

МАТЕМАТИЧЕСКОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Учебная программа дисциплины

Учебное пособие

➤ **Лабораторный практикум**

Методические указания по самостоятельной работе

Банк тестовых заданий в системе UniTest



УДК 519.85
ББК 22.18
Я49

Электронный учебно-методический комплекс по дисциплине «Математическое программное обеспечение» подготовлен в рамках инновационной образовательной программы «Инновационно-образовательный центр технологий поддержки жизненного цикла и качества продукции», реализованной в ФГОУ ВПО СФУ в 2007 г.

Рецензенты:

Красноярский краевой фонд науки;
Экспертная комиссия СФУ по подготовке учебно-методических комплексов дисциплин

Якунина, С. В.

Я49 Математическое программное обеспечение. Версия 1.0 [Электронный ресурс] : лаб. практикум / С. В. Якунина. – Электрон. дан. (1 Мб). – Красноярск : ИПК СФУ, 2008. – (Математическое программное обеспечение : УМКД № 182-2007 / рук. творч. коллектива С. В. Якунина). – 1 электрон. опт. диск (DVD). – Систем. требования : *Intel Pentium* (или аналогичный процессор других производителей) 1 ГГц ; 512 Мб оперативной памяти ; 1 Мб свободного дискового пространства ; привод *DVD* ; операционная система *Microsoft Windows 2000 SP 4 / XP SP 2 / Vista* (32 бит) ; *Adobe Reader 7.0* (или аналогичный продукт для чтения файлов формата *pdf*).

ISBN 978-5-7638-1505-4 (комплекса)

Номер гос. регистрации в ФГУП НТЦ «Информрегистр» 0320802758 от 23.12.2008 г. (комплекса)

Настоящее издание является частью электронного учебно-методического комплекса по дисциплине «Математическое программное обеспечение», включающего учебную программу, учебное пособие, методические указания по самостоятельной работе, контрольно-измерительные материалы «Математическое программное обеспечение. Банк тестовых заданий», наглядное пособие «Математическое программное обеспечение. Презентационные материалы».

В лабораторном практикуме изложен теоретический материал по циклу лабораторных работ, необходимый для освоения дисциплины. Приведено описание девятнадцати лабораторных работ, выполняемых в математических пакетах MathCAD и MatLab, дан порядок их выполнения и варианты заданий.

Предназначен для студентов направления подготовки бакалавров 230100.62 «Информатика и вычислительная техника» укрупненной группы 230000 «Вычислительная техника и информационные технологии».

© Сибирский федеральный университет, 2008

Рекомендовано Инновационно-методическим управлением СФУ
в качестве учебного пособия

Редактор Л. Х. Бочкарева

Разработка и оформление электронного образовательного ресурса: Центр технологий электронного обучения информационно-аналитического департамента СФУ; лаборатория по разработке мультимедийных электронных образовательных ресурсов при КрЦНИТ

Содержимое ресурса охраняется законом об авторском праве. Несанкционированное копирование и использование данного продукта запрещается. Встречающиеся названия программного обеспечения, изделий, устройств или систем могут являться зарегистрированными товарными знаками тех или иных фирм.

Подп. к использованию 01.09.2008

Объем 1 Мб

Красноярск: СФУ, 660041, Красноярск, пр. Свободный, 79

Оглавление

Введение	6
Лабораторная работа 1. Назначение, принципы функционирования и использования MATLAB	7
Теоретическое сведение	7
Лабораторная работа 2. Простейшие вычисления в MATLAB.....	10
Порядок работы.....	10
Задания.....	11
Лабораторная работа 3. Работа с массивами.векторы-столбцы и векторы-строки	12
Порядок работы.....	12
Задания.....	16
Лабораторная работа 4. Работа с массивами	18
Порядок работы.....	18
Задания.....	20
Лабораторная работа 5. Двумерные массивы и матрицы.....	22
Порядок работы.....	22
Задания.....	23
Лабораторная работа 6. Блочные матрицы.....	25
Порядок работы.....	25
Задания.....	27
Лабораторная работа 7. Визуализация матрицы поэлементные операции над ними	29
Порядок работы.....	29
Задания.....	30
Лабораторная работа 8. Графики, диаграммы и гистограммы	32
Порядок работы.....	32
Задания.....	35



Лабораторная работа 9. М-файлы. Численные методы и программирование	38
Порядок работы.....	38
Задания.....	42
Лабораторная работа 10.	
Программирование.операторы цикла.....	43
Порядок работы.....	43
Задания.....	46
Лабораторная работа 11. Программирование. Операторы ветвления. Исключительные ситуации	47
Порядок работы.....	47
Лабораторная работа 12. Программирование прерывания цикла, исключительные ситуации ..	54
Порядок работы.....	54
Задание.....	56
Лабораторная работа 13. Основные возможности пакета SIMULINK. Интерфейс браузера библиотек. Интерфейс окна моделей SIMULINK. Создание модели. Основные приемы подготовки и редактирования модели	57
Порядок работы.....	57
Задания.....	63
Лабораторная работа 14. Назначение, принципы функционирования и использования MATHCAD .	64
Теоретическое сведение	64
Лабораторная работа 15. Работа с установками MATHCAD.....	66
Порядок работы.....	66
Лабораторная работа 16. Переменные и основные математические операции в MATHCAD	70
Задания.....	70

Лабораторная работа 17. Построение двумерных графиков в MATHCAD	73
Задания.....	73
Лабораторная работа 18. Задание собственных функции в MATHCAD. Программирование в MATHCAD.....	74
Задания.....	74
Лабораторная работа 19. Решение систем уравнений и неравенств в MATHCAD	75
Задания.....	75
Библиографический список	76

ВВЕДЕНИЕ

В настоящее время существуют различные математические пакеты, обладающие мощными символьными процессорами, например, такие как Maple и Mathematica. Для выполнения аналитических выкладок применительно к теории электрических машин и с точки зрения систем автоматизированного проектирования (САПР) вполне достаточно возможностей таких программ, как MathCAD и MatLab, которые используют ядро символьного процессора Maple.

В процессе изучения дисциплины студенты на практике могут освоить работу в данных математических пакетах MathCAD и MatLab, приобрести опыт, который в дальнейшем может понадобиться в процессе учебы там, где присутствуют любые численные расчеты и моделирование технических и физических систем, а также требуется выполнение научных и инженерных расчетов при работе с массивами данных. Все это, несомненно, внесет свой вклад в формирование из студентов высококвалифицированных специалистов по направлению 23100.62 «Информатика и вычислительная техника».

При помощи пакетов MathCAD и MatLab можно решить значительный спектр задач, например: проведение математических исследований, требующих вычислений и аналитических выкладок; разработка и анализ алгоритмов; математическое моделирование, компьютерный эксперимент; анализ и обработка данных; визуализация, научная и инженерная графика; разработка графических и расчетных приложений.

Основа MatLab – это работа с матрицами, поэтому даже вычисления со скалярами реализуются как операции с матрицами размера 1×1 . Матричные команды написаны особенно тщательно, поэтому везде, где это возможно, целесообразно пользоваться матричными (векторными) операциями, что ускоряет вычисления и предупреждает возможные ошибки. Матрицам, их свойствам и операциям с ними посвящено огромное количество литературы. MatLab в первую очередь предназначен для численного исследования, а потому оперирует в основном с числовой информацией.

В данном пособии описываются лабораторные работы, позволяющие детально изучить данную дисциплину. И в дальнейшем полученные знания можно будет применять для математических расчетов при изучении других дисциплин.

ЛАБОРАТОРНАЯ РАБОТА 1

НАЗНАЧЕНИЕ, ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ И ИСПОЛЬЗОВАНИЯ MATLAB

Цель работы: посмотреть на слайдах внешний вид MatLab, принципы функционирования и использования MatLab.

Теоретическое сведение

Система MatLab разработана специалистами компании MathWork Inc. Само название произошло от сокращения MATrix LABoratory – матричная лаборатория.

Эта система предназначена осуществлять любые численные расчеты моделирования технических и физических систем, а также выполнять научные и инженерные расчеты, работая при этом с массивами данных. Данный математический пакет можно применять также в системах автоматизированного проектирования (САПР), потому что из всех математических пакетов MatLab ближе всех к идеологии САПР.

В MatLab есть модульность и существуют модули для исследования различных объектов, а также системы визуального моделирования. В ней содержатся различные способы моделирования в Simulink, что очень удобно для анализа больших систем, которые состоят из отдельных блоков. MatLab может состыковываться с реальным оборудованием, имеется также возможность подключать какие-либо программы, написанные в Си. Помимо этого существует возможность данным модулям или библиотекам работать в режиме реального времени. В основе MatLab лежит громадная библиотека научно-технических расчетов. Там очень много различных численных и других методов, которые используются при проектировании САПР. И внутри имеется символьный процессор из Maple. Самая сильная сторона Maple это символьные вычисления и она практически полностью присутствует в MatLab. Следовательно, выбирая MatLab, мы одновременно выбираем самые сильные стороны Maple. На сегодняшний день MatLab самая распространенная математическая программа для технических применений.

Система поддерживает выполнение операций с векторами, матрицами и массивами данных, поддерживает работу с алгебраическими полиномами, осуществляет решение нелинейных уравнений и задач оптимизации, численное интегрирование дифференциальных и разностных уравнений, построение различных трафиков, трехмерных поверхностей и линий уровня.

Основной объект системы MATLAB – числовой массив или другими словами – матрица, в которой допускается применение комплексных элементов. Использование матриц не требует явного указания их размеров.

Система MATLAB обеспечивает выполнение операций с векторами и матрицами даже в режиме непосредственных вычислений. Ею можно пользоваться как мощнейшим калькулятором, в котором наряду с обычными арифметическими и алгебраическими действиями могут использоваться такие сложные операции, как обращение матрицы, вычисление ее собственных значений и векторов, решение систем линейных алгебраических уравнений и много других. Характерной особенностью системы является ее открытость, то есть возможность ее модификации и адаптации к конкретным задачам пользователя.

MATLAB предоставляет широкие возможности для работы с сигналами, для расчета и проектирования аналоговых и цифровых фильтров, включая построение их частотных, импульсных и переходных характеристик. Имеются в системе и средства выполнения спектрального анализа и синтеза, в частности, реализации прямого и обратного преобразования Фурье. Благодаря этому ее довольно удобно использовать при проектировании электронных устройств. С системой MATLAB поставляются свыше ста подробно прокомментированных M-файлов, которые содержат демонстрационные примеры и определения новых операторов и функций. Наличие этих примеров и возможность работать в режиме непосредственных вычислений значительно облегчают изучение системы пользователями, заинтересованными в применении математических расчетов.

Система MATLAB использует собственный M-язык, который сочетает в себе положительные свойства различных известных языков программирования высокого уровня. С языком BASIC систему MATLAB роднит то, что она представляет собой интерпретатор (осуществляет пооператорное компилирование и выполнение программы, не образуя отдельного исполняемого файла), M-язык имеет незначительное количество операторов, в нем отсутствует необходимость объявлять типы и размеры переменных. От языка Pascal система MATLAB позаимствовала объектно-ориентированную направленность, то есть такое построение языка, которое обеспечивает образование новых типов вычислительных объектов на основе типов объектов, уже существующих в языке. Новые типы объектов (в MATLAB они называются классами) могут иметь собственные процедуры их преобразования (они определяют методы этого класса), причем новые процедуры могут быть вызваны с помощью обычных знаков арифметических операций и некоторых специальных знаков, которые применяются в математике.

Принципы сохранения значений переменных в MATLAB наиболее близки к тем, которые присущи языку FORTRAN, а именно: все переменные

являются локальными, то есть действуют лишь в границах той программной единицы (процедуры, функции или главной, управляющей программы), где им присвоены некоторые конкретные значения. При переходе к выполнению другой программной единицы значения переменных предыдущей программной единицы либо теряются (в случае, если выполненная программная единица представляет собой процедуру или функцию), либо становятся недостижимыми (если выполненная программа является управляющей).

В отличие от языков BASIC и Pascal в языке MATLAB нет глобальных переменных, действие которых распространялось бы на все программные единицы. Но при этом язык MATLAB обладает возможностью, которая отсутствует в других языках. Интерпретатор MATLAB позволяет в одном и том же сеансе работы выполнять несколько самостоятельных программ, причем все переменные, используемые в этих программах, являются для них общими и образуют единое рабочее пространство. Это дает возможность более рационально организовывать сложные (громоздкие) вычисления по типу оверлейных структур.

Вышеуказанные особенности системы MATLAB делают ее весьма гибкой и удобной в использовании вычислительной системой.

ЛАБОРАТОРНАЯ РАБОТА 2

ПРОСТЕЙШИЕ ВЫЧИСЛЕНИЯ В МАТЛАВ

Цель работы: научиться вычислять алгебраические выражения с использованием встроенных математических функций (работая из командной строки MatLab).

Порядок работы

1. Вычислите значение выражения

$$\frac{(\cos 8,16\pi - \sin 3,15\pi)^2}{2\operatorname{tg} 5,6 - \operatorname{tg} 3,4 \cdot \cos 3,38\pi} + \sqrt{\log 15,7} \cdot e^{-1/3}.$$

Для вычисления выражения необходимо ввести это выражение в командной строке и нажать "Enter":

```
>> ((cos(8.16*pi) - sin(3.15*pi))^2 / (2*tan(5.6) - tan(3.4)*cos(3.38*pi))) + sqrt(log(15.7))*exp(-1/3)
ans =
    0.0328
```

2. Найдите значения выражений, используя присвоение переменных

$$\text{а) } \frac{\frac{\tan 2,15}{\log 6,45} - \sqrt{\frac{\log 5,8}{\cos 3,4\pi}} + \frac{\sin 2,8\pi}{\log 1,6}}{\frac{\tan 2,15}{\log 6,45} + \frac{\sin 2,8\pi}{\log 1,6}}, \quad \text{б) } \frac{\left(\frac{\sin 2,8\pi}{\ln 1,6}\right)^2 - \sqrt{\frac{\operatorname{tg} 2,15}{\ln 6,45}}}{\sqrt{\frac{\ln 5,8}{\cos 3,4\pi}}}.$$

Чтобы найти значение следующего выражения, можно, воспользовавшись присвоением переменных, значениям $\frac{\tan 2,15}{\log 6,45}$, $\sqrt{\frac{\log 5,8}{\cos 3,4\pi}}$ и $\frac{\sin 2,8\pi}{\log 1,6}$ присвоить, соответственно, x , y и z . В результате, получим более простое выражение:

```
>> x = tan(2.15)/log(6.45);
>> y = sqrt(log(5.8)/cos(3.4*pi));
>> z = sin(2.8*pi)/log(1.6);
>> k = (x-y+z)/(x+z)
k = 1.0000-5.5420i
>> d = (z^2-sqrt(x))/y
>> d = -0.3797-0.6557i
```

3. Сохраните данную работу, написав в командной строке

```
>> save work01-01-08
```

или воспользуйтесь меню **File**, выбрав пункт **Save Workspace As**.

После появления диалогового окна **Save to MAT-File** укажите каталог и имя файла.



Задания

Вариант 1

1. Вычислите значение выражения

$$\frac{\operatorname{tg} 3,2 + \sqrt{e^{1,6} + \cos 8,4\pi}}{(\sin 1,6\pi - 3\operatorname{tg} 2,8)^2} - e^{-1/4} \cdot (1 + \cos 3,45\pi).$$

2. Найдите значения выражений, используя присвоение переменных

$$\text{а) } \frac{\frac{\cos 9,1\pi}{\log 3,4} + \frac{\sqrt{\log 2,6}}{\sin 5,4\pi} - \sqrt{\frac{\operatorname{tg} 5,48}{\log 3,6}}}{\sqrt{\frac{\operatorname{tg} 5,48}{\log 3,6}} - \left(\frac{\cos 9,1\pi}{\log 3,4}\right)^2}, \quad \text{б) } \frac{\frac{\sqrt{\log 2,6}}{\sin 5,4\pi} + \frac{\sqrt{\cos 9,1\pi}}{\sqrt{\log 3,4}}}{\sqrt{\frac{\log 5,48}{\log 3,6}}}.$$

Вариант 2

1. Вычислите значение выражения

$$\frac{\sqrt{6\operatorname{tg} 4,8 - \cos 1,3\pi}}{\sin 0,7\pi + (\log 12,2)^2} - \sqrt{\log 13,5} \cdot (1 - e^{1/2}).$$

2. Найдите значения выражений, используя присвоение переменных

$$\text{а) } \frac{\left(\frac{\cos 4,2\pi}{\sin 7,6\pi}\right)^2 - \frac{\tan 7,46}{\sin 5,4\pi} + \frac{\log 3,2}{2\log 7,3}}{\frac{\cos 4,2\pi}{\sin 7,6\pi} + \sqrt{\frac{\tan 7,46}{\sin 5,4\pi}}},$$

$$\text{б) } \frac{\sqrt{\frac{\log 3,2}{2\ln 7,3}} + \left(\frac{\cos 4,2\pi}{\sin 7,6\pi} - \frac{\tan 7,46}{\sin 5,4\pi}\right)}{\left(\frac{\tan 7,46}{\sin 5,4\pi}\right)^2}.$$

ЛАБОРАТОРНАЯ РАБОТА 3

РАБОТА С МАССИВАМИ. ВЕКТОРЫ-СТОЛБЦЫ И ВЕКТОРЫ-СТРОКИ

Цель работы: научиться работать с массивами, векторами и матрицами (работая из командной строки MatLab).

Порядок работы

1. Вычислите сумму векторов $a = \begin{pmatrix} 2.7 \\ 4.2 \\ 6.9 \end{pmatrix}$, $b = \begin{pmatrix} 5.8 \\ 1.6 \\ 4.2 \end{pmatrix}$.

Для этого необходимо в командной строке ввести сначала вектор **a**, используя квадратные скобки, и между значениями вектора поставить точку с запятой ";", а затем ввести вектор **b** таким же способом.

В конце каждой строки поставить ";", чтобы не загромождать промежуточными данными командное окно. А вот результат **c** выведем, то есть в конце $c = a + b$ не будем ставить точку с запятой ";":

```
>> a = [2.7; 4.2; 6.9];
>> b = [5.8; 1.6; 4.2];
>> c = a + b
c =
    8.5000
    5.8000
   11.1000
```

2. Вывести третий элемент вектора-строки $v = (0.6 \ 6.3 \ 9.4 \ 1.7 \ 5.2)$, заменить третий элемент вектора-строки на 8.1, записать в массив **w** пятый, первый и третий элементы.

Для вычислений вектор-строк необходимо записать их в командной строке также в квадратных скобках, но между значениями поставить запятые или пробелы:

```
>> v = [0.6 6.3 9.4 1.7 5.2];
```

Для того чтобы вывести лишь один элемент из вектора-столбца или вектора строки, необходимо написать его имя и далее в круглых скобках указать тот элемент, который необходимо вывести:

```
>> v(3)
ans = 9.4000
```

Если нам нужно заменить какой-то из элементов на другое значение, то необходимо после $v(3)$ поставить еще знак присваивания и ввести новое значение:

```
>> v(3) = 8.1
v =
    0.6000    6.3000    8.1000    1.7000    5.2000
```

Для помещения определенных элементов вектора в другой вектор в заданном порядке служит *индексация при помощи вектора*. Запись в массив w пятого, первого и третьего элементов v производится следующим образом:

```
>> w = [v(5); v(1); v(3)]
w =
    5.2000
    0.6000
    8.1000
```

3. В массиве вектора-строки $w = (0.3 \ 6.2 \ 7.1 \ 9.4 \ 5.1 \ 3.9 \ 2.2)$ заменить нулями элементы с третьего по седьмой; создать новый массив $w1$, используя элементы массива w со второго по четвертый и составить массив $w2$, содержащий элементы w , кроме пятого (используя сцепление строк).

Для обращения к блокам последовательно расположенных элементов вектора или вектора-строки служит *индексация при помощи знака двоеточия*. В массиве w , соответствующем вектор-строке из семи элементов, требуется заменить нулями элементы с третьего по седьмой:

```
>> w = [0.3 6.2 7.1 9.4 5.1 3.9 2.2];
>> w(3:7) = 0;
>> w
w = 0.3000 6.2000    0    0    0    0    0
>> w = [0.3 6.2 7.1 9.4 5.1 3.9 2.2];
>> w1 = w(2:4)
w1 =
    6.2000    7.1000    9.4000
>> w2 = [w(1:4) w(6:7)]
```

4. Перемножить элементы вектора-столбца $z = \begin{pmatrix} 3.6 \\ 5.4 \\ 1.7 \\ 6.9 \\ 2.2 \\ 4.3 \end{pmatrix}$, найти мини-

мальный и максимальный элементы вектора z и индекс (порядковый номер) максимального элемента (вызвать функцию `max` с двумя выходными аргументами):

```
>> z = [3.6; 5.4; 1.7; 6.9; 2.2; 4.3];
```

Перемножение элементов вектора-столбца или вектора-строки осуществляется при помощи функции prod:

```
>> p = prod(z)
p =
    2.1572e+003
```

Для нахождения минимума и максимума из элементов вектора служат встроенные функции min и max:

```
>> M=max(z)
M = 6.9
>> m = min(z)
m = 1.7
```

Вывод порядкового номера максимального элемента таков:

```
>> [M, k] = max(z)
M = 6.9
k = 4
```

5. Упорядочить вектор-строку $r = (8.4 \ -6.3 \ 2.5 \ -1.2 \ 0.6 \ 5.7)$:

- а) по возрастанию;
- б) по убыванию;
- в) в порядке возрастания их модулей;

г) по возрастанию с двумя выходными аргументами (это приведет к образованию массива индексов соответствия элементов упорядоченного и исходного массивов):

```
>> r = [8.4 -6.3 2.5 -1.2 0.6 5.7]
```

Функция упорядочения вектора по возрастанию его элементов sort следущая:

```
а) >> R = sort(r)
R =
   -6.3000   -1.2000    0.6000    2.5000    5.7000    8.4000
```

Чтобы упорядочить вектор-строку по убыванию, необходимо функцию sort записать со знаком минус и также указать саму вектор-строку со знаком минус:

```
б) >> R1 = -sort(-r)
R1 =
    8.4000    5.7000    2.5000    0.6000   -1.2000   -6.3000
```

Упорядочение элементов в порядке возрастания их модулей производится с привлечением функции abs:

```
в) >> R2 = sort(abs(r))
R2 =
    0.6000    1.2000    2.5000    5.7000    6.3000    8.4000
```

Упорядочение элементов по возрастанию с двумя выходными аргументами выглядит так:

```
r) >> [rs, ind] = sort(r)
rs =
    -6.3000   -1.2000    0.6000    2.5000    5.7000    8.4000
ind =
     2     4     5     3     6     1
```

6. а) Перемножить вектор-строки $v1 = (9 \ 2 \ -5 \ 4)$ и $v2 = (3 \ 7 \ -6 \ 1)$;
 б) возвести во вторую степень вектор $v1$;
 в) все элементы первого вектора $v1$ возвести в степень, равную соответствующим элементам второго вектора $v2$;
 г) разделить вектор $v1$ на $v2$ и $v2$ на $v1$;
 д) к вектору-строке $v = (2 \ 4 \ 10 \ 6)$ прибавить число 1.8, вычесть его из вектора, умножить вектор v на число 3 и разделить на это же число:

```
а) >> v1 = [9 2 -5 4];
>> v2 = [3 7 -6 1];
>> u = v1.*v2
u =
    27 14    30    4
б) >> p = v1.^2
p =
    81 4    25    16
в) P = v1.^v2
P =
    729.0000 128.0000    0.0001    4.0000
г) >> d = v1./v2
d =
    3.0000    0.2857    0.8333    4.0000
>> dinv = v1.\v2
dinv =
    0.3333    3.5000    1.2000    0.2500
д) >> v = [2 4 10 6];
>> s = v + 1.8
s =    3.8000    5.8000    11.8000    7.8000
>> r1 = v - 1.8
r1 =    0.2000    2.2000    8.2000    4.2000
>> p = v*3
     6 12  30  18
>> p1 = v/3
     0.6667    1.3333    3.3333    2.0000
```

Задания

Вариант 1

1. Вычислить сумму векторов: $a = \begin{pmatrix} 9.2 \\ 8.6 \\ 1.4 \end{pmatrix}$, $b = \begin{pmatrix} 7.2 \\ 2.5 \\ 4.9 \end{pmatrix}$.

2. Вывести второй элемент вектора-строки $v = (0.2 \ 8.3 \ 7.8 \ 3.1 \ 6.4)$, заменить четвертый элемент вектора-строки на 5.7, записать в массив w первый, пятый и третий элементы.

3. В массиве вектора-строки $w = (1.8 \ 6.4 \ 9.3 \ 0.5 \ 2.1 \ 3.7 \ 2.9)$ заменить нулями элементы с третьего по седьмой, создать новый массив $w1$, используя элементы массива w со второго по пятый и составить массив $w2$, содержащий элементы w , кроме второго (используя сцепление строк).

4. Перемножить элементы вектора-столбца $z = \begin{pmatrix} 3.7 \\ 2.4 \\ 1.5 \\ 0.2 \\ 9.6 \\ 5.3 \end{pmatrix}$, найти мини-

мальный и максимальный элементы вектора z и индекс (порядковый номер) максимального элемента (вызвать функцию `max` с двумя выходными аргументами).

5. Упорядочить вектор-строку $r = (-0.2 \ 6.3 \ -9.4 \ 3.8 \ 7.4 \ 0.1)$:

а) по возрастанию;

б) по убыванию;

в) в порядке возрастания их модулей;

г) по возрастанию с двумя выходными аргументами (это приведет к образованию массива индексов соответствия элементов упорядоченного и исходного массивов).

6. а) Перемножить вектор-строки $v1 = (2 \ 8 \ -4 \ 3)$ и $v2 = (-5 \ 7 \ -3 \ 1)$;

б) возвести во вторую степень вектор $v1$;

в) все элементы первого вектора $v1$ возвести в степень, равную соответствующим элементам второго вектора $v2$;

г) разделить вектор $v1$ на $v2$ и $v2$ на $v1$;

д) к вектору-строке $v = (8 \ 10 \ 2 \ 4)$ прибавить число 3.8, вычесть его из вектора, умножить вектор v на число 2 и разделить на это же число.

Вариант 2

1. Вычислить сумму векторов: $a = \begin{pmatrix} 3.1 \\ 2.8 \\ 7.3 \end{pmatrix}$, $b = \begin{pmatrix} 9.4 \\ 3.7 \\ 5.8 \end{pmatrix}$.

2. Вывести пятый элемент вектора-строки $v = (0.5 \ 9.2 \ 6.5 \ 2.7 \ 3.4)$, заменить второй элемент вектора-строки на 7.1, записать в массив w второй, пятый и первый элементы.

3. В массиве вектора-строки $w = (2.4 \ 9.3 \ 6.5 \ 0.8 \ 1.9 \ 3.7 \ 4.1)$ заменить нулями элементы со второго по шестой; создать новый массив $w1$, используя элементы массива w с третьего по седьмой и составить массив $w2$, содержащий элементы w , кроме четвертого (используя сцепление строк).

4. Перемножить элементы вектора-столбца $z = \begin{pmatrix} 2.5 \\ 3.9 \\ 8.1 \\ 0.4 \\ 6.2 \\ 5.3 \end{pmatrix}$, найти мини-

мальный и максимальный элементы вектора z и индекс (порядковый номер) максимального элемента (вызвать функцию `max` с двумя выходными аргументами).

5. Упорядочить вектор-строку $r = (-0.4 \ 3.6 \ 0.8 \ -1.3 \ 9.2 \ 6.5)$:

а) по возрастанию;

б) по убыванию;

в) в порядке возрастания их модулей;

г) по возрастанию с двумя выходными аргументами (это приведет к образованию массива индексов соответствия элементов упорядоченного и исходного массивов).

6. а) Перемножить вектор-строки $v1 = (4 \ 1 \ 7 \ -2)$ и $v2 = (-8 \ 3 \ -4 \ 1)$;

б) возвести во вторую степень вектор $v1$;

в) все элементы первого вектора $v1$ возвести в степень, равную соответствующим элементам второго вектора $v2$;

г) разделить вектор $v1$ на $v2$ и $v2$ на $v1$;

д) к вектору-строке $v = (6 \ 2 \ 8 \ 4)$ прибавить число 2.6, вычесть его из вектора, умножить вектор v на число 4 и разделить на это же число.

ЛАБОРАТОРНАЯ РАБОТА 4 РАБОТА С МАССИВАМИ

Цель работы: научиться строить таблицы значений функции, строить графики функции одной переменной, а также научиться перемножать вектора (скалярное произведение, векторное произведение, внешнее произведение).

Порядок работы

1. Выведите таблицу значений функции

$$y(x) = \frac{\cos^2 x}{1 + \sin x} - e^{-x} \cdot \log x$$

в точках 0.1, 0.4, 0.6, 0.7, 1.1, 1.5, 2.3.

Вначале необходимо записать данные значения в вектор-строку x , а затем вычислить значения функции $y(x)$ от каждой этой точки, поэтому необходимо в записи самой функции использовать поэлементные операции:

```
>> x = [0.1 0.4 0.6 0.7 1.1 1.5 2.3];  
>> y = cos(x).^2/(1+sin(x)) - exp(-x).*log(x)  
y =  
    2.3728    0.9035    0.5696    0.4664    0.2576    0.1988  
    0.2058
```

2. Заполните вектор-столбец элементами, начинающимися с нуля и до единицы с шагом 0.3 (используя операцию транспонирование) и вектор-строку, начинающуюся с единицы до двух с шагом -0.2 :

```
>> x = [0:0.3:1]'  
x =  
    0  
    0.3000  
    0.6000  
    0.9000  
>> y = [2: -0.2:1]  
y =  
    2.0000    1.8000    1.6000    1.4000    1.2000    1.0000
```

3. Построить график функции $y(x) = e^x \cdot \cos 5x$ на отрезке $[0, 5]$.

Вывод отображения функции в виде графика состоит из следующих этапов:

1. Задание вектора значений аргумента x .
2. Вычисление вектора y значений функции $y(x)$.
3. Вызов команды `plot` для построения графика.

Команды для задания вектора x и вычисления функции лучше завершать точкой с запятой для подавления вывода в командное окно их значений (после команды `plot` точку с запятой ставить не обязательно, т.к. она ничего не выводит в командное окно). Не забудьте использовать поэлементное умножение `.*`:

```
>> x = [0:0.05:5];
>> y = exp(x).*cos(5*x);
>> plot(x, y)
```

4. Постройте на отрезке $[-1, -0.3]$ графики функций $f(x) = \cos \frac{1}{x^2}$,

$$g(x) = \cos \frac{1.5}{x^2} :$$

```
>> x = [-1:0.005: -0.3];
>> f = cos(x.^ -2);
>> g = cos(1.5*x.^ -2);
>> plot(x, f, x, g)
```

5. а) Найдите скалярное произведение векторов: $a = \begin{bmatrix} 1.5 \\ -4.8 \\ 0.9 \end{bmatrix}$, $b = \begin{bmatrix} 6.3 \\ 9.7 \\ -1.4 \end{bmatrix}$;

б) найдите длину (модуль) вектора a ; в) найдите векторное произведение векторов a и b .

Для вычисления скалярного произведения необходимо просуммировать компоненты вектора, полученного в результате поэлементного умножения a на b , т.е. надо использовать функцию `sum` и поэлементное умножение:

```
а) >> a = [1.5; -4.8; 0.9];    б) >> d = sqrt (sum(a .* a))    в) >> c = cross (a, b)
>> b = [6.3; 9.7; -1.4];      d =
>> s = sum(a.*b)              5.1088                -2.0100
s =                             7.7700
-38.3700                       44.7900
```

6. Найдите объем параллелепипеда, если $a = \begin{bmatrix} 2.5 \\ 0 \\ 0 \end{bmatrix}$, $b = \begin{bmatrix} 0.3 \\ 3.2 \\ 0 \end{bmatrix}$, $c = \begin{bmatrix} -0.3 \\ -2.1 \\ 4.2 \end{bmatrix}$:

```
>> a = [2.5; 0; 0];
>> b = [0.3; 3.2; 0];
>> c = [-0.3; -2.1; 4.2];
>> v = abs (sum(a.*cross(b, c)))
v =
33.6000
```

Задания

Вариант 1

1. Выведите таблицу значений функции

$$y(x) = \frac{2 \cdot \cos^2 x}{1 + \tan x} - e^{2x} \cdot \log x$$

в точках 0.1, 0.3, 0.4, 0.6, 1.3, 1.4, 2.1.

2. Заполните вектор-столбец элементами, начинающимися с нуля и до единицы с шагом 0.1 (используя операцию транспонирование) и вектор-строку, начинающуюся с единицы до двух с шагом -0.2 .

3. Постройте график функции $y(x) = 2 \cdot e^{-x} \cdot \sin 3x$ на отрезке $[0, 4]$.

4. Постройте на отрезке $[-1, -0.5]$ графики функций $f(x) = \sin \frac{1}{3 \cdot x^2}$,

$$g(x) = 2 \cdot \cos \frac{1.8}{x^2}.$$

5. а) Найдите скалярное произведение векторов: $a = \begin{bmatrix} 2.1 \\ -3.7 \\ 0.6 \end{bmatrix}$, $b = \begin{bmatrix} 4.1 \\ 7.5 \\ -0.9 \end{bmatrix}$;

б) найдите длину (модуль) вектора a ; в) найдите векторное произведение векторов a и b .

6. Найдите объем параллелепипеда, если $a = \begin{bmatrix} 1.5 \\ 0 \\ 0 \end{bmatrix}$, $b = \begin{bmatrix} 0.5 \\ 6.1 \\ 0 \end{bmatrix}$, $c = \begin{bmatrix} -0.7 \\ -3.6 \\ 8.5 \end{bmatrix}$.

Вариант 2

1. Выведите таблицу значений функции

$$y(x) = \frac{5 \cdot \cos^2 x}{3 + \sin 2x} + 4 \cdot e^{-x} \cdot \tan x$$

в точках 0.1, 0.2, 0.5, 0.8, 1.1, 1.3, 2.2.

2. Заполните вектор-столбец элементами, начинающимися с нуля и до единицы с шагом 0.2 (используя операцию транспонирование) и вектор-строку, начинающуюся с единицы до двух с шагом -0.2 .

3. Постройте график функции $y(x) = e^{-x} \cdot (1 + \cos 7x)$ на отрезке $[0, 7]$.

4. Постройте на отрезке $[-1, -0.6]$ графики функций $f(x) = \cos \frac{1.4}{x^2}$,
 $g(x) = \sin \frac{1}{2 \cdot x^2}$.

5. а) Найдите скалярное произведение векторов: $a = \begin{bmatrix} 0.5 \\ -1.6 \\ 0.4 \end{bmatrix}$, $b = \begin{bmatrix} 7.3 \\ 4.7 \\ -5.4 \end{bmatrix}$;

б) найдите длину (модуль) вектора a ; в) найдите векторное произведение векторов a и b .

6. Найдите объем параллелепипеда, если $a = \begin{bmatrix} 7.2 \\ 0 \\ 0 \end{bmatrix}$, $b = \begin{bmatrix} 0.4 \\ 8.6 \\ 0 \end{bmatrix}$,

$$c = \begin{bmatrix} -1.8 \\ -4.8 \\ 5.4 \end{bmatrix}.$$

ЛАБОРАТОРНАЯ РАБОТА 5 ДВУМЕРНЫЕ МАССИВЫ И МАТРИЦЫ

Цель работы: научиться вводить матрицы; производить математические операции с ними, поэлементные операции; вычислять функции от элементов матриц, читать и записывать с использованием текстового файла простейшее визуализации матричных данных.

Порядок работы

1. а) Найдите сумму и разность матриц $C = \begin{pmatrix} 4 & 5 & -2 \\ -6 & -3 & 7 \end{pmatrix}$,
 $A = \begin{pmatrix} 4 & -7 & 8 \\ 2 & 6 & 0 \end{pmatrix}$; б) умножьте матрицы C и $B = \begin{pmatrix} 3 & 5 & -2 \\ 7 & 2 & -8 \\ -4 & 6 & 0 \end{pmatrix}$; в) полученную

матрицу умножьте на 5:

```
>> C = [4 5 -2; -6 -3 7];
>> A = [4 -7 8; 2 6 0];
>> S = C + A
S =
     8 -2     6
    -4  3     7
>> R = C - A
R =
     0 12    -10
    -8 -9     7
>> B = [3 5 -2; 7 2 -8; -4 6 0];
>> P = C*B
P =
     55     18    -48
    -67     6     36
>> T = P*5
T =
    275     90   -240
   -335     30    180
```

2. Найдите значение выражения $(A - C)B^2(A + C)^T$:

```
>> (A-C)*B^2*(A+C)'
ans =
```

-4328 1820
4166 -5505

3. Вычислите выражение $[3 \ 2 \ -4] \cdot \begin{pmatrix} 6 & 3 & 0 \\ -8 & 5 & -1 \\ 7 & 2 & -4 \end{pmatrix} \cdot \begin{bmatrix} -6 \\ 5 \\ 1 \end{bmatrix}$:

```
>> a = [3 2 -4];
>> B = [6 3 0; -8 5 -1; 7 2 -4];
>> c = [-6; 5; 1];
>> a*B*c
ans =
    225
```

4. Решите систему из 3 уравнений $\begin{cases} 1.6x_1 + 6.3x_2 - 0.7x_3 = 3.4; \\ 0.8x_1 + 6.8x_2 + 9.1x_3 = 7.5; \\ -0.5x_1 + 5.5x_2 + 8.1x_3 = 5.1; \end{cases}$

```
>> A = [1.6 6.3 -0.7; 0.8 6.8 9.1; -0.5 5.5 8.1];
>> b = [3.4; 7.5; 5.1];
>> x = A\b
x =
    1.1635
    0.2995
    0.4981
```

5. Решите систему линейных уравнений № 4, матрица и вектор правой части которой хранятся в текстовых файлах matr.txt, rside.txt, и запишите результат в файл sol.txt.

Задания

Вариант 1

1. а) Найдите сумму и разность матриц $C = \begin{pmatrix} 1 & 3 & -5 \\ -2 & -6 & 9 \end{pmatrix}$,

$A = \begin{pmatrix} 9 & -5 & 4 \\ 1 & 7 & 0 \end{pmatrix}$; б) умножьте матрицы C и $B = \begin{pmatrix} 6 & 3 & -7 \\ 5 & 4 & -9 \\ -6 & 2 & 1 \end{pmatrix}$; в) полученную

матрицу умножьте на 4.

2. Найдите значение выражения $(A + C)B^3(A + C)^T$.

3. Вычислите выражение $[6 \ 1 \ -5] \cdot \begin{pmatrix} 5 & 3 & 1 \\ -7 & 4 & -1 \\ 8 & 6 & -9 \end{pmatrix} \cdot \begin{bmatrix} -2 \\ 9 \\ 1 \end{bmatrix}$.

4. Решите систему из 3 уравнений $\begin{cases} 0.6x_1 + 8.8x_2 - 4.2x_3 = 9.3; \\ 6.1x_1 + 5.9x_2 + 1.1x_3 = 5.5; \\ -2.5x_1 + 4.5x_2 + 0.1x_3 = 0.1. \end{cases}$

5. Решите систему линейных уравнений № 4, матрица и вектор правой части которой хранятся в текстовых файлах matr.txt, rside.txt, и запишите результат в файл sol.txt.

Вариант 2

1. а) Найдите сумму и разность матриц $C = \begin{pmatrix} 7 & 1 & -3 \\ -4 & -2 & 6 \end{pmatrix}$,

$A = \begin{pmatrix} 3 & -5 & 9 \\ 4 & 7 & 0 \end{pmatrix}$; б) умножьте матрицы C и $B = \begin{pmatrix} 6 & 2 & -3 \\ 4 & 5 & -7 \\ -9 & 1 & 0 \end{pmatrix}$; в) полученную

матрицу умножьте на 6.

2. Найдите значение выражения $(A + C)B^2(A - C)^T$.

3. Вычислите выражение $[4 \ 8 \ -1] \cdot \begin{pmatrix} 3 & 7 & 0 \\ -6 & 4 & -2 \\ 5 & 1 & -8 \end{pmatrix} \cdot \begin{bmatrix} -9 \\ 4 \\ 1 \end{bmatrix}$.

4. Решите систему из 3 уравнений $\begin{cases} 1.3x_1 + 0.6x_2 - 2.5x_3 = 8.1; \\ 5.4x_1 + 2.7x_2 + 0.1x_3 = 7.3; \\ -0.8x_1 + 9.2x_2 + 4.9x_3 = 0.6. \end{cases}$

5. Решите систему линейных уравнений № 4, матрица и вектор правой части которой хранятся в текстовых файлах matr.txt, rside.txt, и запишите результат в файл sol.txt.

ЛАБОРАТОРНАЯ РАБОТА 6 БЛОЧНЫЕ МАТРИЦЫ

Цель работы: научиться конструировать блочные матрицы, заполнять матрицы при помощи индексации, создавать матрицы специального вида.

Порядок работы

1. а) Введите четыре квадратные матрицы (A, B, C, D) размерностью два и создайте из них блочную матрицу $K = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$; б) составьте блочную

матрицу $M = \left(\begin{array}{c|c} S & a \\ \hline b & 3.5 \end{array} \right)$, где $S = \begin{pmatrix} 5 & 0 \\ 6 & 4 \end{pmatrix}$, $a = \begin{pmatrix} 8 \\ 7 \end{pmatrix}$, $b = (-6 \ 3)$; в) выделите

блоки из полученной матрицы $K = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}$ и выделите 3-ю строку из

матрицы M :

```
а)>> K = [A B; C D] или K = [[A; C] [B; D]]
```

```
б)>> S = [5 0; 6 4];
```

```
>> a = [8; 7];
```

```
>> b = [-6 3];
```

```
>> M = [S a; b 3.5] или M = [[S; b] [a; 3.5]]
```

```
M =
```

```
    5    0    8
```

```
    6    4    7
```

```
   -6    3    3.5
```

```
в)>> K1 = K(1:2, 3:4)
```

```
>> K2 = K(3:4, 1:2)
```

```
>> m = M(3, :)
```

```
m =  -6         3         3.5
```

2. Удалите первую строку матрицы K из № 1 и 3-ий столбец.

```
>> K(1, :) = []
```

```
>> K(:, 3) = []
```

3. а) Заполните прямоугольную и квадратную матрицу нулями, единицами и случайными числами при помощи индексации и встроенных функций; б) заполните вектор-строку семью случайными числами; в) создайте диагональную матрицу, у которой все внедиагональные элементы равны нулю:

```

а)>> A0(1:3, 1:4) = 0           >> A1(1:3, 1:4) = 1           >> A = rand(3, 4)
>> B0(1:3, 1:3) = 0           >> B1(1:3, 1:3) = 1           >> B = rand(3)
>> A0 = zeros(3, 4)           >> A1 = ones(3: 4)           >> A = randn(3, 4)
>> B0 = zeros(3)              >> B1 = ones(3)              >> B = randn(3)

```

```
б) >> r = rand(1, 7)
```

```
в) >> d = [1; 2; 3; 4];
```

```
>> D = diag(d)
```

D =

```

1  0  0  0
0  2  0  0
0  0  3  0
0  0  0  4

```

4. Заполните и запишите в файлы матрицы:

$$M = \begin{pmatrix} -2 & 0 & 0 & 0 & 0 & 5 & 5 & 5 & 5 & 5 \\ 0 & -2 & 0 & 0 & 0 & 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & -2 & 0 & 0 & 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 0 & -2 & 0 & 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 0 & 0 & -2 & 5 & 5 & 5 & 5 & 5 \\ 2 & 2 & 2 & 2 & 2 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 1 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 \end{pmatrix}.$$

```
>> M = [-2*eye(5) 5*ones(5); 2*ones(5) eye(5)];
```

```
>> save 'M.txt' M -ascii
```

```
>> G = 4*eye(7) + diag(ones(1, 6), 1) + diag(ones(1, 6), -1)
```

```
>> save 'G.txt' G -ascii
```

Задания

Вариант 1

1. а) Введите четыре квадратные матрицы (A, B, C, D) размерностью два и создайте из них блочную матрицу $K = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$; б) составьте блочную матрицу $M = \left(\begin{array}{c|c} S & a \\ \hline b & 4.8 \end{array} \right)$, где $S = \begin{pmatrix} 3 & 9 \\ 7 & 1 \end{pmatrix}$, $a = \begin{pmatrix} 9 \\ 3 \end{pmatrix}$, $b = (-2 \ 7)$; в) выделите

блоки из полученной матрицы $K = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}$ и выделите 3-ю стро-

ку из матрицы M .

2. Удалите первую строку матрицы K из № 1 и 3-й столбец.

3. а) Заполните прямоугольную и квадратную матрицу нулями, единицами и случайными числами при помощи индексации и встроенных функций; б) заполните вектор-строку семью случайными числами; в) создайте диагональную матрицу, у которой все внедиагональные элементы равны нулю.

4. Заполните и запишите в файлы матрицы:

$$M = \begin{pmatrix} -5 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 3 \\ 0 & -5 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & -5 & 0 & 0 & 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & -5 & 0 & 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & -2 & 3 & 3 & 3 & 3 & 3 \\ 7 & 7 & 7 & 7 & 7 & 1 & 0 & 0 & 0 & 0 \\ 7 & 7 & 7 & 7 & 7 & 0 & 1 & 0 & 0 & 0 \\ 7 & 7 & 7 & 7 & 7 & 0 & 0 & 1 & 0 & 0 \\ 7 & 7 & 7 & 7 & 7 & 0 & 0 & 0 & 1 & 0 \\ 7 & 7 & 7 & 7 & 7 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 6 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 6 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 6 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 6 \end{pmatrix}.$$

Вариант 2

1. а) Введите четыре квадратные матрицы (A, B, C, D) размерностью два и создайте из них блочную матрицу $K = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$; б) составьте блочную

матрицу $M = \left(\begin{array}{c|c} S & a \\ \hline b & 6.7 \end{array} \right)$, где $S = \begin{pmatrix} 4 & 3 \\ 1 & 8 \end{pmatrix}$, $a = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$, $b = (-8 \ 2)$; в) выделите

блоки из полученной матрицы $K = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}$ и выделите 3-ю строку из

матрицы M .

2. Удалите первую строку матрицы K из № 1 и 3-й столбец.

3. а) Заполните прямоугольную и квадратную матрицу нулями, единицами и случайными числами при помощи индексации и встроенных функций; б) заполните вектор-строку семью случайными числами; в) создайте диагональную матрицу, у которой все внедиагональные элементы равны нулю.

4. Заполните и запишите в файлы матрицы:

$$M = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 6 & 6 & 6 & 6 & 6 \\ 0 & 4 & 0 & 0 & 0 & 6 & 6 & 6 & 6 & 6 \\ 0 & 0 & 4 & 0 & 0 & 6 & 6 & 6 & 6 & 6 \\ 0 & 0 & 0 & 4 & 0 & 6 & 6 & 6 & 6 & 6 \\ 0 & 0 & 0 & 0 & 4 & 6 & 6 & 6 & 6 & 6 \\ 3 & 3 & 3 & 3 & 3 & -2 & 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 & 3 & 0 & -2 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 & 3 & 0 & 0 & -2 & 0 & 0 \\ 3 & 3 & 3 & 3 & 3 & 0 & 0 & 0 & -2 & 0 \\ 3 & 3 & 3 & 3 & 3 & 0 & 0 & 0 & 0 & -2 \end{pmatrix}, \quad G = \begin{pmatrix} 7 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 7 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 7 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 7 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 7 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 7 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 7 \end{pmatrix}.$$

ЛАБОРАТОРНАЯ РАБОТА 7

ВИЗУАЛИЗАЦИЯ МАТРИЦИ ПОЭЛЕМЕНТНЫЕ ОПЕРАЦИИ НАД НИМИ

Цель работы: научиться представлять в наглядном виде большие матрицы при помощи визуализации, производить поэлементные операции над матрицами, работать со встроенными функциями, применять функции обработки данных к матрицам, строить графики функций двух переменных.

Порядок работы

1. В матрице M а) вычислите сумму по столбцам и по строкам; б) отсортируйте элементы этой матрицы в порядке возрастания их столбцов и строк; в) вычислите максимальные и минимальные элементы в соответствующих столбцах матрицы M и строках:

$$M = \begin{pmatrix} 2 & -3 & -5 \\ 6 & -2 & 4 \\ 3 & -1 & 1 \end{pmatrix} :$$

```
>> M = [2 -3 -5
        6 -2  4
        3 -1  1];
>> s = sum (M)
s =
    11 -6    0
>> s2 = sum (M, 2)
s2 =
    -6
     8
     3
>> MC = sort (M)
MC =
     2 -1   -5
     3 -2    1
     6  3    4
MR = sort (M, 2)
```

```

MR =
    -5 -3     2
    -2  4     6
    -1  1     3
>> mx = max (M)
mx =
     6 -2     4
>> mn = min (M)
mn =
     2 -2    -5
>> mx1 = max (M, [], 2)           >> mn1 = min (M, [], 2)
mx1 =                               mn1 =
     2                               -5
     6                               -2
     3                               -1
    
```

2. Создайте квадратную матрицу A размера пять, состоящую из случайных целых чисел от нуля до десяти, вычтите из нее матрицу, состоящую из шестерок, и введите необходимые выражения для вычисления норм полученной матрицы.

Используйте полученную квадратную матрицу A . Вычислите приведенные ниже величины (нормы матрицы):

$$p = \max_{1 \leq i \leq n} \sum_{k=1}^n |a_{ik}|, \quad q = \max_{1 \leq k \leq n} \sum_{i=1}^n |a_{ik}|, \quad r = \left(\sum_{i,k=1}^n a_{ik}^2 \right)^{1/2}, \quad s = \sum_{i,k=1}^n |a_{ik}|:$$

```

>> A = round (10*rand(5)) - 6*ones (5)
>> p = max (sum(abs(A)))
>> q = max (sum(abs(A), 2))
>> r = sqrt (sum(sum(abs(A).^2)))
>> s = sum(sum(abs(A)))
    
```

Задания

Вариант 1

1. В матрице M а) вычислите сумму по столбцам и по строкам; б) отсортируйте элементы этой матрицы в порядке возрастания их столбцов и строк; в) вычислите максимальные и минимальные элементы в соответствующих столбцах матрицы M и строках:

$$M = \begin{pmatrix} 2 & -4 & -3 \\ 3 & -1 & 7 \\ 6 & -6 & 0 \end{pmatrix}.$$

2. Создайте квадратную матрицу размера три, состоящую из случайных целых чисел от нуля до семи, вычтите из нее матрицу, состоящую из двоек, и введите необходимые выражения для вычисления норм полученной матрицы.

Используйте полученную квадратную матрицу A . Вычислите приведенные ниже величины (нормы матрицы):

$$p = \max_{1 \leq i \leq n} \sum_{k=1}^n |a_{ik}|, \quad q = \max_{1 \leq k \leq n} \sum_{i=1}^n |a_{ik}|, \quad r = \left(\sum_{i,k=1}^n a_{ik}^2 \right)^{1/2}, \quad s = \sum_{i,k=1}^n |a_{ik}|.$$

Вариант 2

1. В матрице M а) вычислите сумму по столбцам и по строкам; б) отсортируйте элементы этой матрицы в порядке возрастания их столбцов и строк; в) вычислите максимальные и минимальные элементы в соответствующих столбцах матрицы M и строках:

$$M = \begin{pmatrix} 4 & -3 & -4 \\ 2 & -7 & 6 \\ 5 & -2 & 1 \end{pmatrix},$$

2. Создайте квадратную матрицу размера четыре, состоящую из случайных целых чисел от нуля до восьми, вычтите из нее матрицу, состоящую из троек, и введите необходимые выражения для вычисления норм полученной матрицы.

Используйте полученную квадратную матрицу A . Вычислите приведенные ниже величины (нормы матрицы):

$$p = \max_{1 \leq i \leq n} \sum_{k=1}^n |a_{ik}|, \quad q = \max_{1 \leq k \leq n} \sum_{i=1}^n |a_{ik}|, \quad r = \left(\sum_{i,k=1}^n a_{ik}^2 \right)^{1/2}, \quad s = \sum_{i,k=1}^n |a_{ik}|.$$

ЛАБОРАТОРНАЯ РАБОТА 8

ГРАФИКИ, ДИАГРАММЫ И ГИСТОГРАММЫ

Цель работы: научиться представлять векторные и матричные данные в виде диаграмм и гистограмм.

Порядок работы

1. Отобразите функцию $x(t) = \sin t \cdot e^t$ на отрезке $[-1, 1]$ в виде столбчатой диаграммы без промежутков:

```
>> t = [-1:0.1:1];  
>> x = sin (t).*exp (t);  
>> bar (t, x, 1.0)
```

2. Напишите команды построения диаграммы с отдельным сектором, соответствующим максимальному значению среди элементов вектора, автоматически создав вспомогательный вектор. Используйте функции `zeros` для создания нулевого вектора той же длины, что `x`, и `max` с двумя выходными аргументами для поиска номера максимального элемента в векторе `x`. Ниже приведена требуемая последовательность команд:

```
>> parts = zeros (size(data));  
>> [mx, ind] = max (data);  
>> parts (ind) = 1;  
>> pie (data, parts)
```

Визуализация векторных данных может быть осуществлена при помощи `pie3` и `bar3`, которые строят трехмерные круговые и столбчатые диаграммы, например команды

```
>> data = [24.1 17.4 10.9];  
>> parts = [1 0 0];  
>> pie3 (data, parts)
```

которые приводят к появлению трехмерной круговой диаграммы с отделенным сектором.

3. Для увеличения числа интервалов следует в качестве второго аргумента указать число интервалов, например `hist (data, 50)`. Вместо автоматического разбиения на равные интервалы можно использовать собственное, задав вторым аргументом вектор, содержащий центры интервалов:

```
>> data =[0.9 1.0 1.1 1.2 1.4 2.4 3.0 3.3];  
>> centers = [1.1 2.3 3.2];  
>> hist (data, centers)
```

что приводит к построению диаграммы, где звездочки на горизонтальной оси отмечают центры интервалов.

4. Постройте графики функций $f(x) = e^{-0.1x} \sin^2 x$ и $g(x) = e^{-0.2x} \sin^2 x$ на отрезке $[-2\pi, 2\pi]$. Сгенерируйте вектор-строку значений аргумента x и вектор-строк f и g , содержащих значения функций. Команда `plot` с двумя парами аргументов приводит к построению графика:

```
>> x = [-2 * pi : pi / 20 : 2 * pi];
>> f = exp (-0.1*x) .* sin (x) .^2;
>> g = exp (-0.2*x) .* sin (x) .^2;
>> plot (x, f, x, g)
```

Функции не обязательно должны быть определены на одном и том же отрезке. В этом случае при построении графиков MatLab выбирает максимальный отрезок, содержащий остальные. Важно только в каждой паре векторов абсцисс и ординат указать соответствующие друг другу вектора, например:

```
>> x1 = [-pi : 0.01 : 2*pi];
>> f = exp (-0.1*x1) .* sin (x1) .^2;
>> x2 = [-2*pi : 0.01 : pi];
>> g = exp (-0.2*x2) .* sin (x2) .^2;
>> plot (x1, f, x2, g)
```

5. Иногда требуется сравнить поведение двух функций, значения которых сильно отличаются друг от друга. Графики функции с небольшими значениями практически сливаются с осью абсцисс. Установить их вид не удастся. В этой ситуации помогает функция `plotyy`, которая выводит графики в окно с двумя вертикальными осями, имеющими подходящий масштаб.

Сравните, например, две функции: $f(x) = \frac{1}{x^3}$ и $F(x) = 1000 \cdot (x + 0.5)^{-4}$:

```
>> x = [0.5 : 0.01 : 3];
>> f = x .^ -3;
>> F = 1000*(x + 0.5) .^ -4;
>> plotyy (x, f, x, F)
```

6. Аргументы `loglog`, `semilogx` и `semilogy` задаются в виде пары векторов значений абсцисс и ординат так же, как и для функции `plot`. Постройте, например, графики функций $f(x) = \ln 0.5x$ и $g(x) = \sin \ln x$ на отрезке $[0.1, 5]$ в логарифмическом масштабе по оси x :

```
>> x = [0.1 : 0.01 : 10];
>> f = log (0.5*x);
>> g = sin (log (x));
>> semilogx (x, f, x, g)
```

Следующие команды выводят графики изменения суточной температуры, которые снабжены всей необходимой информацией:

```
>> time = [0 4 7 9 10 11 12 13 13.5 14 14.5 15 16 17 18 20 22];
>> temp1 = [14 15 14 16 18 17 20 22 24 28 25 20 16 13 13 14 13];
>> temp2 = [12 13 13 14 16 18 20 20 23 25 25 20 16 12 12 11 10];
>> plot (time, temp1, 'ro-', time, temp2, 'go-')
>> grid on
>> title ('Суточные температуры')
>> xlabel ('Время (час.)')
>> ylabel ('Температура (C)')
>> legend ('10 мая', '11 мая')
```

7. Для того чтобы проверить свои знания о построении графиков и работе с массивами, постройте график функции, заданной кусочным образом:

$$y(x) = \begin{cases} \pi \cdot \sin x, & -2\pi \leq x \leq -\pi; \\ \pi - |x|, & -\pi < x < \pi; \\ \pi \cdot \sin^3 x, & \pi \leq x \leq 2\pi. \end{cases}$$

Сначала необходимо вычислить каждую из трех ветвей, т.е. фактически получить три пары массивов: x_1 и y_1 , x_2 и y_2 , x_3 и y_3 , затем объединить значения абсцисс в вектор x , а значения ординат в y и построить график функции, задаваемой парой массивов x и y .

8. Изучите основные возможности, предоставляемые MatLab для визуализации функций двух переменных, на примере построения графика

$$z(x, y) = 4 \cdot \sin 2\pi x \cdot \cos 1.5\pi y \cdot (1 - x^2) \cdot y \cdot (1 - y)$$

на прямоугольной области определения $x \in [-1, 1]$, $y \in [0, 1]$. Подготовьте матрицы с координатами узлов сетки и значениями функции:

```
>> [X, Y] = meshgrid (-1 : 0.05 : 1);
>> Z = 4*sin (2*pi*X) .* cos (1.5*pi*Y) .* (1 - X.^2) .* Y .* (1 - Y);
```

Для построения каркасной поверхности используется функция `mesh`, вызываемая с тремя аргументами:

```
>> mesh (X, Y, Z)
```

Цвет линий поверхности соответствует значениям функции. MatLab рисует только видимую часть поверхности. При помощи команды `hidden off` можно сделать каркасную поверхность "прозрачной", добавив скрытую часть. Команда `hidden on` убирает невидимую часть поверхности, возвращая графику прежний вид.

Функция `surf` строит каркасную поверхность графика функции и заливает каждую клетку поверхности определенным цветом, зависящим от значения функции в точках, соответствующих углам клетки:

```
>> surf (X, Y, Z)
```

В пределах каждой клетки цвет постоянный. Команда `shading flat` позволяет убрать каркасные линии. Для получения поверхности, плавно залитой цветом, предназначена команда `shading interp`.

При помощи `shading faceted` можно вернуться к виду с каркасными линиями.

В MatLab определена команда `colorbar`, которая выводит рядом с графиком столбик, устанавливающий соответствие между цветом и значением функции. Постройте при помощи `surf` график поверхности и дополните его информацией о цвете:

```
>> surf (X, Y, Z)
```

```
>> colorbar
```

Команды `meshc` или `surfc` позволяют получить более точное представление о поведении функции. Эти команды строят каркасную поверхность или залитую цветом каркасную поверхность и размещают на плоскости $xу$ линии уровня функции (линии постоянства значений функции):

```
>> surfc (X, Y, Z)
```

```
>> colorbar
```

9. Постройте поверхность, состоящую из линий уровня, соответствующих значениям функции от 0 до 0.5 с шагом 0.01:

```
>> levels = [0:0.01:0.5];
```

```
>> contour3 (X, Y, Z, levels)
```

```
>> colorbar
```

Задания

Вариант 1

1. Отобразите функцию $x(t) = \cos t \cdot e^{-t}$ на отрезке $[-1, 1]$ в виде столбчатой диаграммы без промежутков.

2. Напишите команды построения диаграммы с отдельным сектором, соответствующим минимальному значению среди элементов вектора, автоматически создав вспомогательный вектор. Используйте функции `zeros` для создания нулевого вектора той же длины, что x , и `max` с двумя выходными аргументами для поиска номера минимального элемента в векторе x .

3. Постройте диаграмму, где звездочки на горизонтальной оси будут отмечать центры интервалов.

4. Постройте графики функций $f(x) = e^{x/5} \cos^2 x$ и $g(x) = 5 \cdot e^{-x} \sin^2 x$ на отрезке $[-2\pi, 2\pi]$. Сгенерируйте вектор-строку значений аргумента x и вектор-строк f и g , содержащих значения функций.

5. Сравните две функции: $f(x) = \frac{1}{x^3}$ и $F(x) = 1000 \cdot (x + 0.2)^{-4}$.

6. Постройте графики функций $f(x) = \log 0.7x$ и $g(x) = \cos \log x$ на отрезке $[0.1, 5]$ в логарифмическом масштабе по оси x .

7. Постройте график функции, заданной кусочным образом:

$$y(x) = \begin{cases} \pi \cdot \sin x, & -2\pi \leq x \leq -\pi; \\ \pi - |x|, & -\pi < x < \pi; \\ \pi \cdot \sin^3 x, & \pi \leq x \leq 2\pi. \end{cases}$$

Сначала необходимо вычислить каждую из трех ветвей, т.е. фактически получить три пары массивов: x_1 и y_1 , x_2 и y_2 , x_3 и y_3 , затем объединить значения абсцисс в вектор x , а значения ординат в y и построить график функции, задаваемой парой массивов x и y .

8. Постройте график функции

$$z(x, y) = 6 \cdot \sin 2\pi \cdot x \cdot \cos 1.5\pi \cdot y \cdot (1 - x^2) \cdot y \cdot (1 - y)$$

на прямоугольной области определения $x \in [-1, 1]$, $y \in [0, 1]$:

а) построить график с каркасной поверхностью (сделать ее «прозрачной» и вернуть в обычный вид);

б) залить каждую клетку поверхности определенным цветом, зависящим от значения функции в точках, соответствующих углам клетки;

в) получить поверхность, плавно залитую цветом;

г) вывести рядом с графиком столбик, устанавливающий соответствие между цветом и значением функции.

9. Постройте поверхность, состоящую из линий уровня, соответствующих значениям функции от 0 до 0.5 с шагом 0.01.

Вариант 2

1. Отобразите функцию $x(t) = \cos t \cdot \log t$ на отрезке $[-1, 1]$ в виде столбчатой диаграммы без промежутков.

2. Напишите команды построения диаграммы с отдельным сектором, соответствующим минимальному значению среди элементов вектора, автоматически создав вспомогательный вектор. Используйте функции `zeros` для создания нулевого вектора той же длины, что x , и `max` с двумя выходными аргументами для поиска номера минимального элемента в векторе x .

3. Постройте диаграмму, где звездочки на горизонтальной оси будут отмечать центры интервалов.

4. Постройте графики функций $f(x) = e^{x/5} \cos^2 x$ и $g(x) = 8 \cdot e^{-x} \sin^2 x$ на отрезке $[-2\pi, 2\pi]$. Сгенерируйте вектор-строку значений аргумента x и вектор-строк f и g , содержащих значения функций.

5. Сравните две функции: $f(x) = \frac{1}{x^3}$ и $F(x) = 1000 \cdot (x + 0.2)^{-4}$.

6. Постройте графики функций $f(x) = \log 0.3x$ и $g(x) = \sin(\log x)$ на отрезке $[0.1, 5]$ в логарифмическом масштабе по оси x

7. Постройте график функции, заданной кусочным образом:

$$y(x) = \begin{cases} \pi \cdot \sin x, & -2\pi \leq x \leq -\pi; \\ \pi - |x|, & -\pi < x < \pi; \\ \pi \cdot \sin^3 x, & \pi \leq x \leq 2\pi. \end{cases}$$

Сначала необходимо вычислить каждую из трех ветвей, т.е. фактически получить три пары массивов: x_1 и y_1 , x_2 и y_2 , x_3 и y_3 , затем объединить значения абсцисс в вектор x , а значения ординат в y и построить график функции, задаваемой парой массивов x и y .

8. Постройте график функции

$$z(x, y) = 3 \cdot \sin 2\pi \cdot x \cdot \cos 1.3\pi \cdot y \cdot (1 - x^2) \cdot y \cdot (1 - y)$$

на прямоугольной области определения $x \in [-1, 1]$, $y \in [0, 1]$:

а) построить график с каркасной поверхностью (сделать ее «прозрачной» и вернуть в обычный вид);

б) залить каждую клетку поверхности определенным цветом, зависящим от значения функции в точках, соответствующих углам клетки;

в) получить поверхность, плавно залитую цветом;

г) вывести рядом с графиком столбик, устанавливающий соответствие между цветом и значением функции.

9. Постройте поверхность, состоящую из линий уровня, соответствующих значениям функции от 0 до 0.5 с шагом 0.01.

ЛАБОРАТОРНАЯ РАБОТА 9 М-ФАЙЛЫ. ЧИСЛЕННЫЕ МЕТОДЫ И ПРОГРАММИРОВАНИЕ

Цель работы: научиться работать с М-файлами.

Порядок работы

Предположим, что в вычислениях часто необходимо использовать функцию

$$e^{-x} \cdot \sqrt{\frac{x^2 + 1}{x^4 + 0.1}}.$$

Тогда имеет смысл один раз написать файл-функцию, а потом вызывать его всюду, где необходимо вычисление этой функции. Откройте в редакторе М-файлов новый файл и наберите текст:

```
function f = myfun (x)
f = exp (-x)*sqrt ((x^2+1)/(x^4+0.1));
```

Слово `function` в первой строке определяет, что данный файл содержит файл-функцию. Первая строка является заголовком функции, в которой размещается имя функции и списки входных и выходных аргументов. В примере, приведенном выше, имя функции `myfun` – один входной аргумент x и один выходной – f . После заголовка следует *тело функции* (оно в данном примере состоит из одной строки), где и вычисляется ее значение. Важно, что вычисленное значение записывается в f . Не забудьте поставить точку с запятой для предотвращения вывода лишней информации на экран.

Теперь сохраните файл в рабочем каталоге. Обратите внимание на то, что выбор пункта **Save** или **Save as** меню **File** приводит к появлению диалогового окна сохранения файла, в поле **File name** которого уже содержится название `myfun`. Не изменяйте его, сохраните файл-функцию в файле с предложенным именем!

Теперь созданную функцию можно использовать так же, как и встроенные `sin`, `cos` и другие, например из командной строки:

```
>> y = myfun (1.3)
y =
    0.2600
```

Файл-функция, приведенная выше, имеет один существенный недостаток. Попытка вычисления значений функции от массива приводит к ошибке, а не к массиву значений, так как это происходит при вычислении встроенных функций.

Необходимо просто при вычислении значения функции использовать поэлементные операции.

Измените тело функции на:

```
function f = myfun (x)
f = exp (-x).*sqrt ((x.^2+1)./(x.^4+1));
```

(не забудьте сохранить изменения в файле myfun.m).

Теперь аргументом функции myfun может быть как число, так и вектор или матрица значений, например:

```
>> x = [1.3 7.2];
>> y = myfun (x)
y =
    0.2600    0.0001
```

Переменная y, в которую записывается результат вызова функции myfun, автоматически становится вектором нужного размера.

Постройте график функции myfun на отрезке [0, 4] из командной строки или при помощи файл-программы:

```
x =[0 : 0.5 : 4];
y = myfun (x);
plot (x, y)
```

MatLab предоставляет еще одну возможность работы с файл-функциями – это использование их в качестве аргументов некоторых команд. Например, для построения графика служит специальная функция fplot, заменяющая последовательность команд, приведенную выше. При вызове fplot имя функции, график которой требуется построить, заключается в апострофы, пределы построения указываются в вектор-строке из двух элементов:

```
fplot ('myfun', [0 4])
```

Постройте графики myfun при помощи plot и fplot на одних осях, воспользовавшись функцией hold on. Обратите внимание, что график, построенный при помощи fplot, более точно отражает поведение функции, т.к. fplot сама подбирает шаг аргумента, уменьшая его на участках быстрого изменения отображаемой функции.

Нахождение корней произвольных уравнений осуществляет встроенная функция fzero, для определения всех корней полиномов применяется roots.

Встроенная функция fzero позволяет приближенно вычислить корень уравнения по заданному начальному приближению. В самом простом варианте fzero вызывается с двумя входными и одним выходным аргументом $x = fzero ('myf', x0)$, где myf – имя файл-функции, вычисляющей левую часть

уравнения; x_0 – начальное приближение к корню; x – найденное приближенное значение корня. Решите, например, на отрезке $[-5, 5]$ уравнение

$$\sin x - x^2 \cos x = 0 .$$

Перед нахождением корней полезно построить график функции, входящей в левую часть уравнения. Конечно, построить график можно при помощи `plot`, но все равно понадобится написать файл-функцию, поэтому имеет смысл воспользоваться `fplot`, которая к тому же позволяет получить более точный график по сравнению с `plot`.

Ниже приведен текст требуемой файл-функции:

```
function y = myf (x)
y = sin (x) - x.^2.*cos (x);
```

Теперь постройте график `myf`, используя `fplot`, и нанесите сетку:

```
fplot ('myf', [-5 5])
grid on
```

Из графика `myf` видно, что функция на этом отрезке имеет четыре корня. Один корень равен нулю, в чем нетрудно убедиться, подставив $x = 0$ в уравнение.

Уточните значение корня, расположенного вблизи $x = -5$, при помощи `fzero`:

```
>> x1 = fzero ('myf', -5)
Zero found in the interval: [-4.7172, -5.2].
x1 =
    -4.7566
```

Итак, приближенное значение корня равно -4.7566 . Проверьте ответ, вычислив значение функции `myf` в точке `x1`

```
>> myf (x1)
ans =
    2.6645e-015
```

Конечно, то, что значение функции близко к нулю, вообще говоря, не означает, что приближенное значение корня расположено достаточно близко к его точному значению.

Важной особенностью `fzero` является то, что она вычисляет только те корни, в которых функция меняет знак, а не касается оси абсцисс.

В качестве исследуемой функции может выступать и встроенная математическая функция, например:

```
>> fzero ('sin', [2 4])
```

Zero found the interval: [2, 4].

ans =

3.14159265358979

Полином в MatLab задается вектором его коэффициентов, например, для определения полинома:

$$p = x^7 + 3.2x^5 - 5.2x^4 + 0.5x^2 + x - 3$$

следует использовать команду

```
>> p = [1 0 3.2 -5.2 0 0.5 1 -3];
```

Число элементов вектора, т.е. число коэффициентов полинома, всегда на единицу больше его степени, нулевые коэффициенты должны содержаться в векторе.

Функция `polyval` предназначена для вычисления значения полинома от некоторого аргумента:

```
>> polyval(p, 1)
```

ans =

-2.5000

Аргумент может быть матрицей или вектором, в этом случае производится поэлементное вычисление значений полиномами результат представляет матрицу или вектор того же размера, что и аргумент.

Нахождение сразу всех корней полиномов осуществляется при помощи функции `roots`, в качестве аргумента которой указывается вектор с коэффициентами полинома. Функция `roots` возвращает вектор корней полинома, в том числе и комплексных:

```
>> r = roots(p)
```

r =

-0.5668 + 2.0698i

-0.5668 - 2.0698i

1.2149

0.5898 + 0.6435i

0.5898 - 0.6435i

-0.6305 + 0.5534i

-0.6305 - 0.5534i

Число корней полинома, как известно, совпадает со степенью полинома. Убедитесь в правильности работы `roots`, вычислив значение полинома от вектора его корней:

```
>> polyval(p, r)
```

ans =

1.0e-012 *
-0.1008 + 0.0899i
-0.1008 - 0.0899i
-0.0666
0.0027 - 0.0018i
0.0027 + 0.0018i
0.0102 - 0.0053i
0.0102 + 0.0053i

Обратите внимание, что в верхней строке результата содержится общий множитель $1.0e-012$, на который следует помножить каждое число получившегося вектора.

Задания

Вариант 1

1. Написать файл-функцию *myfun* (с тригонометрическими функциями, степенью, экспонентой, π и квадратным корнем).

Вычислить значение этой функции в нескольких точках.

Постройте график функции *myfun* при помощи *plot* и *fplot* на одних осях на отрезке $[0, 4]$ из командной строки или при помощи файл-программы.

2. Решите уравнение $6 \cdot \cos x + x^2 \cdot \sin x = 0$ на отрезке $[-10; 10]$. Постройте график функции. Вычислите значение корней уравнения.

3. Вычислите все корни полинома $p = x^8 + 6,8x^7 + 0,4x^5 + 2,1x^3 - x + 4$.

Вариант 2

1. Написать файл-функцию *myfun* (с тригонометрическими функциями, степенью, экспонентой, π и квадратным корнем).

Вычислить значение этой функции в нескольких точках.

Постройте график функции *myfun* при помощи *plot* и *fplot* на одних осях на отрезке $[0, 4]$ из командной строки или при помощи файл-программы.

2. Решите уравнение $4 \cdot \sin x + x^2 \cdot \cos x = 0$ на отрезке $[-10; 10]$. Постройте график функции. Вычислите значение корней уравнения.

3. Вычислите все корни полинома $p = x^8 + 1,8x^7 + 0,2x^5 + 3,5x^3 - x - 8$.

ЛАБОРАТОРНАЯ РАБОТА 10

ПРОГРАММИРОВАНИЕ. ОПЕРАТОРЫ ЦИКЛА

Цель работы: научиться использовать операторы цикла, как в файлах-процедурах, так и в файлах функциях.

Порядок работы

Выполнение схожих повторяющихся действий в MatLab осуществляется при помощи операторов циклов *for* и *while*. Цикл *for* предназначен для выполнения заданного числа повторяющихся действий, а *while* – для действий, число которых заранее неизвестно, но известно условие продолжения цикла.

Самое простое использование *for* осуществляется следующим образом:

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

Здесь *count* – переменная цикла, *start* – ее начальное значение, *final* – конечное значение, а *step* – шаг, на который увеличивается *count* – при каждом следующем заходе в цикл. Цикл заканчивается как только значение *count* становится больше *final*. Переменная цикла может принимать не только целые, но и вещественные значения любого знака.

Пусть требуется вывести семейство кривых для $x \in [0, 2\pi]$, которое задано функцией, зависящей от параметра $y(x, a) = e^{-ax} \sin x$, для значений параметра от -0.1 до 0.1 . Можно, конечно, последовательно вычислять $y(x, a)$ и строить ее графики для различных значений a от -0.1 до 0.1 , но гораздо удобнее использовать цикл *for*. Наберите текст файл-процедуры в редакторе М-файлов:

```
figure
x = [0 : pi / 30 : 2*pi];
for a = -0.1 : 0.02 : 0.1
    y = exp (-a*x) .* sin(x);
    hold on
    plot(x, y)
end
```

сохраните в файле FORdem1.m и запустите его на выполнение (или из редактора М-файлов, или из командной строки, набрав в ней команду FORdem1 и нажав <Enter>).

В результате выполнения FORdem1 появится графическое окно, которое содержит требуемое семейство кривых.

Напишите файл-программу для вычисления суммы:

$$S = \sum_{k=1}^{10} \frac{1}{k!}.$$

Алгоритм вычисления суммы использует накопление результата, то есть сначала сумма равна нулю, затем в переменную k заносится единица, вычисляется $1/k!$ (то есть $1/1!$), добавляется к S и результат снова заносится в S . Далее k увеличивается на единицу, и процесс продолжается пока последним слагаемым не станет $1/10!$.

Файл-программа FORdem2 вычисляет искомую сумму:

```
% ФАЙЛ-ПРОГРАММА ДЛЯ ВЫЧИСЛЕНИЯ СУММЫ
% 1/1! + 1/2! + ... + 1/10!

% обнуление S для накопления суммы
S = 0;
% накопление суммы в цикле
for k = 1 : 10
    S = S + 1 / factorial (k);
end
% вывод результата в командное окно
s
```

Если шаг цикла равен 1, то его можно не указывать.

Наберите файл-программу в редакторе М-файлов, сохраните в текущем каталоге в файле FORdem.m и выполните ее. Результат отображается в командном окне, т.к. в последней строке файл-программы содержится S без точки с запятой для вывода значения переменной S:

```
S =
    1.7183
```

Обратите внимание, что остальные строки файл-программы, которые могли бы повлечь вывод на экран промежуточных значений, завершаются точкой с запятой для подавления вывода в командное окно.

Первые две строки с комментариями не случайно отделены пустой строкой от остального текста программы. Именно они выводятся на экран, когда пользователь при помощи команды help из командной строки получает информацию о том, что делает FORdem2.

Цикл for оказывается полезным при выполнении повторяющихся похожих действий в том случае, когда их число заранее определено. Обойти это ограничение позволяет более гибкий цикл while.

Требуется найти сумму для заданного x (разложение в ряд $\sin x$):

$$S(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

Конечно, до бесконечности суммировать не удастся, но можно накапливать сумму, пока слагаемые являются не слишком маленькими, скажем больше 10^{-10} (по модулю). Циклом for здесь не обойтись, т.к. заранее неизвестно значение k , обеспечивающее малость текущего слагаемого. Выход состоит в применении цикла while, который работает, пока выполняется *условие цикла*:

```
while условие цикла
    команды MatLab
end
```

В данном примере условием цикла является то, что текущее слагаемое $x^k / k!$ больше 10^{-10} . Для записи условия в формате, понятной MatLab, следует использовать знак больше (>).

Текст файл-функции mysin, вычисляющей сумму ряда, выглядит так:

```
function s = mysin (x)
% Вычисление синуса разложением в ряд
% Использование: y = mysin (x), -pi<x<pi
s = 0;
k = 0;
while abs (x .^ (2*k+1) / factorial (2*k+1)) > 1.0e-10
    s = s + (-1) ^k*x. ^ (2*k+1) / factorial (2*k+1);
    k = k + 1;
end
```

Обратите внимание, что у цикла while, в отличие от for, нет переменной цикла, поэтому пришлось до начала цикла k присвоить ноль, а внутри цикла увеличивать k на единицу.

Сравните теперь результат, построив графики функций mysin и sin на отрезке $[-\pi, \pi]$ на одних осях, например, при помощи fplot (команды можно задать из командной строки):

```
>> fplot ('mysin', [-pi, pi])
>> hold on
>> fplot ('sin', [-pi, pi], 'k.')
```

Задания

Вариант 1

1. Напишите файл-программу и файл-функцию для вычисления суммы

$$S = \sum_{k=1}^{10} \frac{e^k}{x^k}.$$

2. Напишите файл-программу для вычисления выражения

$$Z_{x,y} = \frac{x + y - 1}{x - y}$$

для $x = 2, 3, \dots, n + 1$ и $y = 1, 2, \dots, n$.

3. а) Напишите файл-функции для нахождения суммы ряда для заданного x (разложение в ряд $\cos x$ и e^x):

$$S(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} \quad \text{и} \quad S(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

соответственно, пока слагаемые суммы больше 10^{-10} .

б) сравните полученные результаты, построив графики функций mycos и \cos на отрезке $[-\pi, \pi]$ на одних осях, и myexp и \exp , на отрезке $[-2\pi, 2\pi]$.

Вариант 2

1. Напишите файл-программу и файл-функцию для вычисления суммы

$$S = \sum_{k=1}^{20} \frac{e^k}{x^k}.$$

2. Напишите файл-программу для вычисления выражения

$$Z_{x,y} = \frac{y + x - 1}{y - x}$$

для $x = 1, 2, \dots, n$ и $y = 2, 3, \dots, n + 1$.

3. а) Напишите файл-функции для нахождения суммы ряда для заданного x (разложение в ряд $\cos x$ и e^x):

$$S(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} \quad \text{и} \quad S(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

соответственно, пока слагаемые суммы больше 10^{-10} .

б) сравните полученные результаты, построив графики функций mycos и \cos на отрезке $[-\pi, \pi]$ на одних осях, и myexp и \exp , на отрезке $[-2\pi, 2\pi]$.

ЛАБОРАТОРНАЯ РАБОТА 11

ПРОГРАММИРОВАНИЕ. ОПЕРАТОРЫ ВЕТВЛЕНИЯ. ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ

Цель работы: научиться использовать операторы ветвления и прерывание цикла в исключительных ситуациях.

Порядок работы

Условный оператор `if` и оператор переключения `switch` позволяют создать гибкий разветвляющийся алгоритм выполнения команд, в котором при выполнении определенных условий работает соответствующий блок операторов или команд MatLab. Практически во всех языках программирования имеются аналогичные операторы.

Оператор `if` может применяться в простом виде для выполнения блока команд при удовлетворении некоторого условия или в конструкции `if-elseif-else` для написания разветвляющихся алгоритмов.

Начнем с простейшего примера, то есть файл-функции для вычисления выражения $\sqrt{x^2 - 1}$.

Создание файл-функции не должно вызвать затруднений. Она работает для любых значений x , причем для $-1 < x < 1$ результат является комплексным числом. Предположим, что вычисления происходят в области действительных чисел и требуется вывести предупреждение о том, что результат является комплексным числом. Перед вычислением функции следует произвести проверку значения аргумента x и вывести в командное окно предупреждение, если модуль x не превосходит единицы. Здесь уже не обойтись без условного оператора `if`, применение которого в самом простом случае выглядит так:

```
if условие
    команды MatLab
end
```

Если условие верно, то выполняются команды MatLab, размещенные между `if` и `end`, а если условие неверно, то происходит переход к командам, расположенным после `end`.

Команда `warning` служит для вывода предупреждения в командное окно.

Файл-функция, проверяющая значение аргумента, выглядит так:

```
function f = Rfun (x);
% вычисляет sqrt (x^2-1)
% выводит предупреждение, если результат комплексный
```

```
% использование y = Rfun (x)

% проверка аргумента
if abs (x) < 1
    warning ('результат комплексный')
end

% вычисление функции
f = sqrt (x^2 - 1);
```

Теперь вызов Rfun от аргумента, меньшего единицы по модулю, приведет к выводу в командное окно предупреждения:

```
>> y = Rfun (0.2)
результат комплексный
y =
    0 + 0.97979589711327i
```

Файл-функция Rfun только предупреждает о том, что ее значение комплексное, все вычисления с ней продолжаются. Если же комплексный результат означает ошибку вычислений, то следует прекратить выполнение функции, используя error вместо warning.

Напишите файл-функцию root2, которая по коэффициентам квадратного уравнения находит только вещественные его корни, а для комплексных выдает ошибку.

При составлении файл-функций следует предусмотреть еще один вид контроля – проверку количества входных и выходных параметров. Если пользователь вызовет функцию root2 с двумя входными параметрами, то получит сообщение об ошибке при выполнении того оператора файл-функции, который содержит неопределенный параметр. В случае вызова функции root2 с одним выходным аргументом или без аргументов будет вычислен только первый корень квадратного уравнения, что также введет пользователя в заблуждение. Лучше заранее предупредить пользователя о характере ошибки и прекратить работу файл-функции. Кроме того, следует учесть, что файл-функция root2 не может принимать массивы в качестве входных аргументов. Если даже использовать поэлементные операции при вычислениях, то дискриминант уравнения D будет массивом, а что такое $D > 0$ для массива – пока неизвестно.

Дополните функцию root2 вышеописанными видами контроля, предотвращающими неправильное ее использование. Встроенные функции nargin и nargout возвращают число входных и выходных аргументов соответственно. Для проверки, являются ли входные аргументы числами, следует сначала найти размеры соответствующих переменных при помощи size, а затем про-

верить их на равенство единице. Вывод текста в командное окно в ходе выполнения файл-программы или файл-функции осуществляется оператором `disp`, сам текст указывается в апострофах `disp ('текст')`:

```
function [x1 x2] = root2(a, b, c);
if nargin ~= 3
    error('Введите 3 входных аргумента!')
end
if narginout ~= 2
    error('Введите 2 выходных аргумента!')
end
[na ma] = size(a);
[nb mb] = size(b);
[nc mc] = size(c);
if na ~= 1 | ma ~= 1 | nb ~= 1 | mb ~= 1 | nc ~= 1 | mc ~= 1
    error('Аргументы должны быть числами, а не массивами!')
end
if real(a) ~= a | real(b) ~= b | real(c) ~= c
    error('Аргументы должны быть действительными!');
end
d = b^2 - 4*a*c;
if d < 0
    error('Уравнение не имеет решения в области действительных чисел!');
end
x1 = (-b - sqrt(d)) / (2*a);
x2 = (-b + sqrt(d)) / (2*a);
```

В общем виде оператор ветвления представляет конструкцию `if-elseif-else`, работу которой хорошо поясняет пример файл-функции `ifdem`:

```
function ifdem (a);
% Пример использования структуры if-elseif-else
if (a == 0)
    disp ('а равно нулю')
elseif a == 1
    disp ('а равно единице')
elseif a == 2
    disp ('а равно двум')
elseif a >= 3
    disp ('а больше или равно трем')
```

```

else
    disp ('a меньше трех, но не ноль, не единица и не двойка')
end

```

В зависимости от выполнения того или иного условия работает соответствующая ветвь программы, если все условия неверны, то выполняются команды, размещенные после else. Вызовы функции ifdem с различными аргументами позволяют убедиться в вышесказанном:

```

>> ifdem (1)
a равно единице
>> ifdem (1.2)
a меньше трех, но не ноль, не единица и не двойка
>> ifdem (2)
a равно двум
>> ifdem (3)
a больше или равно трем
>> ifdem (-1)
a меньше трех, но не ноль, не единица и не двойка

```

Ветвей может быть сколько угодно (добавьте несколько elseif с похожими условиями в ifdem) или только две, например:

```

if (a == 0)
    disp ('a равно нулю')
else
    disp ('a не равно нулю')
end

```

В случае двух ветвей используется завершающее else, а elseif пропускается. Оператор if должен заканчиваться end.

Файл-функция ifdem хорошо демонстрирует работу оператора if, но на практике оказывается бесполезной. Действительно, полезный пример – вычисление кусочно-заданной функции

$$y(x) = \begin{cases} \sin x, & x < -\pi; \\ x / \pi, & -\pi \leq x < \pi; \\ -\cos x, & x \geq \pi, \end{cases}$$

которая реализуется файл-функцией rpwfun.

Ее текст приведен ниже:

```

function y = rpwfun (x);
% вычисляет кусочно-линейную функцию

```

```

% sin(x) -1,      если x < -pi
% y(x) =         x,      если -pi <= x < 0
% pi*cos(x),     если x >= 0
% использование y = rpwfun(x), x – число;
if x < -pi
    y = sin (x) -1
elseif x < pi % проверка x > -pi не нужна!
    y = x / pi;
else          % здесь x > pi
    y = -cos (x);
end

```

Обратите внимание на следующие моменты:

число ветвей if–elseif–else равно трем

Во второй ветви достаточно только проверить, что $x < \pi$, а условие $x \geq -\pi$ уже выполнено (иначе бы отработала первая ветвь в if–elseif–else и оператор if закончил работу).

В последней ветви нет смысла проверять какие бы то ни было условия, она работает, если все предыдущие условия неверны, что как раз соответствует $x \geq \pi$.

Для построения графика кусочно-заданной функции rpwfun следует воспользоваться командой fplot ('rpwfun', [-3*pi, 3*pi]).

Построение графика rpwfun не случайно осуществляется при помощи fplot. Функцией plot воспользоваться не удастся, т.к. требуется предварительно вычислить вектор значений функции от вектора аргументов, а rpwfun не умеет работать с входным *аргументом-вектором*. Убедиться в этом можно, построив график rpwfun командами:

```

x = [-3*pi : 0.1 : 3*pi];
y = rpwfun(x);
plot (x, y)

```

Никакой ошибки при выполнении файл-функции не возникает, однако график строится неправильно. Дело в том, что вектор x входит в условия оператора if. Операции отношения (<, <=, >, >=, ~=) могут *специальным образом* применяться и к векторам, а при обычном применении в данном примере не дают ожидаемого результата. Выход состоит в применении следующего алгоритма вычисления кусочно-заданной функции, для реализации которого достаточно понимания вышеописанного материала:

1. Проверка числа входных аргументов, если число входных аргументов не равно единице, то завершение работы файл-функции с сообщением об ошибке (выход по ошибке).

2. Проверка, является ли входной аргумент вектором или числом, один из размеров входного аргумента должен быть равен единице. Если это условие не выполняется, то выход по ошибке.

3. Нахождение длины входного аргумента.

4. Создание вектора выходного аргумента того же размера, что и входной аргумент и заполнение его нулями.

5. Перебор всех элементов входного вектора с использованием цикла `for`, вычисление от них значений кусочно-заданной функции и запись в соответствующие элементы выходного вектора.

Попробуйте составить файл-функцию `rwfun1` самостоятельно (входной аргумент – число или вектор).

Входными аргументами файл-функции `rwfun1` могут быть как число, так и вектор, причем, если входной аргумент является вектор-строкой (вектор-столбцом), то результат тоже вектор-строка (вектор-столбец).

В качестве завершающего упражнения попробуйте улучшить `rwfun1` (создав новую файл-функцию `rwfun2`) так, чтобы ее входным аргументом могла быть и матрица. Учтите, что вектор в MatLab так же, как и матрица, является двумерным массивом, у которого один из размеров равен единице. Очевидно, что для перебора элементов входного аргумента придется использовать вложенные циклы `for`.

Предположим, что при работе с функцией двух аргументов

$$e^{-|x \cdot y|} \sin \pi x \cdot \cos \pi x^2,$$

тогда приходится визуализировать ее четырьмя различными способами: каркасной поверхностью, сплошной поверхностью, выводом линии уровня и строением освещенной поверхности. Удобно создать файл-функцию, один из входных аргументов которой `vis` будет определять способ визуализации. Если `vis` равен единице, то строится каркасная поверхность, для `vis`, равного двум, – сплошная и т.д. Можно, конечно, использовать оператор `if` в полном виде `if-elseif-else`, который в зависимости от значения `vis` выполняет нужную ветвь программы, выводящую соответствующий график. Однако оператор переключения `switch` позволяет написать более наглядную программу. Применение `switch` поясняет следующий фрагмент:

```
switch a
    case -1
        disp('a = -1')
    case 0
        disp('a = 0')
    case (2, 3, 4)
        disp('a равно 2 или 3 или 4')
    otherwise
        disp('a не равно -1, 0, 1')
end
```

Каждая ветвь определяется оператором `case`. Переход в нее выполняется тогда, когда переменная оператора `switch` (в данном примере `a`) принимает значение, указанное после `case`, или одно из значений из списка `case`. После выполнения какой-либо из ветвей происходит выход из `switch`, при этом значения, заданные в других `case`, уже *не повторяются*. Если подходящих значений для `a` не нашлось, то выполняется ветвь программы, соответствующая `otherwise`.

Оператор `switch` как нельзя лучше подходит для решения поставленной задачи о выводе различных графиков исследуемой функции. Попробуйте написать файл-функцию (`myplot3D`) самостоятельно. Входными аргументами являются границы построения исследуемой функции по каждой из переменных `xmin`, `xmax`, `umin`, `umax` и способ построения графика, определяемый `vis`. Все пять входных аргументов должны быть числами, причем `xmin` меньше `xmax` и `umin` меньше `umax`. Не забудьте сделать соответствующую проверку! Выходные аргументы в данном случае не требуются.

Оператор `switch` удобно применять тогда, когда есть соответствие между *дискретными* значениями некоторой переменной и последующими действиями.

А теперь попробуйте сами улучшить функцию `rwfun1`, чтобы входными аргументами могло быть число, вектор-строка, вектор-столбец и матрица.

ЛАБОРАТОРНАЯ РАБОТА 12

ПРОГРАММИРОВАНИЕ ПРЕРЫВАНИЯ ЦИКЛА, ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ

Цель работы: научиться использовать прерывание цикла в исключительных ситуациях.

Порядок работы

Хорошо написанная программа предотвращает ошибочные действия, которые приводят к досрочному ее завершению. Проверка входных и выходных аргументов файл-функции является одним из способов контроля. MatLab предоставляет в распоряжение программиста еще два средства: прерывание цикла и обработку исключительных ситуаций.

При организации циклических вычислений следует позаботиться о том, чтобы внутри цикла не возникало ошибок. Например, пусть дан массив x , состоящий из целых чисел и требуется сформировать новый массив y по правилу $y(i) = x(i) / x(i + 1)$. Очевидно, что задача может быть решена при помощи цикла `for`. Но если один из элементов исходного массива равен нулю, то при делении получится `Inf` и последующие вычисления могут оказаться бесполезны. Предотвратить подобную ситуацию можно выходом из цикла, если текущее значение $x(i)$ равно нулю. Следующий фрагмент программы демонстрирует использование оператора `break` для прерывания цикла:

```
...
y = zeros(length(x) -1)
for l = 1 : length(x) -1
    if x(i) == 0
        break
    end
    y(i) = x(i + 1) / x(i);
end
...
```

При выполнении условия $x(i) == 0$ оператор `break` заканчивает цикл и происходит выполнение операторов, которые расположены в строках, следующих за `end`. Оператор `break` можно использовать и с циклом `while`. В случае вложенных циклов `break` осуществляет выход из внутреннего цикла.

Часть некорректных математических операций в MatLab, в отличие от многих языков программирования, не приводит к завершению работы программы. При делении на ноль получается бесконечность `Inf`, деление ноля на

ноль приводит к NaN, сумма бесконечности и числа имеет результатом бесконечность. Однако есть ошибки, приводящие к окончанию работы программы. К таким ошибкам относится, например, работа с несуществующими файлами. Предположим, что в процессе выполнения программы следует считать в переменную данные из файла, преобразовать их и отобразить в виде круговой диаграммы, а затем продолжить некоторые вычисления, которые не связаны со считанными данными. Последовательность операторов, соответствующая требуемым действиям, приведена ниже:

```
A = load ('my.dat');  
pie(A)  
X = [1 2 -1 -2];  
X = X.^2
```

Если MatLab обнаруживает файл my.dat в путях поиска и считывает данные из него, то этот фрагмент программы работает успешно.

Однако, если файл найти не удалось или при чтении из него возникли ошибки, то MatLab выведет сообщение в командное окно и закончит выполнение программы. Выходом из подобных ситуаций является конструкция try...catch, позволяющая обойти исключительные ситуации, которые приводят к ошибке, и предпринять некоторые действия в случае их возникновения. Схема использования try...catch выглядит следующим образом:

```
try  
    % операторы, выполнение которых  
    % может привести к ошибке  
catch  
    % операторы, которые следует выполнить  
    % при возникновении ошибки в блоке  
    % между try и catch  
end
```

Фрагмент программы, приведенный выше, лучше оформить с использованием try и catch так, как это сделано ниже:

```
try  
    A = load ('my.dat');  
    pie(A)  
catch  
    disp ('не могу найти файл my.dat')
```

```
end
X = [1; 2; -1; -2];
X = X.^2
```

Теперь при отсутствии нужного файла `my.dat` программа выдаст сообщение об этом и продолжит работу:

```
>> не могу найти файл my.dat
X =
    1     4     1     4
```

А теперь, зная операторы цикла, ветвления, прерывание цикла в исключительных ситуациях, попробуйте запрограммировать следующие задания, при этом, используя всевозможные проверки и не пренебрегая комментариями.

Задание

1. Используя оператор цикла, вычислите значения синуса при 21 значении аргумента от 0,2 до 4 с шагом 0,2.
2. Создайте матрицу произвольного размера. Заполните случайными числами. Найдите минимальный и максимальный элемент матрицы.
3. Задать массив целых положительных чисел и определить, сколько в нем простых чисел. Если простые числа найдены, то вывести их количество и сами эти числа.
4. Написать рекурсивную функцию *fact* для вычисления факториала числа N :

$$N! = \begin{cases} 1, & \text{если } N = 0; \\ N \times (N-1)!, & \text{если } N > 0. \end{cases}$$

5. Написать программу, которая будет выводить на экране пилообразную функцию.

ЛАБОРАТОРНАЯ РАБОТА 13

ОСНОВНЫЕ ВОЗМОЖНОСТИ ПАКЕТА SIMULINK. ИНТЕРФЕЙС БРАУЗЕРА БИБЛИОТЕК. ИНТЕРФЕЙС ОКНА МОДЕЛЕЙ SIMULINK. СОЗДАНИЕ МОДЕЛИ. ОСНОВНЫЕ ПРИЕМЫ ПОДГОТОВКИ И РЕДАКТИРОВАНИЯ МОДЕЛИ

Цель работы: научиться работать в Simulink.

Порядок работы

В состав системы MatLab входит пакет моделирования динамических систем – Simulink.

Основными блоками будут генератор синусоидальных сигналов и блок нелинейности, моделирующий передаточную характеристику ограничителя. Кроме того, к этим блокам надо добавить регистрирующий блок – осциллограф. Так как функциональная схема моделируемого устройства в данном случае вполне очевидна, то мы можем перейти к ее реализации.

Для создания модели данного устройства сделайте следующие действия:

1. Откроем окно новой модели Simulink.
2. Расположим это окно рядом с окном браузера библиотек.
3. Из раздела библиотеки Sources перенесем в окно модели источник синусоидального сигнала Sine Wave.
4. Из раздела библиотеки Nonlinear перенесем в окно модели нелинейный блок – ограничитель Saturation.
5. Из раздела библиотеки Sinks перенесем в окно модели блок осциллографа Scope.
6. Выполним соединение между блоками.
7. Проверим установку времени моделирования: Start time = 0 и Stop time = 20.
8. Щелкнув дважды по блоку Sine Wave, в появившемся окне параметров источника синусоидального сигнала установим амплитуду, равную 5.
9. Запустим модель на исполнение, нажав кнопку Start Simulation в панели инструментов окна модели.

Нетрудно заметить, что масштаб отображения осциллограммы у осциллографа оказался не совсем удачным: изображение осциллограммы мало по высоте, поскольку при порогах в 0.5 масштаб в 5 условных единиц уровня получается слишком крупным. Заметим, что мы не указываем размерность

осциллограммы по вертикали. В зависимости от условий задачи это могут быть метры (задача на движение), вольты (электронный ограничитель) и т.д.

Для смены масштаба достаточно щелкнуть правой кнопкой мыши в окне осциллограммы. В появившемся контекстном меню нужно выбрать команду Axes Properties ..., которая служит для задания масштаба осциллограммы.

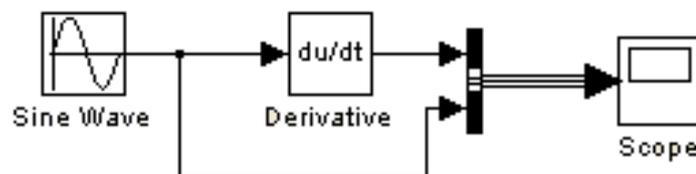
В открывшемся окне свойств осей надо заменить значения $Y\text{-min} = -5$ и $Y\text{-max} = 5$, например, на $Y\text{-min} = -0.8$ и $Y\text{-max} = 0.8$. Нетрудно заметить, что теперь осциллограмма стала гораздо более представительной.

В контекстном меню осциллограммы видна еще команда – Autoscale (Автомасштабирование). Эта же команда реализуется кнопкой Autoscale в панели инструментов окна осциллограммы. Эта команда устанавливает масштаб, при котором окно осциллограммы используется полностью. В данном случае это означает, что осциллограмма будет иметь максимально возможный размер.

Итак, как и следовало ожидать, в результате моделирования получена синусоида с обрезанными на уровне 0.5 сверху и снизу вершинами. При этом результат получен мгновенно (см. данные в строке состояния окна модели (время моделирования – 0). Столь быстрое получение явно верного результата достигается далеко не всегда. Чем сложнее модель, тем больше усилий и времени придется затратить на то, чтобы добиться ее правильной работы.

Блоки моделей обычно имеют входы и выходы. Как правило, выход какого-либо блока подключается ко входу следующего блока и т.п. Для этого курсор мыши устанавливается на выходе блока, от которого должно исходить соединение. При этом курсор превращается в большой крестик из тонких линий. Держа нажатой левую кнопку мыши, надо плавно переместить курсор ко входу следующего блока, где курсор мыши приобретет вид крестика из тонких сдвоенных линий. Добившись протяжки линии ко входу следующего блока, надо отпустить левую кнопку мыши. Соединение будет завершено, и в конце его появится жирная стрелка.

Для примера создайте модель дифференцирующего устройства:



Внимательно присмотревшись к осциллограммам, мы видим, что при входном синусоидальном сигнале выходной сигнал является косинусоидой. Это вполне отвечает математическим соотношениям для данного случая (как известно, производная $\sin(x)$ есть $\cos(x)$).

Однако в самом начале процесса дифференцирования хорошо будет виден изъян работы модели: при $t = 0$ производная равна не 1, а 0. Это связано с тем, что процесс начинается при нулевых начальных условиях. Но довольно быстро ситуация исправляется и в дальнейшем выходной сигнал становится косинусоидальным. Таким образом, дифференцирующее устройство можно использовать для точного сдвига на 90° гармонического сигнала.

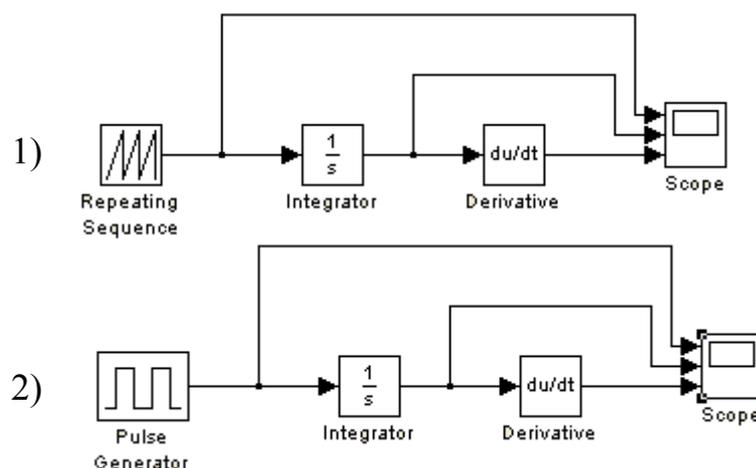
Виртуальный осциллограф – самое важное из регистрирующих устройств. Он позволяет представить результаты моделирования в виде временных диаграмм тех или иных процессов.

Окно параметров осциллографа содержит:

- Number of axes – число осей (каналов) осциллографа;
- Time range – пределы временного интервала;
- Tick labels – вывод / скрытие отметок по осям;
- Sampling – установка временных соотношений: Decimation (в десятичных долях времени со значением по умолчанию 1) или Sample Time (в тактах эталонного времени, по умолчанию 0).

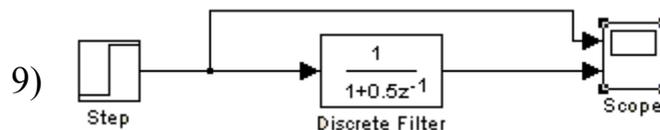
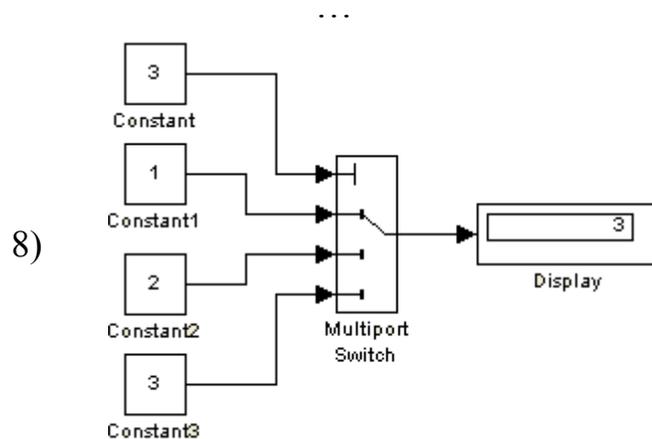
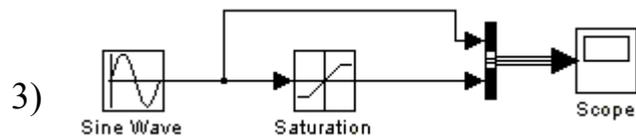
Параметр Numbers of axes позволяет превратить одноканальный осциллограф в многоканальный.

Для примера создайте модель: 1) интегрирования пилообразных импульсов с последующим дифференцированием; 2) интегрирования прямоугольных импульсов с последующим дифференцированием:

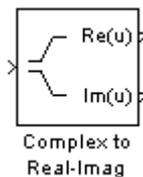


а также создайте модели с использованием: 3) ограничения (Saturation); 4) блока с зоной нечувствительности (Dead Zone); 5) релейного блока (Relay); 6) блока ограничения скорости (Rate Limiter); 7) блока следящего квантования (Quantizer); 8) блока многовходового переключателя (Multiport Switch); 9) блока дискретного фильтра (Discrete Filter); 10) попробуйте создать что-нибудь свое.

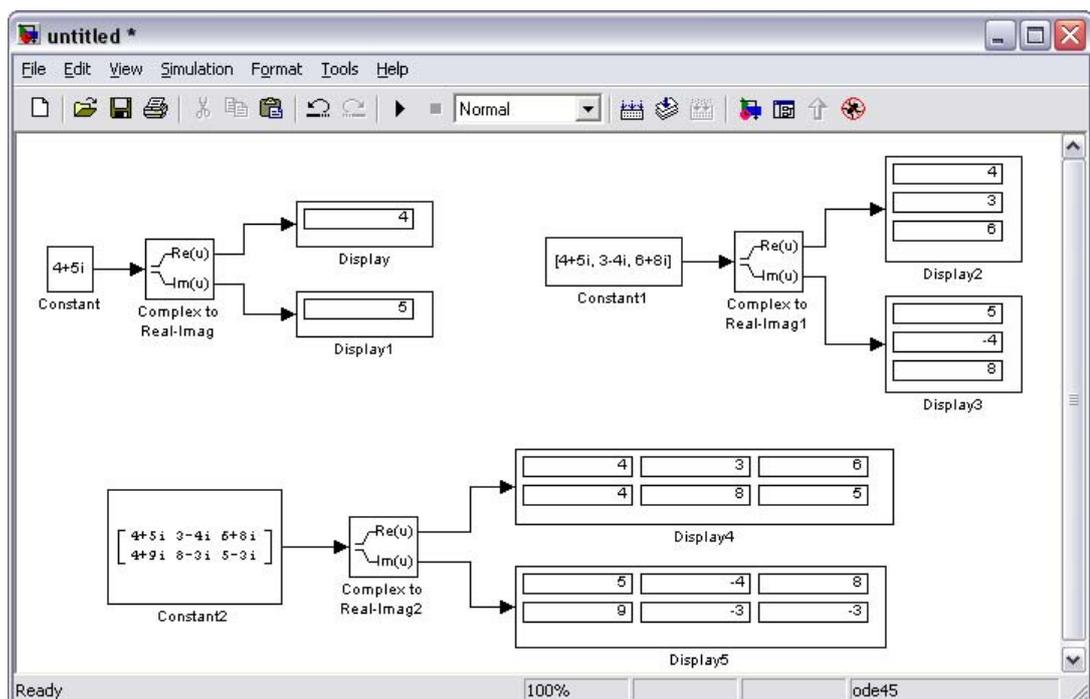
Например:



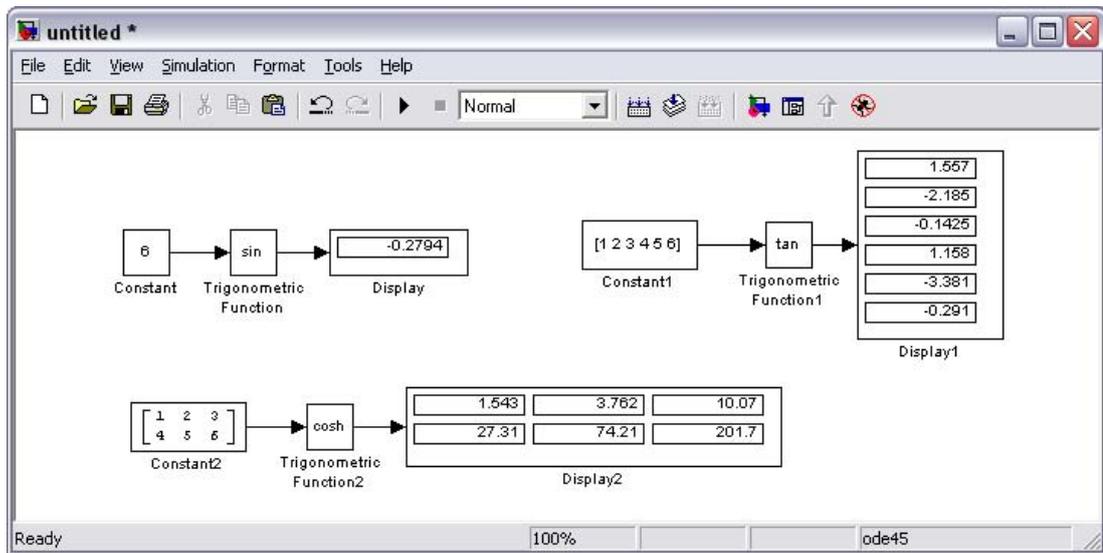
Чтобы разложить комплексное число на действительную и мнимую часть можно воспользоваться блоком Complex to Real-Imag:



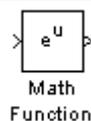
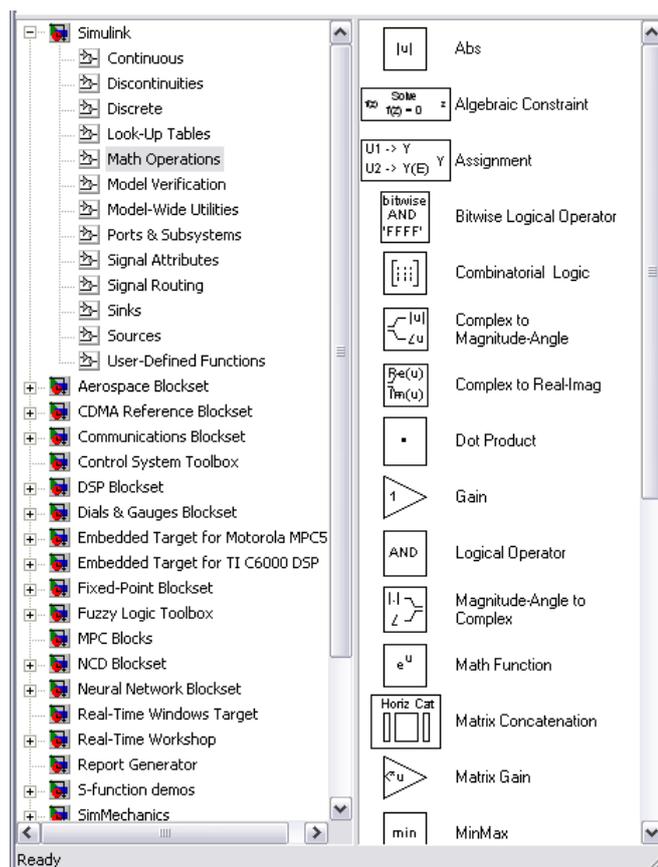
Разложить можно как число, так и вектор или даже матрицу:



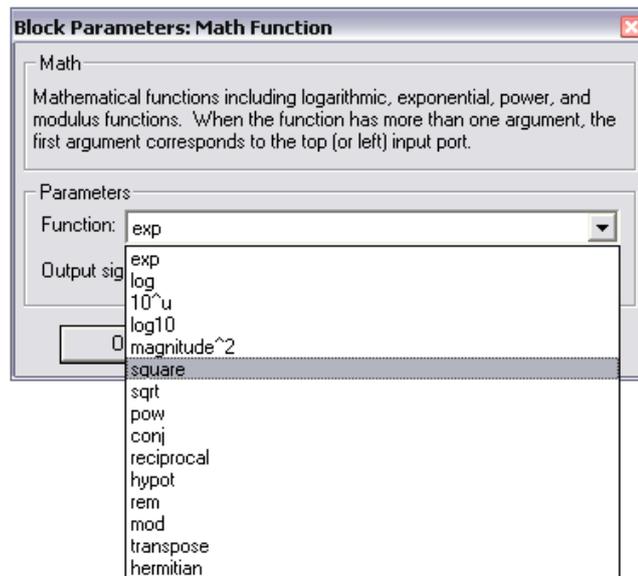
Тригонометрические функции в блоке Trigonometric Function выглядят так:



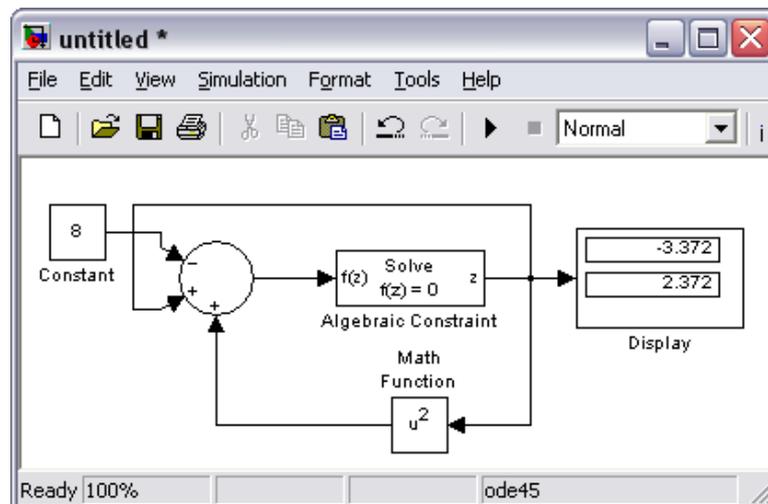
Математические операции находятся в раскрывающемся списке Simulink -> Math Operatons:



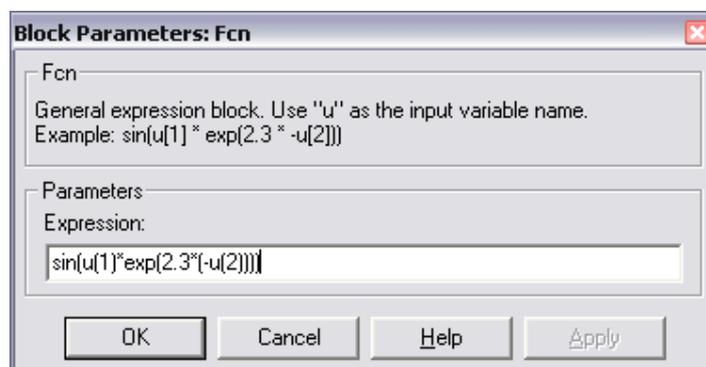
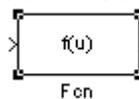
Там в математических операциях находится блок MathFunction, в котором содержатся многие математические функции:



Чтобы найти корни алгебраического уравнения, например, $x^2 + x - 8 = 0$, необходимо воспользоваться блоком Algebraic Constraint:



Также в Simulink есть блок для задания функции



Задания

1. Комплексные числа разложить на действительную и мнимую части и вывести результат. А также вектор и матрицу разложить на действительную и мнимую частоты.
2. Вычислить значения тригонометрической функции (\sin , \tan , \arccos , \cosh) для числа, вектора и матрицы и вывести результат.
3. Сложить, перемножить два вектора, две матрицы и вывести результат.
4. Вычислить e^u , u^x , u^T , $\log u$, \sqrt{u} , если u – число, вектор, матрица.
5. Ввести вектор-строку и вектор-столбец и найти \min и \max элемент.
6. Найти корни алгебраического выражения $x^2 + x = 9$.
7. Вычислить значение функции $\cos(u1) + \sin(u2) - 0.5 \exp(-u3)$, где u – вектор-строка, состоящая из $u1$, $u2$, $u3$.
8. Решить систему нелинейных алгебраических выражений

$$\left. \begin{array}{l} x^2 + y^2 = 6 \\ x + y = 2 \end{array} \right\}$$

ЛАБОРАТОРНАЯ РАБОТА 14

НАЗНАЧЕНИЕ, ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ И ИСПОЛЬЗОВАНИЯ MATHCAD

Цель работы: посмотреть на слайдах внешний вид MathCAD, принципы функционирования и использования MathCAD.

Теоретическое сведение

Для автоматизации математических расчетов используются разнообразные вычислительные средства – от программируемых микрокалькуляторов до сверхмощных суперЭВМ. И тем не менее такие расчеты остаются сложным делом. Более того, применение компьютеров внесло новые трудности: прежде чем начать расчеты, пользователь должен освоить основы программирования, изучить один или несколько языков программирования и численные методы расчетов. Положение стало меняться после специализированных программных комплексов для автоматизации математических и инженерно-технических расчетов. К таким комплексам относятся пакеты программ MathCAD, MatLab, Eureka. MathCAD – один из наиболее мощных и универсальных из них.

MathCAD является интегрированной системой программирования, ориентированной на проведение математических и инженерно-технических расчетов. Система MathCAD содержит текстовый редактор, вычислитель и графический процессор.

Текстовый редактор служит для ввода и редактирования текстов. Тексты являются комментарии. Входящие в них математические выражения не выполняются. Текст может состоять из слов, математических выражений и формул, спецзнаков. Отличительная черта системы – использование общепринятой в математике символики (деление, умножение, квадратный корень).

Вычислитель обеспечивает вычисление по сложным математическим формулам, имеет большой набор встроенных математических функций, позволяет вычислять ряды, суммы, произведения, определенный интеграл, производные, работать с комплексными числами, решать линейные и нелинейные уравнения, проводить минимизацию функции, выполнять векторные и матричные операции и т.д. Легко можно менять разрядность чисел и погрешность интеграционных методов.

Графический процессор служит для создания графиков. Он сочетает простоту общения с пользователем с большими возможностями графических средств. Графика ориентирована на решение типичных математических задач. Возможно быстрое изменение размеров графиков, наложение их на текстовые надписи и перемещение их в любое место документа.

MathCAD ориентирован на IBM-совместимые ПК, работает с PC-DOS и MS-DOS с версиями 2.0 и выше. Поддерживаются основные типы адапте-

ров и дисплеев. Минимальная память – 512 Кб. MathCAD автоматически поддерживает работу с математическим процессором. Последний заметно повышает скорость расчетов и вывода графиков, что существенно в связи с тем, что MathCAD всегда работает в графическом режиме. Это связано с тем, что только в этом режиме можно формировать на экране специальные математические символы и одновременно применять их вместе с графиками и текстом. MathCAD поддерживает работу со многими типами принтеров, а также с плоттерами.

MathCAD – система универсальная, то есть она может использоваться в любой области науки и техники, везде, где применяются математические методы. Запись команд в системе MathCAD на языке, очень близком к стандартному языку математических расчетов, упрощает постановку и решение задач.

К важным достоинствам новых версий MathCAD относятся: настройка под любой мало-мальски известный тип печатающих устройств, богатый набор шрифтов, возможность использования всех инструментов Windows, прекрасная графика и современный многооконный интерфейс. В новые версии MathCAD включены эффективные средства оформления документов в цвете, возможность создания анимированных (движущихся) графиков и звукового сопровождения, тут же текстовый, формульный и графический редакторы, объединенные с мощным вычислительным потенциалом. Предусмотрена и возможность объединения с другими математическими и графическими системами для решения особо сложных задач. Отсюда и название таких систем – *интегрированные* системы.

Впрочем, в решении задач интеграции создатели MathCAD пошли намного дальше. Эта система обеспечивает подлинную интеграцию с целым рядом других математических, графических и офисных систем. Для этого в нее включен специальный системный интегратор MathConnex. Летом 1999 г. выпущена новейшая версия системы – MathCAD 2000. В ней существенно увеличено число встроенных функций, улучшены графические возможности, повышены скорость вычислений и удобство работы.

Уже более 10 лет фирма MathSoft – производитель MathCAD – вносит вклад в технические новации в широком спектре рынков и индустрий, включая военную и аэрокосмическую отрасли, телекоммуникации, электронное и электрическое оборудование, гражданскую инженерию и конструирование, технические услуги и высшее и специальное образование. MathCAD является наиболее широко используемым программным обеспечением прикладной математики и поддерживает среду технических вычислений, оптимизированную по технической эффективности и новациям. MathCAD включает множество операторов, встроенных функций и алгоритмов решения разнообразных математических задач. Программа обладает повышенной точностью и быстродействием вычислений. Вывод формул в естественном виде, интерактивный режим работы, рациональная организация рабочего пространства (запросы, графики, данные отображаются на одном рабочем листе), поддержка символьной математики – все это делает систему MathCAD достойным продуктом.

ЛАБОРАТОРНАЯ РАБОТА 15

РАБОТА С УСТАНОВКАМИ MATHCAD

Цель работы: научиться работать с установками MathCAD.

Порядок работы

1. Установка параметров страницы.

Для этого нужно открыть меню **File -> Page Setup** и изучить вкладку.

С помощью **Size** (Размер) выбрать размер листа A4 (210 x 297 мм) .

С помощью **Orientation** выбрать ориентацию листа **Portrait** (вертикально) или **Landscape** (горизонтально).

Установить **Portrait** (вертикально).

С помощью Поля (мм) установите поля:

Left (слева) – 25 мм,

Right (справа) – 10 мм,

Top (вверху) – 30 мм,

Bottom (внизу) – 20 мм.

Посмотреть вид страницы на экране.

Данные установки позволяют нормально распечатывать и подшивать листы, обеспечивая при этом также достаточное поле для размещения материала.

Если желательно выводить большие графики, то можно развернуть страницу горизонтально.

С помощью **Orientation** выберите ориентацию листа **Landscape** (горизонтально) и убедитесь, что MathCAD сам поменял параметры **Left, Right, Top, Bottom**. Посмотреть вид страницы на экране.

2. Установка колонтитулов (поля выше и ниже текста).

Колонтитулы устанавливаются один раз, а затем повторяются при печати. На экране они не видны. Их можно посмотреть на экране в режиме просмотра макета страницы.

Войти в меню **View -> Headers/Footers** (Голова/Подножие). Изучить вкладку. Имеется возможность выбора между верхним **Header** и нижним **Footer** колонтитулами. Они по структуре идентичны.

Выбрать **Header** (верхний колонтитул). Имеется возможность располагать информацию слева, посередине и справа. Для этого существуют, соответственно, три окна: **Left, Center, Right**. В каждом из них можно поместить одинаковую информацию, представленную слева внизу пиктограммами:

в верхнем ряду:

[Insert File Name] – "Имя файла", которое позволяет знать, в каком файле находится данный материал.

[Insert File Path] – "Путь файла", то есть вложение в папки, который позволяет найти файл среди папок.

[Insert Page Number] – "Номер страницы в файле", обычная нумерация страниц по порядку.

[Insert Number of Pages] – "Общее количество страниц в файле", используется для формирования пояснений типа "с.7 из 22".

в нижнем ряду:

[Insert Date Last Saved] – "Дата последнего сохранения", которая дает информацию о дате последних исправлений в файле; необходимо для определения наиболее актуальной версии при накоплении распечаток.

[Insert Time Last Saved] – "Время последнего сохранения", которое дает информацию о времени последних изменений в файле.

[Insert Date] – "Текущая дата" – дата создания файла.

[Insert Time] – "Текущее время" – время создания файла.

Данная информация фактически соответствует имеющейся в каталоге для данного файла. Она переносится на бумагу. Чтобы не смешивать её с основным текстом, её располагают в колонтитулах. Она автоматически повторяется на всех страницах при печати.

Чтобы ввести команду, нужно мышкой поместить курсор в соответствующее окно, написать нужный текст в нужном месте пиктограммой и ввести команду. Команду можно и написать, например {f}.

Чтобы убрать команду, нужно мышкой поместить курсор рядом с соответствующей командой и стереть её обычным образом.

Установите в левом окне **Header** последовательно указатели [Путь файла, т.е. вложение в папки] и [Имя файла]. Нажать "ОК". Из главного меню открыть **File -> Print Preview** (предпросмотр макета страницы).

Особенностью MathCAD является то, что он не понимает названий файлов и папок с пробелами, как это допускается в Windows. Поэтому рекомендуется названия папок и файлов для MathCAD писать латиницей и слитно.

В окне до и после команды можно помещать поясняющий текст.

Введите перед командами текст: Файл.

С помощью просмотра макета страницы убедитесь, что он появился в колонтитуле.

Выводимый текст можно форматировать, выбирая шрифт и его тип (нормальный, курсив, жирный). Для этого во вкладке используют опцию **Format**.

С помощью **Format** можно установить различные шрифты.

Например, можно установить в правой части верхнего колонтитула нумерацию страниц, введя с помощью пиктограмм текст: Стр. {n} из {nn}

А с помощью просмотра макета страницы можно убедиться, что он появился в колонтитуле.

Также можно помещать в колонтитулах различные картинки, НО только в формате BMP. Для этого используется кнопка **Image** (изображение).

Если размер рисунка больше, чем поле (установленное расстояние до края страницы), то MathCAD предлагает увеличить соответствующее поле.

Попробуйте сами вставить в нижний колонтитул слева текущие дату и время с соответствующими поясняющими надписями, а справа – дату и время последнего сохранения. Для этого в левом окне нижнего колонтитула **Footer** написать текст и ввести пиктограммами команды: Дата {d} и время {t} создания. В правом окне написать текст и ввести пиктограммами команды: Дата {fd} и время {ft} сохранения. Для проверки работы команд выждать 1–2 минуты, сохранить файл и просмотреть его с помощью предпросмотрщика. Убедиться, что время сохранения файла изменилось, а время создания осталось прежним.

Для наглядности отделения колонтитулов от материала страницы можно использовать рамки.

Для изменения номера первой страницы в меню установок колонтитулов **View** → **Headers/Footers** используется установка **Start at Page Number**.

Чтобы на первой странице не появлялась информация колонтитулов, можно задать опцию **Different header and footer on first page** (другие верхний и нижний колонтитулы на первой странице).

Например, попробуйте установить опцию и проверить вид колонтитулов на первой и второй страницах. Можно убедиться, что для первой страницы появятся свои отдельные окна верхнего и нижнего колонтитулов, в которые можно ввести специальную информацию. Часто на первой странице колонтитулы пустые.

3. Установка ссылок

В MathCAD можно использовать не один лист, а несколько. Чтобы они совместно работали, их нужно вызывать одновременно и специальным образом. Для этого используются ссылки **Insert** → **Reference**. Необходимость такого подхода возникает, например, когда какие-то листы используются одновременно различными программами. При этом, возможно, необходимо, чтобы изменения в каких-то листах были общими для этих программ. Разделение на листы аналогично разделению обычных программ на подпрограммы или библиотеки. Такой способ организации вычислений позволяет также уменьшить размер листа перед пользователем.

Попробуйте сами организовать вызов других листов из основного листа. Для этого необходимо встать курсором в свободном поле, войти в меню **Insert** → **Reference**, изучить вкладку. В верхней части имеется узкое поле для ввода имени листа (файла), который должен вызываться. Чтобы найти его, можно воспользоваться кнопкой **Browse** и в каталогах найти нужный файл MathCAD-а. В нижней части имеется опция **Use relative path for reference** ("Использовать относительный путь для ссылки"). Значение этой опции следующее: если её не пометить, то вызов ссылки будет происходить по полному пути, который имеется для вызываемого файла на данном компьютере. И если перенести оба файла на другой компьютер с другим деревом каталогов, то ссылка не будет найдена. Чтобы этого не происходило, во-первых, устанавливаются указанную опцию, а во-вторых, оба файла помещают

в одном каталоге. Проверить работу с опцией и без опции **Use relative path for reference** (создав другой каталог и перенеся файлы в него).

Ссылки имеют гипертекстовую связь. Если нажать мышкой на ссылку, то вызовется соответствующий файл.

Вызов файлов через ссылки осуществляется иерархически. Обычно имеется основной вызывающий файл, который вызывает несколько других. Они, в свою очередь, могут вызывать другие файлы и т.д.

Все файлы (вызывающие и вызываемые) после вызова основного вызывающего файла оказываются в памяти компьютера и работают как единый файл, то есть расчёты производятся последовательно в порядке расположения файлов в ссылках (с учётом ссылок внутри файлов).

Особенность заключается в том, что при внесении изменений в файл ссылки, чтобы это изменение начало действовать на вызывающий файл (если они связаны между собой), необходимо, во-первых, сохранить файл ссылки, а во-вторых, заново вызвать вызывающий главный файл.

4. Установка гипертекстовых связей

Гипертекст позволяет организовать переходы по выделенным текстовым фрагментам (служащим в качестве кнопок) к другим файлам. Такими "кнопками" могут быть любые части материала: текст, формулы, вставленные рисунки, графики. Для этого используется меню **Insert** → **Hyperlink** с соответствующей вкладкой. Для её использования требуется выделить фрагмент текста, формулу, рисунок, график и после этого войти в меню **Insert** → **Hyperlink**. На открывшейся вкладке вверху расположена зона ввода имени файла. Для её заполнения можно использовать кнопку **Browse**, через которую можно найти в каталоге нужный файл, на который организуется гиперссылка. В нижней части в длинной пустой строке можно поместить поясняющую надпись, которая будет выводиться в левом нижнем углу MathCAD при наведении на "кнопку" гиперссылки указателя мышки.

Отличие гипертекста от ссылки заключается в следующем:

при установке ссылки соответствующий файл автоматически загружается при вызове главного файла, в котором имеется ссылка типа **Reference**;

при установке гипертекстовой связи вызываемый файл загружается в память только при нажатии гипертекстовой ссылки (кнопки) типа **Hyperlink**.

ЛАБОРАТОРНАЯ РАБОТА 16

ПЕРЕМЕННЫЕ И ОСНОВНЫЕ МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ В MATHCAD

Цель работы: научиться работать с различными математическими функциями MathCAD, при этом, используя присвоение переменным соответствующих значений и применяя различные математические операции.

Задания

Выполните следующие задания самостоятельно:

Вариант 1

1. Переменной x присвойте значение $x = \sqrt{2} \cdot \sin\left(\frac{\pi}{3}\right)$.
2. Переменной m присвойте комплексное значение $5 + 2i$, используя символ мнимости i , и переменной n – комплексное значение $3 - j$, используя j .
3. Посчитайте значения следующих функций от числа x : арккосинус, арктангенс, синус, косинус, котангенс, показательная функция e , натуральный логарифм, десятичный логарифм и гиперболический арксинус. Значение x задать самостоятельно.
4. Вычислите значение функции $\sin(x)$ и $\cos(x)$ на отрезке $[0, 10]$ с шагом 0.5.
5. Вычислите значение функции $z(x, y) = 4 \cdot (1 - x^2) \cdot y \cdot (1 + y)$ от двух переменных x и y . При этом x изменяется в диапазоне от 0 до 10, а y – от 5 до 20.

6. а) Найдите сумму и разность матриц

$$C = \begin{pmatrix} 4 & 5 & -2 \\ -6 & -3 & 7 \end{pmatrix}, \quad A = \begin{pmatrix} 4 & -7 & 8 \\ 2 & 6 & 0 \end{pmatrix};$$

б) умножьте матрицы C и $B = \begin{pmatrix} 3 & 5 & -2 \\ 7 & 2 & -8 \\ -4 & 6 & 0 \end{pmatrix}$;

в) полученную матрицу умножьте на 5.

г) из матрицы B извлеките второй столбец и вторую строчку;

д) транспонируйте матрицу B и найдите определитель этой матрицы.

7. Вычислите выражение $\begin{bmatrix} 3 & 2 & -4 \end{bmatrix} \cdot \begin{pmatrix} 6 & 3 & 0 \\ -8 & 5 & -1 \\ 7 & 2 & -4 \end{pmatrix} \cdot \begin{bmatrix} -6 \\ 5 \\ 1 \end{bmatrix}$.

8. Вычислите сумму векторов $a = \begin{pmatrix} 2.7 \\ 4.2 \\ 6.9 \end{pmatrix}$ и $b = \begin{pmatrix} 5.8 \\ 1.6 \\ 4.2 \end{pmatrix}$.

9. а) Перемножьте вектор-строки $v_1 = (9 \ 2 \ -5 \ 4)$ и $v_2 = (3 \ 7 \ -6 \ 1)$;
 б) возведите во вторую степень вектор v_1 ;
 в) вычислите сумму элементов вектора-строки v_1 .

10. Добавьте в матрицу $C = \begin{pmatrix} 4 & 5 & -2 \\ -6 & -3 & 7 \end{pmatrix}$ еще одну строчку, чтобы

получилась квадратная матрица.

11. Создайте матрицу три на три, воспользовавшись функцией *matrix* ().

12. Создайте матрицу, объединив векторы $a = \begin{pmatrix} 2.7 \\ 4.2 \\ 6.9 \end{pmatrix}$ и $b = \begin{pmatrix} 5.8 \\ 1.6 \\ 4.2 \end{pmatrix}$.

13. Из матрицы $B = \begin{pmatrix} 3 & 5 & -2 \\ 7 & 2 & -8 \\ -4 & 6 & 0 \end{pmatrix}$ выделите подматрицу $B = \begin{pmatrix} 3 & 5 & -2 \\ 7 & 2 & -8 \\ -4 & 6 & 0 \end{pmatrix}$.

Вариант 2

1. Переменной x присвойте значение $x = \sqrt{3} \cdot \cos\left(\frac{\pi}{4}\right)$.

2. Переменной m присвойте комплексное значение $3+i$, используя символ мнимости i , и переменной n – комплексное значение $8-4j$, используя j .

3. Посчитайте значения следующих функций от числа x : арккосинус, арктангенс, синус, косинус, котангенс, показательную функцию e , натуральный логарифм, десятичный логарифм и гиперболический арксинус. Значение x задать самостоятельно.

4. Вычислите значение функции $\sin(x)$ и $\cos(x)$ на отрезке $[0, 10]$ с шагом 0.5.

5. Вычислите значение функции $z(x, y) = 4 \cdot (1-x^2) \cdot y \cdot (1+y)$ от двух переменных x и y . При этом x изменяется в диапазоне от 0 до 10, а y – от 5 до 20.

6. а) Найдите сумму и разность матриц

$$C = \begin{pmatrix} 5 & 6 & -3 \\ -8 & -4 & 9 \end{pmatrix}, \quad A = \begin{pmatrix} 2 & -5 & 1 \\ 3 & 7 & 0 \end{pmatrix};$$

б) умножьте матрицы C и $B = \begin{pmatrix} 4 & 1 & -2 \\ 6 & 3 & -9 \\ -5 & 7 & 0 \end{pmatrix}$;

в) полученную матрицу умножьте на 5.

г) из матрицы B извлеките второй столбец и вторую строчку;

д) транспонируйте матрицу B и найдите определитель этой матрицы.

7. Вычислите выражение $\begin{bmatrix} 3 & 2 & -4 \end{bmatrix} \cdot \begin{pmatrix} 6 & 3 & 0 \\ -8 & 5 & -1 \\ 7 & 2 & -4 \end{pmatrix} \cdot \begin{bmatrix} -6 \\ 5 \\ 1 \end{bmatrix}$.

8. Вычислить сумму векторов $a = \begin{pmatrix} 2.7 \\ 4.2 \\ 6.9 \end{pmatrix}$, $b = \begin{pmatrix} 5.8 \\ 1.6 \\ 4.2 \end{pmatrix}$.

9. а) Перемножить вектор-строки $v1 = (9 \ 2 \ -5 \ 4)$ и $v2 = (3 \ 7 \ -6 \ 1)$;

б) возведите во вторую степень вектор $v1$;

в) вычислите сумму элементов вектора-строки $v1$.

10. Добавьте в матрицу $C = \begin{pmatrix} 4 & 5 & -2 \\ -6 & -3 & 7 \end{pmatrix}$ еще одну строчку, чтобы

получилась квадратная матрица.

11. Создайте матрицу три на три, воспользовавшись функцией *matrix* ().

12. Создайте матрицу, объединив вектора $a = \begin{pmatrix} 2.7 \\ 4.2 \\ 6.9 \end{pmatrix}$ и $b = \begin{pmatrix} 5.8 \\ 1.6 \\ 4.2 \end{pmatrix}$.

13. Из матрицы $B = \begin{pmatrix} 4 & 1 & -2 \\ 6 & 3 & -9 \\ -5 & 7 & 0 \end{pmatrix}$ выделите подматрицу $B = \begin{pmatrix} 4 & 1 & -2 \\ 6 & 3 & -9 \\ -5 & 7 & 0 \end{pmatrix}$.

ЛАБОРАТОРНАЯ РАБОТА 17

ПОСТРОЕНИЕ ДВУМЕРНЫХ ГРАФИКОВ В MATHCAD

Цель работы: научиться строить двумерные графики в MathCAD.

Задания

Выполните следующие задания самостоятельно.

Вариант 1

1. а) Постройте графики функций $y(x)$ и $g(x)$ на одних осях на отрезке $[0, 50]$ с шагом 0.05:

$$y(x) = \sin(2 \cdot x), \quad g(x) = \cos(0.5 \cdot x);$$

б) нанесите сетку и поменяйте цвет и толщину линий, подпишите оси.

2. а) Постройте график поверхности $z(x, y)$ на прямоугольной области определения $x \in [-1, 1]$, $y \in [0, 1]$:

$$z(x, y) = 4 \cdot \sin 2\pi x \cdot \cos 1.5\pi y \cdot (1 - x^2) \cdot y \cdot (1 - y);$$

б) залейте его цветом.

3. Постройте поверхности (сферы) с параметрическим заданием, при этом $i = 0 \dots 20$ $j = 0 \dots 20$, $k_i = i \cdot \frac{\pi}{20}$, $n_j = j \cdot \frac{\pi}{10}$, $X_{i,j} = \sin(k_i) \cdot \cos(n_j)$, $Y_{i,j} = \sin(k_i) \cdot \sin(n_j)$, $Z_{i,j} = \cos(k_i)$.

Вариант 2

1. Постройте графики функций $y(x)$ и $g(x)$ на одних осях на отрезке $[0, 3]$ с шагом 0.05:

$$y(x) = \tan(x - 3), \quad g(x) = -2 \cdot \tan(x);$$

б) нанесите сетку и поменяйте цвет и толщину линий, подпишите оси.

2. а) Постройте график поверхности $z(x, y)$ на прямоугольной области определения $x \in [-1, 1]$, $y \in [0, 1]$:

$$z(x, y) = 4 \cdot \sin 2\pi x \cdot \cos 1.5\pi y \cdot (1 - x^2) \cdot y \cdot (1 - y);$$

б) залейте его цветом.

3. Постройте поверхности (сферы) с параметрическим заданием, при этом $i = 0 \dots 20$ $j = 0 \dots 20$, $k_i = i \cdot \frac{\pi}{20}$, $n_j = j \cdot \frac{\pi}{10}$, $X_{i,j} = \sin(k_i) \cdot \cos(n_j)$, $Y_{i,j} = \sin(k_i) \cdot \sin(n_j)$, $Z_{i,j} = \cos(k_i)$.

ЛАБОРАТОРНАЯ РАБОТА 18

ЗАДАНИЕ СОБСТВЕННЫХ ФУНКЦИИ В MATHCAD. ПРОГРАММИРОВАНИЕ В MATHCAD

Цель работы. Научиться создавать свои собственные функции в MathCAD и программировать в среде MathCAD.

Задания

Выполните следующие задания самостоятельно:

1. Создайте свою собственную функцию, которая будет вычислять значение на отрезке $[0, 50]$ и постройте график этой функции.
2. Используя оператор цикла, вычислите значения синуса при значении аргумента от 0,2 до 10 с шагом 0,2.
3. Создайте матрицу произвольного размера. Заполните случайными числами. Найдите минимальный и максимальный элементы матрицы.
4. Напишите рекурсивную функцию *fact* для вычисления факториала числа N :

$$N! = \begin{cases} 1, & \text{если } N = 0; \\ N \times (N-1)!, & \text{если } N > 0. \end{cases}$$

5. Напишите программу, которая будет выводить на экране пилообразную функцию и постройте ее график.
6. Напишите программу, которая по коэффициентам квадратного уравнения находит только вещественные его корни.
7. Напишите программу, вычисляющую кусочно-заданную функцию

$$y(x) = \begin{cases} \sin x, & x < -\pi; \\ x / \pi, & -\pi \leq x < \pi; \\ -\cos x, & x \geq \pi. \end{cases}$$

Постройте ее график.

8. Напишите файл-программу для вычисления суммы

$$S = \sum_{k=1}^{10} \frac{e^k}{x^k}.$$

ЛАБОРАТОРНАЯ РАБОТА 19

РЕШЕНИЕ СИСТЕМ УРАВНЕНИЙ И НЕРАВЕНСТВ В MATHCAD

Цель работы: научиться решать системы уравнений и неравенств в среде MathCAD.

Задания

Выполните следующие задания самостоятельно.

Вариант 1

1. Решите систему уравнений
$$\begin{cases} 5 \cdot x^2 + 3\pi \cdot y - 7 = 0; \\ 6 \cdot x - y + 8 = 0. \end{cases}$$

2. Решите неравенство $8 \cdot x^2 - 6 \cdot x + 1 > 0$.

3. Найдите корни нелинейного уравнения $e^x - a \cdot x^2 = 0$ при $a = 1-10$ и постройте график.

4. Решите систему из 3 уравнений
$$\begin{cases} 1.6x_1 + 6.3x_2 - 0.7x_3 = 3.4; \\ 0.8x_1 + 6.8x_2 + 9.1x_3 = 7.5; \\ -0.5x_1 + 5.5x_2 + 8.1x_3 = 5.1. \end{cases}$$

Вариант 2

1. Решите систему уравнений
$$\begin{cases} 4 \cdot x^2 + 6\pi \cdot y - 8 = 0; \\ 8 \cdot x - y + 6 = 0. \end{cases}$$

2. Решите неравенство $6 \cdot x^2 - 4 \cdot x + 1 > 0$.

3. Найдите корни нелинейного уравнения $e^x - a \cdot x^2 = 0$ при $a = 1-10$ и постройте график.

4. Решите систему из 3 уравнений
$$\begin{cases} 1.4x_1 + 6.1x_2 - 0.5x_3 = 3.6; \\ 0.6x_1 + 2.8x_2 + 9.1x_3 = 7.2; \\ -0.4x_1 + 5.3x_2 + 8.1x_3 = 5.1. \end{cases}$$

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ануфриев, И. Е. Самоучитель MatLab 5.3/6.x / И. Е. Ануфриев. – СПб. : БХВ-Петербург, 2002. – 736 с.
2. Васильев, А. Н. Mathcad 13 на примерах / А. Н. Васильев. – СПб. : БХВ-Петербург, 2006. – 528 с.
3. Дьяконов, В. Mathcad : учебный курс / В. Дьяконов. – СПб. : Питер, 2001. – 624 с.
4. Дьяконов, В. П. Система MathCAD : справ. / В. П. Дьяконов. – М. : Радио и связь, 1993. – 128 с.
5. Дьяконов, В. Simulink 4. Специальный справочник / В. Дьяконов. – СПб. : Питер, 2002. – 528 с.
6. Лазарев, Ю. Моделирование процессов и систем в MATLAB : учебный курс / Ю. Лазарев. – СПб. : Питер; Киев : Издательская группа ВНУ, 2005. – 512 с.
7. Макаров, Е. Г. Инженерные расчёты в MathCAD : учебный курс / Е. Г. Макаров. – СПб.: Питер, 2005. – 448 с.
8. Медведев, В. С. Control System Toolbox. MATLAB 5 для студентов / В. С. Медведев, В. Г. Потемкин. – М. : ДИАЛОГ-МИФИ, 1999. – 287 с.
9. Мэтьюз, Дж. Г. Численные методы. Использование MATLAB : пер. с англ. / Дж. Г. Мэтьюз, К. Д. Финк. – 3-е изд. – М. : Издательский дом "Вильямс", 2001. – 720 с.
10. Кривелёв, А. В. Основы компьютерной математики с использованием системы MATLAB / А. В. Кривелёв. – М. : Лекс-Книга, 2005. – 496 с.
11. Лазарев, Ю. Ф. MatLab 5.x / Ю. Ф. Лазарев. – К.: Издательская группа ВНУ, 2000. – 384 с. – ISBN 966-552-068-7.
12. Леоненков, А. Нечёткое моделирование в среде MATLAB и fuzzyTECH / А. Леоненков. – СПб. : БХВ-Петербург, 2003. – 736 с.
13. Очков, В. Ф. Физические и экономические величины в Mathcad и Maple / В. Ф. Очков. – М. : Финансы и статистика, 2002. – 192 с.
14. Ракитин, В. И. Руководство по методам вычислений и приложения MathCAD / В. И. Ракитин. – М. : ФИЗМАТЛИТ, 2005. – 264 с.
15. Черняк, А. А. Высшая математика на базе Mathcad. Общий курс / А. А. Черняк, Ж. А. Черняк, Ю. А. Доманова. – СПб. : БХВ-Петербург, 2004. – 608 с.