 18
 \end{cases}$$

#### 4.6. Реализация метода наименьших квадратов в MATLAB

**Задание:** Подберите коэффициенты полиномов  $y = a_0 + a_1x + a_2x^2 + a_3x^3$  и  $y = b_0 + b_1x + b_2x^2$  методом наименьших квадратов (МНК) с помощью функции *polyfit* ( $x, y, k$ ), где  $x$  – массив абсцисс экспериментальных точек,  $y$  – массив ординат (табл. 4.2),  $k$  – степень полинома.

Затем вычислите значение полинома на всем интервале (максимум  $u$ ) с использованием функции *polyval* ( $kf, t$ ), где  $kf$  – массив коэффициентов полинома,  $t$  – точка, в которой необходимо вычислить значение (или момент времени для реального случая).

Постройте графики подобранных зависимостей (в одном окне) и подпишите оси и легенду.

Обозначьте на графиках экспериментальные точки.

Таблица 4.2

Исходные данные

X (время замера)					Y (эксперимент)				
1	2	3	4	5	49	56	51	68	53
6	7	8	9	10	68	43	55	47	56
11	12	13	14	15	56	28	37	36	54
16	17	18	19	20	65	47	58	55	39

**Решение:** Производится подбор коэффициентов полиномов  $y = a_0 + a_1x + a_2x^2 + a_3x^3$  и  $y = b_0 + b_1x + b_2x^2$  методом наименьших квадратов (МНК) с помощью функции *polyfit*( $x, y, k$ ). Функция  $p = \text{polyfit}(x, y, k)$  находит коэффициенты полинома  $p(x)$  степени  $k$ , который аппроксимирует функцию  $y(x)$  в смысле метода наименьших квадратов. Выходом является строка  $p$  длины  $n+1$ , содержащая коэффициенты аппроксимирующего полинома.

Затем вычисляется значение полинома на всем интервале (максимум  $u$ ) с использованием функции *polyval*( $kf, t$ ) (рис. 4.13–4.16). Функция *polyval*( $kf, t$ ) вычисляет значение полинома с коэффициентами  $kf$  в точке  $t$ . Затем строятся графики подобранных зависимостей (рис. 4.17).

```

Lab3_ch3_Task1.m x Lab3_ch3_Task2.m x task3.m x +
1   %Подбор коэффициентов полиномов
2   x=[1:1:20]
3   y=[49 56 51 68 53 68 43 55 47 56 56 28 37 36 54 65 47 58 55 39]
4   kf1=polyfit(x,y,3)
5   kf2=polyfit(x,y,2)
6   polyval(kf1,x)
7   polyval(kf2,x)
8   %Построение графиков
9   y1=53.0157+1.6639*x-0.3069*x.^2+0.0112*x.^3
10  y2=58.9447-1.3614*x+0.0446*x.^2
11  plot(x,y1,'g'), hold on
12  plot(x,y2,'r'), hold on
13  plot(x,y,'ko'), hold off, grid on
14  set(0,'DefaultAxesFontSize',14,'DefaultAxesFontName','Times New Roman')
15  title('График полиномов')
16  xlabel('Время замера')
17  ylabel('Эксперимент')
18  legend('Полином 3-й степени','Полином 2-й степени','Экспериментальные точки')

```

Рис. 4.13. \*.m-файл с функциями

```

Command Window
>> Lab3_ch3_Task1

x =

Columns 1 through 18

     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18

Columns 19 through 20

    19    20

y =

Columns 1 through 18

    49    56    51    68    53    68    43    55    47    56    56    28    37    36    54    65    47    58

Columns 19 through 20

    55    39

```

Рис. 4.14. Экспериментальные точки

```

kf1 =

    0.0112    -0.3069    1.6639    53.0157

kf2 =

    0.0446    -1.3614    58.9447

```

Рис. 4.15. Коэффициенты полиномов 3-й и 2-й степени

```

Command Window
ans =

Columns 1 through 11
54.3839 55.2051 55.5465 55.4748 55.0572 54.3604 53.4516 52.3976 51.2654 50.1220 49.034

Columns 12 through 20
48.0693 47.2940 46.7752 46.5800 46.7752 47.4280 48.6051 50.3737 52.8006

ans =

Columns 1 through 11
57.6279 56.4003 55.2619 54.2127 53.2527 52.3819 51.6002 50.9078 50.3046 49.7906 49.365

Columns 12 through 20
49.0302 48.7838 48.6266 48.5585 48.5797 48.6901 48.8897 49.1785 49.5565
fx >>

```

Рис. 4.16. Значения полиномов на всем интервале

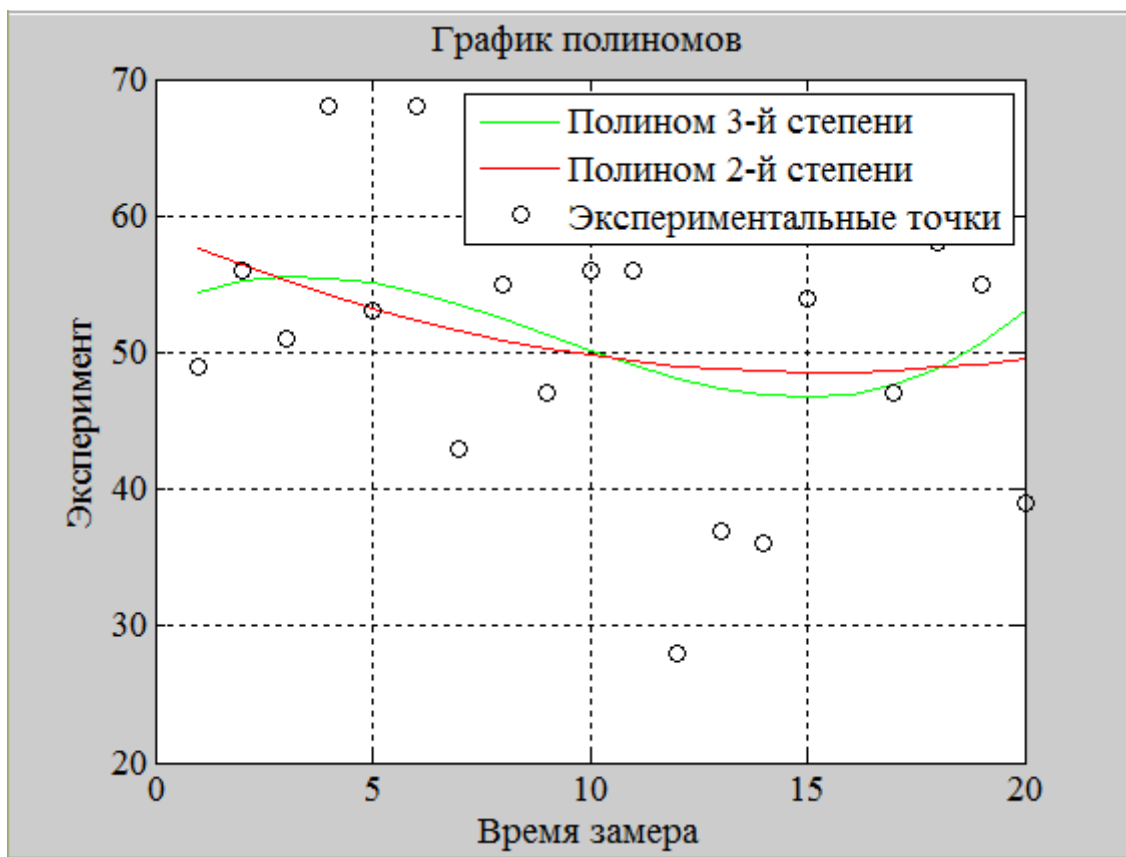


Рис. 4.17. Графики полиномов и экспериментальных точек

Исходные данные лучше описывает полином 3-й степени, так как он точнее передает характер разброса экспериментальных данных.



## 4.7. Определение кислотности раствора в MATLAB

*Задание:* Определить кислотность раствора

$$[\text{HA}]_0 = \frac{[\text{H}^+] - K_w}{K_a} + [\text{H}^+] - \frac{K_w}{[\text{H}^+]},$$

где  $[\text{H}^+]$  – концентрация ионов водорода. Значение pH – это отрицательный десятичный логарифм концентрации ионов водорода:  $\text{pH} = -\log([\text{H}^+])$ .

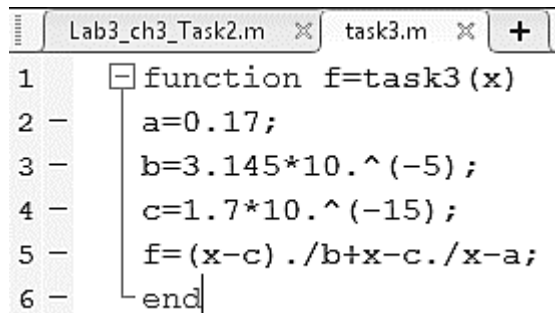
Написать функцию для нахождения pH, если известно:

$[\text{HA}]_0$ , моль <sup>-1</sup>	$K_a$	$K_w$
0,17	$3,145 \cdot 10^{-5}$	$1,7 \cdot 10^{-15}$

*Решение:* Запишем уравнение для определения концентрации ионов водорода ( $x$ ):

$$0,17 = \frac{x - 1,7 \cdot 10^{-15}}{3,145 \cdot 10^{-5}} + x - \frac{1,7 \cdot 10^{-15}}{x};$$

Далее создадим \*.m-файл с функцией для определения  $[\text{H}^+]$  раствора (рис. 4.18).



```
Lab3_ch3_Task2.m task3.m +
1 function f=task3(x)
2     a=0.17;
3     b=3.145*10.^(-5);
4     c=1.7*10.^(-15);
5     f=(x-c)./b+x-c./x-a;
6     end
```

Рис. 4.18. Функция для нахождения  $[\text{H}^+]$

Корни уравнения вида  $f(x)=0$  по заданному приближению позволяет найти встроенная функция  $fzero(\text{name}, x_0)$  (рис. 4.19),

где

- 1) *name* – имя М-функции, вычисляющей левую часть уравнения;
- 2)  $x_0$  – начальное приближение к корню или интервал его изоляции  $[a, b]$ ;

```
Lab3_ch3_Task2.m x +
1 - fplot('task3',[0 20])
2 - grid on
3 - set(0,'DefaultAxesFontSize',14,'DefaultAxesFontName','Times New Roman')
4 - title('График функции')
5 - xlabel('x')
6 - ylabel('y')
7 - x=fzero('task3',0.004)
8 - pH=-log(x)
```

Рис. 4.19. \*m-файл с командами

pH раствора равен 12,139 (рис. 4.20).

```
Command Window
>> x=fzero('task3',0.004)
pH=-log(x)

x =

    5.3463e-06

pH =

    12.1391

fx >>
```

Рис. 4.20. Командное окно с ответом

График функции представлен на рис. 4.21.

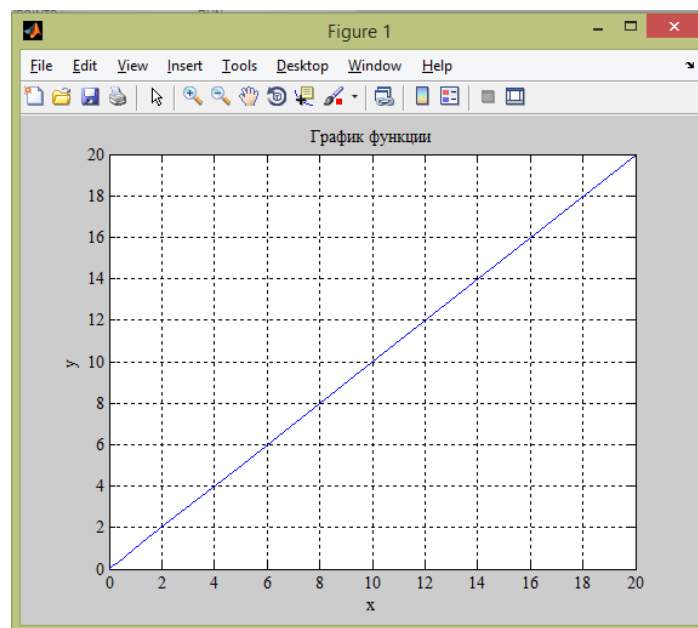


Рис. 4.21. График функции

#### **4.8. Требования к отчету по лабораторной работе**

Отчет по лабораторной работе выполняется на листах белой бумаги формата А4 в печатном виде.

Отчет формируется в следующем порядке:

1. Титульный лист.
2. Содержание.
3. Цель работы.
4. Задание.
5. Теоретическая часть. Краткие теоретические положения об основных функциональных возможностях пакета прикладных программ MatLAB.
6. Практическая (расчетная) часть.
7. Выводы по результатам выполнения работы.
8. Библиографический список.
9. Приложение.

При оформлении отчета используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страниц ставятся по центру в верхнем колонтитуле.

### **5. Лабораторная работа 5. Язык программирования R-Language для решения математических задач**

**Цель работы:** Используя визуальную среду для языка R, решить расчетно-графические задачи.

#### **5.1. Язык программирования R**

R – язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом в рамках проекта GNU. Язык создавался как аналогичный языку S, разработанному в Bell Labs, и является его альтернативной реализацией, хотя между языками есть существенные отличия, но в большинстве своем код на языке S работает в среде R [10].

R широко используется как статистическое программное обеспечение для анализа данных и фактически стал стандартом для статистических программ.

R доступен под лицензией GNU GPL. Распространяется в виде исходных кодов, а также откомпилированных приложений под ряд операционных систем: FreeBSD, Solaris и другие дистрибутивы Unix и Linux, Microsoft Windows, Mac OS X.

В R используется интерфейс командной строки, хотя доступны и несколько графических интерфейсов пользователя, например пакет R Commander, RKWard, RStudio, Weka, Rapid Miner, KNIME[en], а также средства интеграции в офисные пакеты .

В 2010 году R вошел в список победителей конкурса журнала Infoworld[en] в номинации на лучшее открытое программное обеспечение для разработки приложений.

R поддерживает широкий спектр статистических и численных методов и обладает хорошей расширяемостью. Пакеты представляют собой библиотеки для работы специфических функций или специальных областей применения. Несколько базовых пакетов присутствуют сразу после установки R на компьютер, без них система просто не работает (скажем, пакет, который так и называется base, или пакет grDevices, который управляет выводом графиков), а также «рекомендованные» пакеты (пакет для специализированного кластерного анализа cluster, пакет для анализа нелинейных моделей nlme и другие). Кроме того, можно поставить любой из почти восьми тысяч (на середину 2016 года) доступных на CRAN пакетов. При наличии доступа в Интернет это можно сделать прямо из R командой `install.packages()`.

Еще одной особенностью R являются графические возможности, заключающиеся в возможности создания качественной графики, которая может включать математические символы.

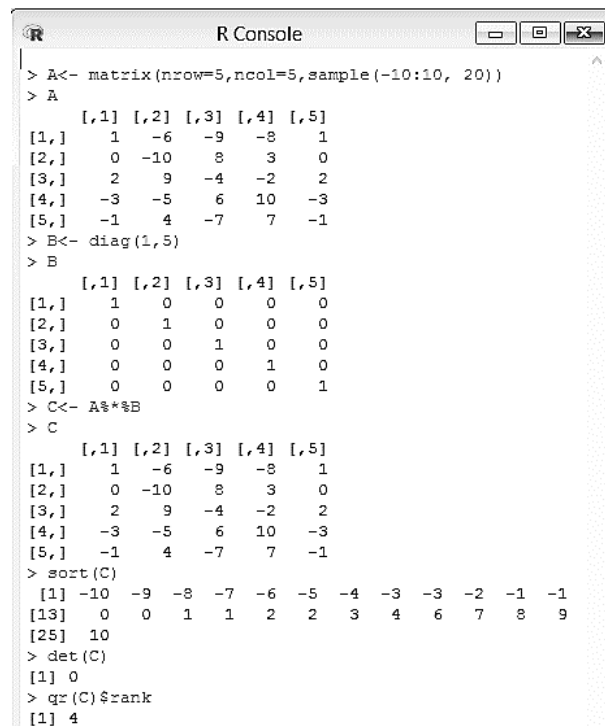
## 5.2. Работа с матрицами в визуальной среде для языка R. Сортировка массива

*Задание:* Сгенерируйте матрицу A (5x5), состоящую из случайных элементов отрезка  $[-10, 10]$ . Сгенерируйте матрицу B (5x5), все элементы диагонали которой – единицы. Получите матрицу C умножением матрицы A на B и отсортировав результат по возрастанию. Найдите детерминант и ранг матрицы C.

*Решение:* Генерируется матрица A (5x5), состоящая из случайных элементов отрезка  $[-10, 10]$ , с помощью функции *matrix*, где *nrow* и *ncol* – количество строк и столбцов матрицы соответственно, а *sample* – функция для нахождения случайных чисел на отрезке.

Далее генерируется матрица B (5x5), все элементы диагонали которой – единицы, с помощью функции *diag*. Умножением матрицы A на B (используется знак «*%\*%\**») получается матрица C.

Далее производится сортировка результата по возрастанию (*sort(C)*). Детерминант и ранг матрицы C определяется с помощью функций *det(C)* и *qr(C)\$rank* соответственно. На рис. 5.1 приведена консоль с результатами команд.



```
R Console
> A<- matrix(nrow=5,ncol=5,sample(-10:10, 20))
> A
      [,1] [,2] [,3] [,4] [,5]
[1,]  1  -6  -9  -8   1
[2,]  0 -10   8   3   0
[3,]  2   9  -4  -2   2
[4,] -3  -5   6  10  -3
[5,] -1   4  -7   7  -1
> B<- diag(1,5)
> B
      [,1] [,2] [,3] [,4] [,5]
[1,]  1   0   0   0   0
[2,]  0   1   0   0   0
[3,]  0   0   1   0   0
[4,]  0   0   0   1   0
[5,]  0   0   0   0   1
> C<- A*B
> C
      [,1] [,2] [,3] [,4] [,5]
[1,]  1  -6  -9  -8   1
[2,]  0 -10   8   3   0
[3,]  2   9  -4  -2   2
[4,] -3  -5   6  10  -3
[5,] -1   4  -7   7  -1
> sort(C)
 [1] -10  -9  -8  -7  -6  -5  -4  -3  -3  -2  -1  -1
[13]  0   0   1   1   2   2   3   4   6   7   8   9
[25]  10
> det(C)
[1] 0
> qr(C)$rank
[1] 4
```

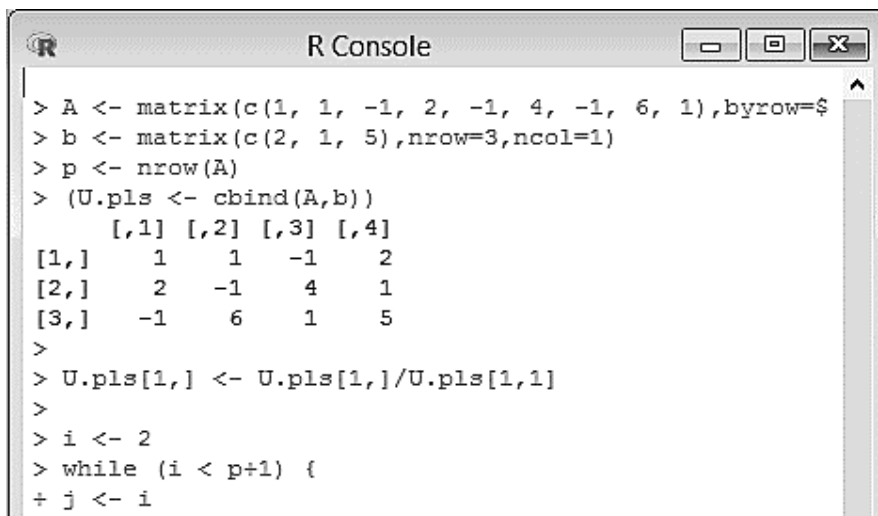
Рис. 5.1. Результаты выполнения команд в консоли

### 5.3. Решение системы уравнений методом Гаусса

*Задание:* Решите систему уравнений методом Гаусса:

$$\begin{cases} x + y - z = 2 \\ 2x - y + 4z = 1 \\ -x + 6y + z = 5 \end{cases}$$

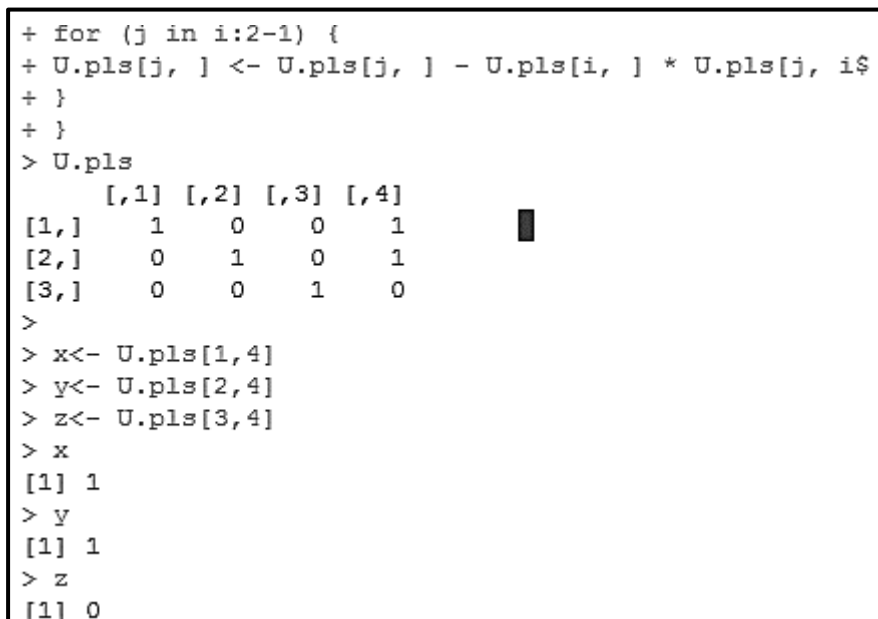
*Решение:* При помощи функции *matrix* задаются матрицы A и b. Далее вычисляется число строк матрицы A с помощью функции *nrow* и создается расширенная матрица *U.pls* (рис. 5.2).



```
> A <- matrix(c(1, 1, -1, 2, -1, 4, -1, 6, 1),byrow=$
> b <- matrix(c(2, 1, 5),nrow=3,ncol=1)
> p <- nrow(A)
> (U.pls <- cbind(A,b))
      [,1] [,2] [,3] [,4]
[1,]    1    1   -1    2
[2,]    2   -1    4    1
[3,]   -1    6    1    5
>
> U.pls[1,] <- U.pls[1,]/U.pls[1,1]
>
> i <- 2
> while (i < p+1) {
+ j <- i
```

Рис. 5.2. Расширенная матрица

С использованием циклов матрица приводится к единичному виду:



```
+ for (j in i:2-1) {
+ U.pls[j, ] <- U.pls[j, ] - U.pls[i, ] * U.pls[j, i$
+ }
+ }
> U.pls
      [,1] [,2] [,3] [,4]
[1,]    1    0    0    1
[2,]    0    1    0    1
[3,]    0    0    1    0
>
> x<- U.pls[1,4]
> y<- U.pls[2,4]
> z<- U.pls[3,4]
> x
[1] 1
> y
[1] 1
> z
[1] 0
```

Рис. 5.3. Измененная матрица и ответы в правом столбце

Решение системы уравнений:

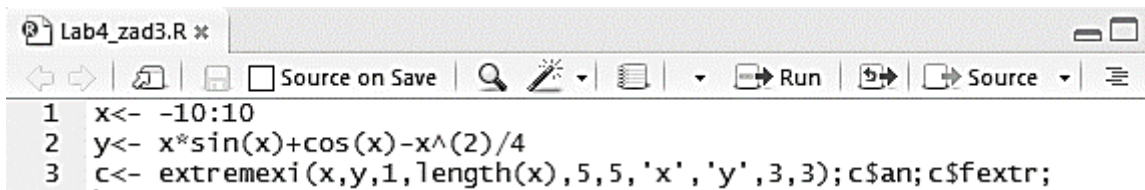
$$\begin{cases} x = 1 \\ y = 1 \\ z = 0 \end{cases}$$

$$\text{Проверка: } \begin{cases} 1+1-0=2 \\ 2 \cdot 1-1+4 \cdot 0=1 \\ -1+6 \cdot 1+0=5 \end{cases}$$

#### 5.4. Нахождение экстремумов функции

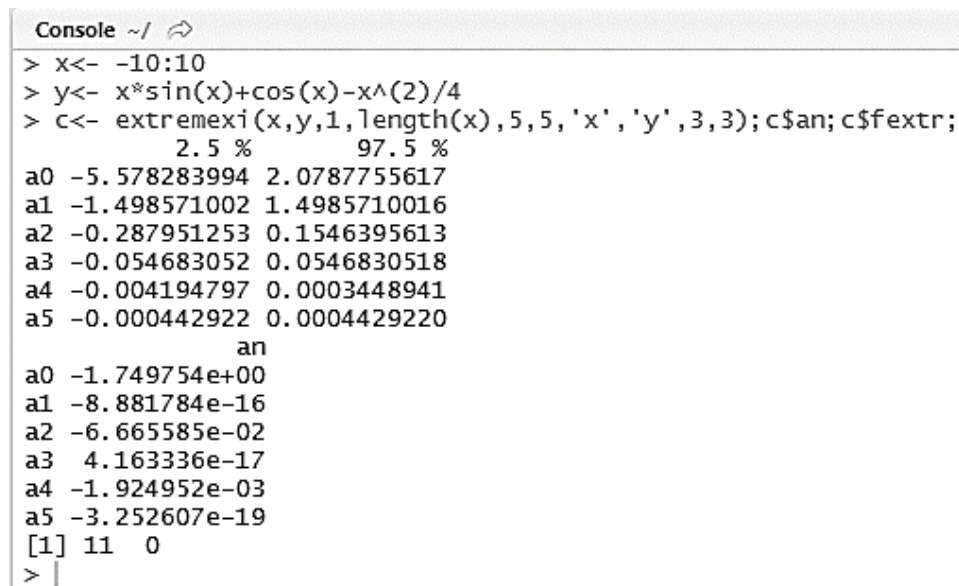
*Задание:* Найти все экстремумы функции при помощи оболочки для языка R:  $y = x \cdot \sin(x) + \cos(x) - \frac{x^2}{4}$ , на отрезке  $[-10, 10]$ .

*Решение:* Экстремумы функции определяются с помощью функции *extremexi* (рис. 5.4, 5.5).



```
1 x<- -10:10
2 y<- x*sin(x)+cos(x)-x^(2)/4
3 c<- extremexi(x,y,1,length(x),5,5,'x','y',3,3);c$an;c$fextr;
```

Рис. 5.4. Задаем функцию



```
Console ~/
> x<- -10:10
> y<- x*sin(x)+cos(x)-x^(2)/4
> c<- extremexi(x,y,1,length(x),5,5,'x','y',3,3);c$an;c$fextr;
      2.5 %      97.5 %
a0 -5.578283994 2.0787755617
a1 -1.498571002 1.4985710016
a2 -0.287951253 0.1546395613
a3 -0.054683052 0.0546830518
a4 -0.004194797 0.0003448941
a5 -0.000442922 0.0004429220
      an
a0 -1.749754e+00
a1 -8.881784e-16
a2 -6.665585e-02
a3 4.163336e-17
a4 -1.924952e-03
a5 -3.252607e-19
[1] 11 0
> |
```

Рис. 5.5. Все экстремумы функции

Следует обратить внимание, что использование оболочки для языка R позволяет существенно упростить работу, например, автоматическим построением графиков на вкладке *Plots* (рис. 5.6).

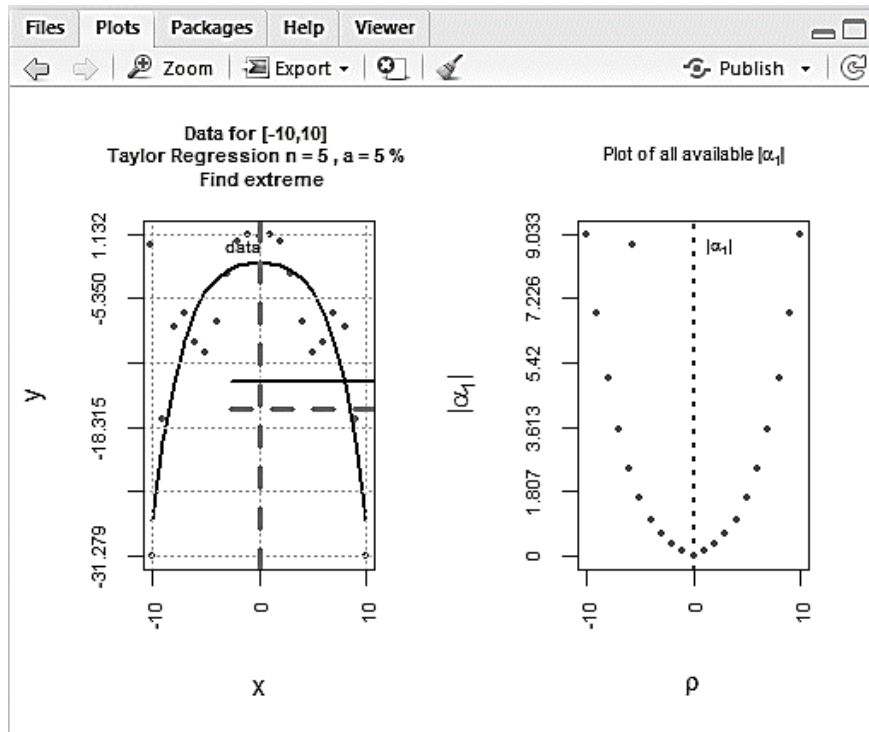


Рис. 5.6. Графики функции

## 5.5. Построение графика функции

**Задание:** Постройте график функции  $y=ax^2+bx+c$ , где:

a	b	c	$X \in$
-4	36	75	$[-2,10]$

**Решение:** Настраиваемый график заданной функции строится с помощью встроенной функции *plot* (рис. 5.7, 5.8).

```
x<- -2:10
y<- -4*x^2+36*x+75
plot(x,y,type="b")
title(main="График функции y=-4x^2+36x+75")
```

Рис. 5.7. Скрипт для задания



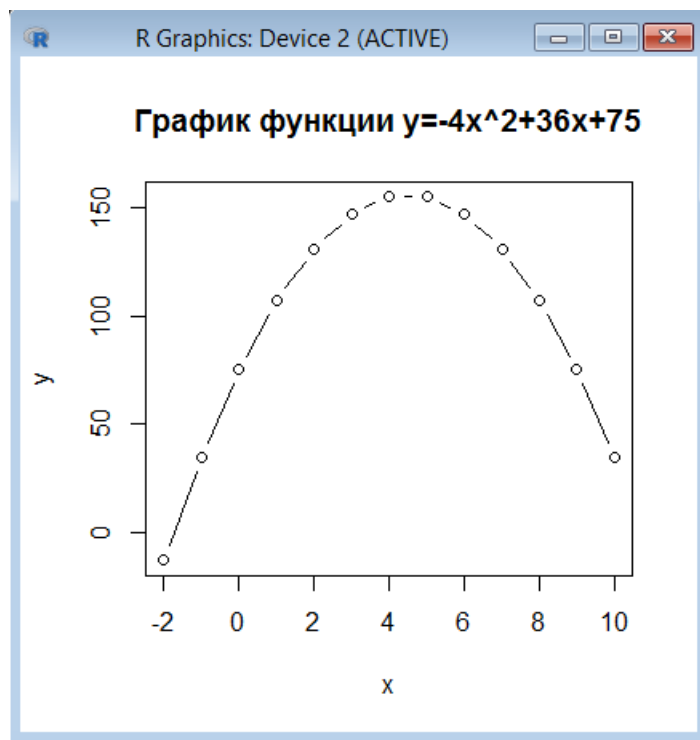


Рис. 5.8. График функции  $y$

С помощью множества параметров на данном графике можно не только подписать оси, но и построить сетку, ввести цветовые обозначения, добавить численные подписи к ключевым точкам и многое другое.

## 5.6. Решение системы нелинейных уравнений

*Задание:* Решите графически систему нелинейных уравнений:

$$\begin{cases} x^2 - y^2 = 1 \\ 2\sin(x-1) + y = 1 \end{cases};$$

*Решение:* Графическое решение системы нелинейных уравнений приведено на рис. 5.9, 5.10.

```
x<- -10:10
y<- 1-2*sin(x-1)
y2<- (x^2-1)^(1/2)
plot(x,y,type="b")
lines(x,y2,col="red")
title(main= ("Графическое решение системы нелинейных уравнений"))|
```

Рис. 5.9. Скрипт решения задания 5.6



Рис. 5.10. Графическое решение

С помощью графика можно определить два решения системы уравнений:

$$\begin{cases} x_1 = 1,2 \\ y_1 = 0,6 \end{cases}; \quad \begin{cases} x_2 = -1,9 \\ y_2 = 1,6 \end{cases}$$

## 5.7. Вычисление определенного интеграла

*Задание:* С использованием графической оболочки для языка R, вычислите определенный интеграл:

$$\int_1^{10} \frac{\ln(x^2 - 1)}{5 + e^x} dx;$$

*Решение:* Определенный интеграл в оболочке для R языка вычисляется с помощью функции *integral* (рис. 5.11).

```

Lab4_zad6.R * Lab4_zad7.R *
Source on Save
1 fun<- function(x) c(log10(x^2-1)/(5+exp(x)))
2 G<- integral(fun,1,10,arrayValued = TRUE, method = "simpson")
3 G

```

Рис. 5.11. Скрипт с решением

Ответ:  $x = -0,247$  (рис. 5.12).

```
Console ~/ ↵
> fun<- function(x) c(log10(x^2-1)/(5+exp(x)))
> G<- integral(fun,1,10,arrayvalued = TRUE, method = "simpson")
> G
[1] -0.2473692
```

Рис. 5.12. Решение определенного интеграла

## 5.8. Работа с условными операторами

*Задание:* Создайте условие на основе результатов предыдущего задания, в котором при  $x > y$  и  $x < z$  ( $y=0, z=10$ ) –  $x$  будет округляться до десятых и добавляться в конец вектора, сгенерированного из 10 случайных чисел от 0 до 10.

*Решение:* Создается условие на основе результатов предыдущего задания, в котором при  $x > y$  и  $x < z$  ( $y=0, z=10$ ) –  $x$  будет округляться до десятых и добавляться в конец вектора, сгенерированного из 10 случайных чисел от 0 до 10. Используется оператор условия *IF*, следует обратить внимание, что синтаксис языка R соответствует так называемым C (си)-подобным языкам программирования, поэтому каждое логическое условие берется в скобки и соединяется при помощи  $\&\&$  – логического «И» или  $\|\|$  – логического «ИЛИ».

```
Lab4_zad7.R x
Source on Save
1 x<- -0.2473692
2 z<- 10
3 y<- 0
4 if ((z>x)&&(y<x)){
5   k<- round(x, digits = 1);t<- sample(0:10, 9);
6   v<- c(t,k);view(k)
7 } else {}
8 print("x<0 or x>10")
9 }
```

Рис. 5.13. Скрипт с условием

Сгенерировать случайные числа можно с помощью функции *sample*, с указанием интервала и количества чисел. С помощью функции *c* создается новый вектор.

Результатом является строка « $x < 0$  or  $x > 10$ », так как  $-0,247 < 0$  (рис. 5.14).

```

Console ~/ ↻
> x<- -0.2473692
> z<- 10
> y<- 0
> if ((z>x)&&(y<x)){
+   k<- round(x, digits = 1);t<- sample(0:10, 9);
+   v<- c(t,k);view(k)
+ } else {
+   print("x<0 or x>10")
+ }
[1] "x<0 or x>10"
>

```

Рис. 5.14. Вывод ответа в консоли

### 5.9. Работа с индексированными матрицами

*Задание:* Создайте матрицу А размерностью 4x4, элементы которой – числа по возрастанию от 1 до 16. Найдите сумму элементов по строкам и столбцам и занесите эти результаты в одну индексированную матрицу (индексами сделать буквы алфавита).

*Решение:* Создается матрица А размерностью 4x4, элементы которой – числа по возрастанию от 1 до 16. Находится сумма элементов по строкам и столбцам и заносятся эти результаты в одну индексированную матрицу М (рис. 5.15).

```

A<- matrix(1:16, nrow=4, ncol=4, byrow=TRUE)
sumacol<- colSums(A, na.rm=FALSE, dims=1)
sumastr<- rowSums(A, na.rm=FALSE, dims=1)
M<- matrix(c(sumacol,sumastr), ncol=4, nrow=2, byrow=TRUE)
rownames(M)<- c("А", "Б")
colnames(M)<- c("В", "Г", "Д", "Е")
М
А

```

Рис. 5.15. Скрипт задания

Результат приведен на рис. 5.16. В строке А перечислены суммы элементов по столбцам, в строке Б – по строкам.

```

> М
  В Г Д Е
А 28 32 36 40
Б 10 26 42 58
> А
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16

```

Рис. 5.16. Матрица М и матрица А

## 5.10. Экспорт данных из визуальной среды для языка R

*Задание:* Полученную в предыдущем задании индексированную матрицу экспортируйте в документ формата `.csv`.

*Решение:* Полученная в пункте 5.9 индексированная матрица экспортируется в документ формата `.csv` с помощью функции `write.csv`. Документ назван `matr.csv`. Он показан на рис. 5.18.

```
> M
  В Г Д Е
А 28 32 36 40
Б 10 26 42 58
> write.csv(M, file="matr.csv")
```

Рис. 5.17. Экспорт матрицы в документ формата `.csv`

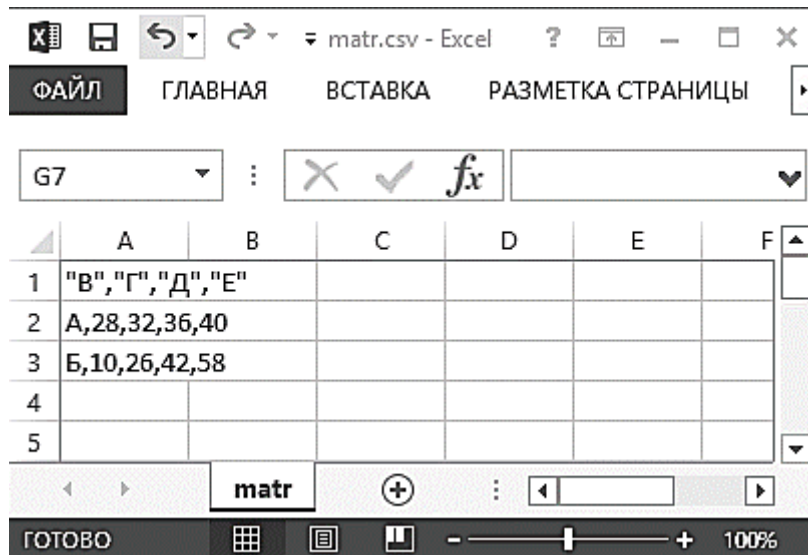


Рис. 5.18. Документ `matr.csv`

## 5.11. Решение системы нелинейных уравнений с использованием языка R

*Задание:* Найдите численное решение системы нелинейных уравнений с использованием языка R.

$$\begin{cases} x^2 - y^2 = 1 \\ 2 \sin(x-1) + y = 1 \end{cases}$$

*Решение:* Численное решение системы нелинейных уравнений с использованием языка R приведено на рис. 5.19, 5.20.

```

1 dslnex<- function(x){
2   y<- numeric(2)
3   y[1]<- (2-cos(x[1]))/2
4   y[2]<- ((x[1])^2-3)^(1/2)
5   y
6 }
7 xstart<- c(3.2,0.5)
8 fstart<- dslnex(xstart)
9 nleqslv(xstart, dslnex, control=list(trace=1, btol=0.01, delta=-2))

```

Рис. 5.19. Скрипт решение системы нелинейных уравнений

На 7 и 8 строке задается начальное приближение, а на строке 9 вызывается встроенная функция для решения методом Ньютона с заданной точностью. Каждая встроенная функция имеет множество параметров, полный список всех параметров можно посмотреть в разделе справка (Help).

```

> nleqslv(xstart, dslnex, control=list(trace=1, btol=0.01, delta=-2))
Algorithm parameters
-----
Method: Broyden  Global strategy: double dogleg (initial trust region = -2)
Maximum stepsize = 1.79769e+308
Scaling: fixed
ftol = 1e-08 xtol = 1e-08 btol = 0.01 cndtol = 1e-12

Iteration report
-----
Iter      Jac      Lambda      Eta      Dlt0      Dltn      Fnorm      Largest |f|
0
1  Ns
$x
[1] 3.2 0.5

$fvec
[1] 1.499147 2.690725

$termcd
[1] 6

$message
[1] "Jacobian is singular (1/condition=0.0e+000) (see allowsingular option)"

$scalex
[1] 1 1

$nfcnt
[1] 0

$njcnt
[1] 1

$iter
[1] 1

```

Рис. 5.20. Результат численного решения системы методом Ньютона

## 6. Обзор специализированных программных комплексов и языков программирования, используемых в инженерной практике

В настоящее время, в век информационных технологий, большинство узкоспециализированных программных продуктов либо устарели, либо вошли в состав программных продуктов с широкими функциональными возможностями. Разработчики пакетов прикладных программ стали развивать свои продукты путем расширения, т.е. включая в них различные смежные (по постановке или методу решения) функциональные возможности.

Далее представлен краткий обзор некоторых специализированных и универсальных программных комплексов и языков программирования для решения математических задач.

### 6.1. Система компьютерной алгебры Mathematica [11]

**Mathematica** – система компьютерной алгебры, используемая во многих научных, инженерных, математических и компьютерных областях. Изначально система была придумана Стивеном Вольфрамом, в настоящее время разрабатывается компанией Wolfram Research.

*Возможности:*

#### а) Аналитические преобразования

- Решение систем полиномиальных и тригонометрических уравнений и неравенств, а также трансцендентных уравнений, сводящихся к ним.
- Решение рекуррентных уравнений.
- Упрощение выражения.
- Нахождение пределов.
- Интегрирование и дифференцирование функций.
- Нахождение конечных и бесконечных сумм и произведений.
- Решение дифференциальных уравнений и уравнений в частных производных (метод вариации параметров, метод стрельбы, метод конечных элементов и др.).
- Преобразования Фурье и Лапласа, а также Z-преобразование.

- Преобразование функции в ряд Тейлора, операции с рядами Тейлора: сложение, умножение, композиция, получение обратной функции и т. д.

- Вейвлет-анализ.

#### **б) Численные расчеты**

- Вычисление значений функций, в том числе специальных, с произвольной точностью.

- Решение систем уравнений.

- Нахождение пределов.

- Интегрирование и дифференцирование.

- Нахождение сумм и произведений.

- Решение дифференциальных уравнений и уравнений в частных производных.

- Полиномиальная интерполяция функции от произвольного числа аргументов по набору известных значений.

- Преобразования Фурье и Лапласа, а также Z-преобразование.

- Расчет вероятностей (Вероятность и статистика используются для моделирования неопределенности. Язык Вольфрам использует символные распределения и процессы в качестве моделей для случайных величин и случайных процессов. Модели автоматически вычисляются на основе данных или аналитически строятся из библиотеки встроенных распределений и процессов).

#### **в) Теория чисел**

- Определение простого числа по его порядковому номеру, определение количества простых чисел, не превосходящих данное.

- Дискретное преобразование Фурье.

- Разложение числа на простые множители, нахождение НОД и НОК.

#### **г) Линейная алгебра**

- Операции с матрицами: сложение, умножение, нахождение обратной матрицы, умножение на вектор, вычисление экспоненты, получение определителя.

- Поиск собственных значений и собственных векторов.



#### **д) Графика и звук**

- Построение графиков функций, в том числе параметрических кривых и поверхностей.
- Построение геометрических фигур: ломаных, кругов, прямоугольников, и т. д.
- Воспроизведение звука, график которого задается аналитической функцией или набором точек.
- Импорт и экспорт графики во многих растровых и векторных форматах, а также звука.

#### **е) Построение и манипулирование графами**

В системе Mathematica имеются возможности, позволяющие находить разрезы и разбиения графов, а также автоматически визуализировать группы вершин и связывать грани. Дополнительные распределения вероятностей на графах, туры по графам и компоновки графов расширяют богатую коллекцию функциональных возможностей. Также имеются улучшения производительности различных функций.

- Обширная поддержка задач о потоках в сетях, включая нахождение потоков максимальной или минимальной стоимости.
- Новые функции нахождения разреза графа и разрезающих вершин графа, а также вершинной и реберной связности графа.
- Нахождение сбалансированных разбиений графов, имеющих наименьшее число ребер, соединяющих подмножества.
- Новые и оптимизированные распределения вероятностей на графах.
- Возможность задания свойств случайных графов и автоматического нахождения вероятностей и математических ожиданий.
- Высоко оптимизированное нахождение Эйлеровых и Гамильтоновых цепей.
- Поддержка нахождения решения задачи китайского почтальона.
- Проверки планарности графов.
- Улучшенная производительность укладки графов и их рендеринга.
- Несколько новых алгоритмов укладки графов, среди которых планарная, спектральная и укладка в виде шаров.
- Новые эмпирические образцы графов, а также новые теоретические графы и новые свойства графов.
- Обширное улучшение производительности во всех функциях

для работы с графами и сетями.

## 6.2. Язык программирования Python

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

**SciPy** – библиотека для языка программирования Python с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчетов.

*Возможности:*

- поиск минимумов и максимумов функций;
- вычисление интегралов функций;
- поддержка специальных функций;
- обработка сигналов;
- обработка изображений;
- работа с генетическими алгоритмами;
- решение обыкновенных дифференциальных уравнений;
- и др.

Целевая аудитория – пользователи продуктов MATLAB и Scilab.

Для визуализации результатов расчетов часто применяется библиотека *Matplotlib*, являющаяся аналогом средств вывода графики MATLAB.

Библиотека SciPy распространяется по условиям лицензии BSD (т.е. свободное некоммерческое использование).

### 6.3. Язык программирования Julia

Julia – высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков (например, MATLAB и Octave), однако имеет некоторые существенные отличия. Julia написана на Си, C++ и Scheme. В стандартный комплект входит JIT-компилятор, благодаря чему, по утверждению авторов языка, приложения, полностью написанные на языке, практически не уступают в производительности приложениям, написанным на статически компилируемых языках вроде Си или C++. Также язык имеет встроенную поддержку большого числа команд для распределенных вычислений.

Основной задачей при создании была разработка универсального языка, способного работать с большим объемом вычислений и при этом гарантировать максимальную производительность. Поскольку большой объем вычислений выполняется именно в облачных средах, то в языке была сразу реализована поддержка облачных вычислений и параллельного программирования.

В языке была изначально реализована модель построения больших параллельных приложений, основанная на глобальном распределенном адресном пространстве. Такая модель подразумевает возможность производить операции (в том числе и их передачу между машинами) со ссылкой на объект, расположенный на другой машине, также участвующей в вычислениях. Этот механизм позволяет отслеживать, какие вычисления на каких системах выполняются, а также подключать к производимым вычислениям новые машины.

### 6.4. Язык программирования Fortran

**Фортра́н (Fortran)** – первый язык программирования высокого уровня, получивший практическое применение, имеющий транслятор и испытавший дальнейшее развитие. Создан в период с 1954 по 1957 год группой программистов под руководством Джона Бэкуса в корпорации

IBM (Язык Планкалкюль, претендующий на пальму первенства, был изобретен еще в 1945 году, но не был реализован вплоть до 2000 года). Название Fortran является сокращением от **FOR**mula **TRAN**slator (переводчик формул). Фортран широко используется в первую очередь для научных и инженерных вычислений. Одно из преимуществ современного Фортрана – большое количество написанных на нем программ и библиотек подпрограмм.

Имеется большое количество написанных на Фортране (в большей части на старых версиях языка) различных математических библиотек для матричной алгебры и решения систем линейных уравнений, библиотеки для решения дифференциальных уравнений, интегральных уравнений и их систем, аппроксимации функций, специальных функций, быстрых преобразований Фурье, математической статистики и других математических дисциплин. Эти библиотеки поставляются, как правило, с компилятором. Ряд таких пакетов создавался на протяжении десятилетий и популярен в научной среде по настоящее время, например, IMSL (англ.).

Большинство таких библиотек является фактически достоянием человечества: они доступны в исходных кодах, хорошо документированы, отлажены и весьма эффективны.

Современный Фортран (Fortran 95 и Fortran 2003) приобрел черты, необходимые для эффективного программирования, для новых вычислительных архитектур; позволяет применять современные технологии программирования, в частности, обобщенное и модульное программирование, ООП, сохраняя, при этом, преемственность с более ранними версиями. Одна из наиболее главных концепций развития современного Фортрана – средства поддержки параллельности и векторные операции.

### **6.5. Пакет прикладных программ STATISTICA [12]**

Пакет прикладных программ (ППП) STATISTICA – это универсальная интегрированная система, предназначенная для статистического анализа и обработки данных. Содержит многофункциональную систему для работы с данными, широкий набор статистических модулей, в которых собраны группы логически связанных между собой статистических проце-

дур, специальный инструментарий для подготовки отчетов, мощную графическую систему для визуализации данных, систему обмена данными с другими Windows-приложениями.

С помощью реализованных в системе STATISTICA языков программирования (SQL, STATISTICA BASIC), снабженных специальными средствами поддержки, легко создаются законченные пользовательские решения и встраиваются в различные другие приложения или вычислительные среды.

Система STATISTICA производится фирмой StatSoft Inc. (США), основанной в 1984 г. в городе Тулса (США). Первые программные продукты фирмы (PsyhoStat-2,3) были предназначены для обработки социологических данных. В 1985 г. StatSoft выпускает первую систему статистического анализа для компьютеров Apple Macintosh (StatFast) и статистический пакет для IBM PC (STATS+). В 1986 г. начинается работа по созданию интегрированных статистических пакетов комплексной обработки данных [15].

В 1991 г. выходит первая версия системы STATISTICA/DOS. Эта программа представляла собой новое направление развития статистического программного обеспечения, так как в ней реализован графически ориентированный подход к анализу данных. Программа обладала рядом существенных преимуществ перед другими статистическими программами. Например, за счет оптимизации удалось добиться повышения скорости обработки более чем в 10 раз, программа могла анализировать фактически неограниченный объем данных. В 1992 г. вышла версия STATISTICA для Macintosh. В 1994 г. выходит версия STATISTICA 4.5 для Windows, которая сразу же занимает лидирующее положение среди статистических пакетов. В результате сравнительного тестирования с профессиональными системами BMDP 1.0, SPSS 6.1, STATGRAPHICS 1.0, SYSTAT 5.01 она получает первое место в некоторых ведущих научных и компьютерных изданиях.

В конце 1995 г. вышла версия STATISTICA 5.0 с более удобным пользовательским интерфейсом и полной совместимостью с Windows 95. В этой версии реализованы новые мощные возможности численного и графического анализа данных. STATISTICA 5.0 полностью удовлетворяет ос-

новным стандартам среды Windows. Это стандарты пользовательского интерфейса (MDI); использование технологий DDE (динамического обмена данными из других приложений); OLE (связывания и внедрения объектов, поддержка основных операций с буфером обмена) и др. В отличие от предыдущих версий в STATISTICA 5.0 включен внутренний язык программирования STATISTICA BASIC, который позволит пользователю расширять возможности системы. Например, нарастить систему по своему усмотрению, добавив собственную панель инструментов с тем или иным методом статистического анализа.

Система STATISTICA имеет более полумиллиона зарегистрированных пользователей во всем мире. Пользователями системы являются крупнейшие университеты, исследовательские центры, компании, банки всего мира, государственные учреждения. Имеются версии системы на немецком, французском, японском, испанском, польском и других языках. В 1999 г. состоялся выпуск русской версии STATISTICA 5.1. Корпорацией Софтлайн во главе с В.П. Боровиковым издано большое число книг с подробным описанием системы STATISTICA 5.0. Технология работы с основными процедурами проиллюстрирована на большом количестве примеров.

Появление операционной системы Windows XP привело к необходимости структурных изменений программы STATISTICA и созданию новой версии STATISTICA 6.0. В ней существенно изменены структуры интерфейса, диалоговых окон. Некоторые модули исключены, так как в новой версии за счет расширения возможностей они потеряли актуальность. Русифицированная версия называется STATISTICA 6.1.

Не будет ошибочным утверждение, что из существующих статистических пакетов (SAS, BMDP, STADIA, OLIMP, STATGRAPHICS, SPSS и др.) ППП STATISTICA является наиболее известным в России. Известность пакет приобрел благодаря деятельности компании StatSoft (RUSSA), являющейся дилером компании производителя StatSoft Inc(USA). На сайте [www.statsoft.ru](http://www.statsoft.ru) можно найти всю необходимую информацию о пакете STATISTICA.

## Заключение

Пакеты прикладных программ являются мощным инструментом автоматизации решаемых пользователем задач, практически полностью освобождая его от необходимости знать, как выполняет компьютер те или иные функции и процедуры по обработке информации.

В настоящее время имеется широкий спектр ППП, различающихся по своим функциональным возможностям и способам реализации.

Данный лабораторный практикум содержит пять работ, направленных на ознакомление с основным функционалом и получение практических навыков работы с универсальными программными средствами решения математических задач, такими как табличный процессор MS Excel, вычислительная среда MathCAD, пакет прикладных программ MATLAB, язык программирования R.

Изучив основные функциональные возможности программных средств, представленных в пособии, пользователь получает возможность решения широкого круга задач, связанных с математическим моделированием, статистической обработкой данных, автоматизированными численными и символьными вычислениями и др.

## Библиографический список

1. Гартман Т.Н., Клушин Д.В. Основы компьютерного моделирования химико-технологических процессов: учебное пособие для вузов. М. : Академкнига, 2006. 416 с.
2. Табличный редактор Microsoft Excel [Электронный ресурс]. Режим доступа: [http://www.plam.ru/compinet/office\\_2007\\_multimediinyi\\_kurs/p4.php](http://www.plam.ru/compinet/office_2007_multimediinyi_kurs/p4.php) – (дата обращения: 01.11.2017)
3. Построение графиков Mathcad [Электронный ресурс]. Режим доступа: <http://eco.sutd.ru/mathcad/docs/mathcad/graphy.htm> (дата обращения: 03.11.2017)
4. Mathcad. Справочник по высшей математике. Математический анализ [Электронный ресурс]. Режим доступа: <http://old.exponenta.ru/soft/Mathcad/learn/ma/ma.asp> (дата обращения: 03.11.2017)
5. Руководство пользователя Mathcad [Электронный ресурс]. Режим доступа: [http://old.exponenta.ru/soft/Mathcad/UsersGuide/chapter13/13\\_6.asp](http://old.exponenta.ru/soft/Mathcad/UsersGuide/chapter13/13_6.asp) (дата обращения 14.05.2017)
6. Примеры аппроксимации функций [Электронный ресурс]. Режим доступа: <http://eco.sutd.ru/mathcad/docs/fit/examples.htm> (дата обращения 14.05.2017)
7. Руководство пользователя Mathcad [Электронный ресурс]. Режим доступа: [http://old.exponenta.ru/soft/Mathcad/UsersGuide/chapter15/15\\_2.asp](http://old.exponenta.ru/soft/Mathcad/UsersGuide/chapter15/15_2.asp) (дата обращения 14.05.2017)
8. Матричный метод решения систем уравнений [Электронный ресурс]. Режим доступа: [http://www.webmath.ru/poleznoe/formules\\_5\\_3.php](http://www.webmath.ru/poleznoe/formules_5_3.php) (дата обращения 12.05.2017)
9. Экстремумы функций в Mathcad [Электронный ресурс]. Режим доступа: <https://allmathcad.com/ru/ekstremumy-funksij.html> (дата обращения 12.05.2017)
10. Распознавание. «Язык программирования R» [Электронный ресурс]. Режим доступа: <http://www.machinelearning.ru/wiki/index.php?title=R> (Дата обращения 20.05.2017)
11. Wolfram mathematica, официальный сайт [Электронный ресурс]. Режим доступа: <http://www.wolfram.com/mathematica/> (дата обращения: 30.11.2017)
12. ППП Statistica [Электронный ресурс]. Режим доступа: <http://statlab.kubsu.ru/node/3> (дата обращения 12.05.2017)