Министерство образования и науки Российской Федерации

ФГБОУ ВПО «Тамбовский государственный технический университет»

Институт автоматики и информационных технологий

П.В. Балабанов, Н.А. Конышева, Д.А. Любимова

Система автоматизированных расчетов MATLAB

Лабораторный практикум

по дисциплине «Системы автоматизированных расчетов» для студентов дневной, заочной и очно-заочной форм обучения направления

27.03.02 «Управление качеством»

Тамбов 2015

Рецензент
Доктор технических наук
А.И. Петрашев

Утверждено Методическим советом ФГБОУ ВПО «ТГТУ»
(протокол № 6 от 20.11.2015)

СОДЕРЖАНИЕ

Лабораторная работа № 1

ИНТЕРФЕЙС ОСНОВНОГО ОКНА MATLAB R2013b. ПЕРЕМЕННЫЕ И ФУНКЦИИ В MATLAB

Цель работы: изучить интерфейс основного окна MatLab R2013b, математические и логические функции системы.

Задание. По заданному варианту (см. табл. 1.1) рассчитать значения выражений в режиме прямых вычислений и в режиме программы.

Методические указания

MatLab – одна из старейших и тщательно проработанных систем автоматизации математических расчетов, созданная еще в конце 70-х годов. Она обладает широкими возможностями, является гибкой и расширяемой системой, легко приспосабливаемой к решению различных классов задач, и характеризуется высокой скоростью вычислений в сравнении с другими подобными системами.

Своим названием (от английского MATrix LABoratory – «матричная лаборатория») система MatLab обязана ориентации на матричные и векторные вычисления. Обычно такие вычисления требуют подготовки специальных и достаточно сложных программ, которые ранее писались на языках высокого уровня – Бейсике, Фортране, Паскале, Си и др. Система MatLab выполняет операции над матрицами и векторами даже в режиме прямых вычислений без какого-либо программирования. Ею можно пользоваться как мощнейшим калькулятором, в котором наряду с обычными арифметическими и алгебраическими действиями могут использоваться такие сложные операции, как обращение матрицы, решение систем линейных уравнений и многое другое.

Но главная отличительная черта системы – это легкость ее модификации и адаптация к конкретным задачам пользователя. Пользователь может ввести в систему любую новую команду, оператор или функцию и пользоваться затем ими так же просто, как и заведомо встроенными операторами и функциями. Новые определения в системе MatLab хранятся в виде файлов на диске, имеющих расширение *.m. Это делает набор операторов и функций практически неограниченным.

В базовый набор слов системы входят: спецзнаки, знаки арифметических и логических операций, арифметические, алгебраические, тригонометрические и некоторые специальные функции, функции быстрого преобразования Фурье и фильтрации, векторные и матричные функции, средства для работы с комплексными числами, операторы построения гистограмм, графиков в декартовой и полярной системах координат, трехмерных поверхностей и др.

Достоинством системы является достаточно простой и традиционный входной язык, напоминающий язык высокого уровня Бейсик. При этом система дает возможность редактировать программы с помощью любого текстового редактора.

В отношении графических возможностей MatLab можно отметить, что система позволяет на одном графике представлять множество кривых и оформлять их различными способами. Графики можно также выводить в несколько окон. Удобно и эффективно в MatLab реализована возможность построения трехмерных поверхностей и фигур.

Таким образом, система MatLab рассчитана на серьезное применение и открывает перед пользователем огромные возможности по реализации сложных математических методов расчетов, обладает гибкостью и легко адаптируется к решению различных задач, за счет возможности создания пользователем своих новых определений.

1. Интерфейс основного окна Matlab R2013b

Как любая программа, MatLab R2013b имеет элемент графического интерфейса программы – *лента (вкладка)*, которая упрощает поиск необходимых элементов посредством командных вкладок, расположенных в верхней части окна (рис. 1.1): Home, Plots, Apps.

Лента динамична, т. е. по мере смены задач вкладки в ней перемещаются. Например, при работе с документами (панель *File* на ленте *Home*) лента *Plots* с панелями для построения и редактирования графиков не отображается. Для каждого отдельного программного компонента пакета MatLab лента будет содержать различное количество вкладок и помещенных в них инструментов. Рассмотрим вкладки, которые необходимы для решения задач в режиме вычислений – *лента Home*:

New script – новый сценарий (выводит окно редактора с новым m-файлом);

New (открытие подменю с позициями):

New script – открывает окно редактора/отладчика m-файлов;

Function – открывает окно с шаблоном для создания функции пользователя;

Figure – открытие пустого окна графики;

Open – выводит подменю с позициями:

Open – открывает существующий файл (m-файл);

Find files – открывает окно для поиска заданного файла;

На вкладке *Home* расположены кнопки:

Import data – открывает окно импорта файлов данных (рис. 1.2);

Save workspace – открывает окно записи рабочей области в виде файла с заданным именем;

New variable – создание новой переменной (рис. 1.3);

Clear workspace – очистка рабочей области;

Set Path –открывает окно установки путей доступа файловой системы;

Preference –открывает окно настройки элементов интерфейса;

Simulink – открывает окно браузера библиотек Simulink;

Help – открывает окно справки.

Набор кнопок панели инструментов обеспечивает выполнение наиболее часто необходимых команд и вполне достаточен для повседневной работы с системой.

О назначении кнопок говорят и всплывающие подсказки, появляющиеся, когда курсор мыши устанавливается на соответствующую кнопку. Они имеют вид желтого прямоугольника с текстом короткой справки.

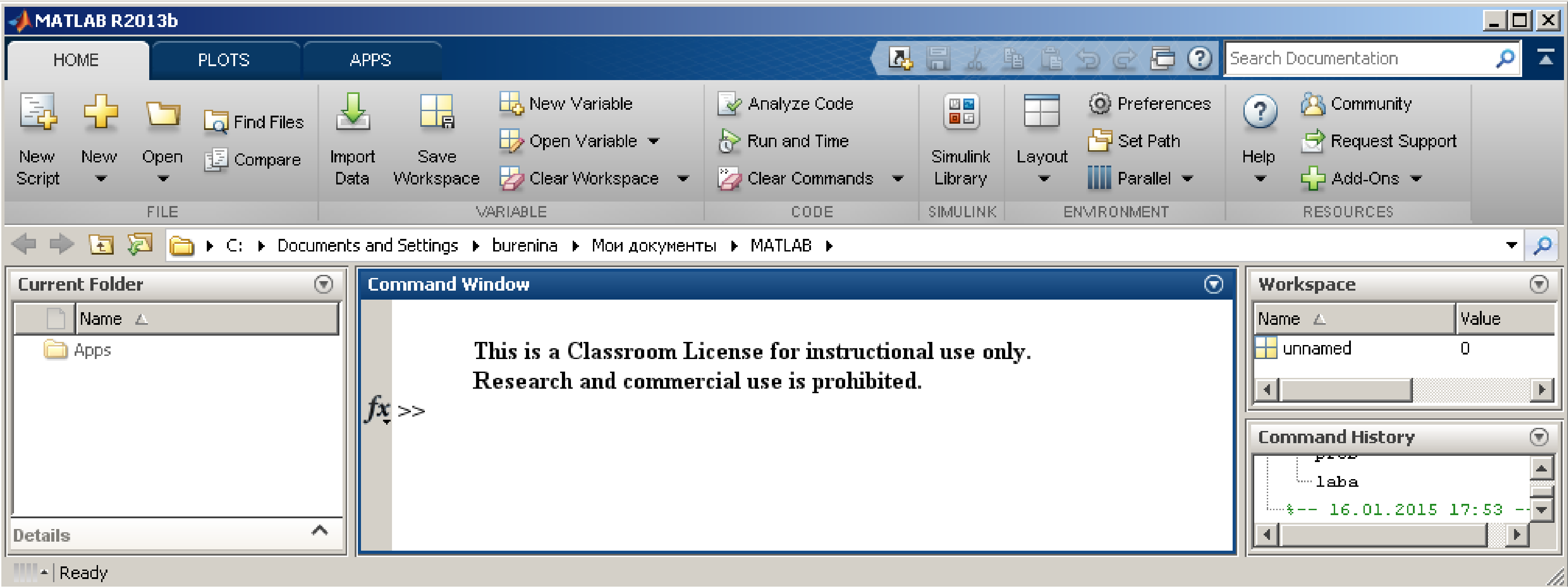

Рис. 1.1. Окно Matlab R2013b

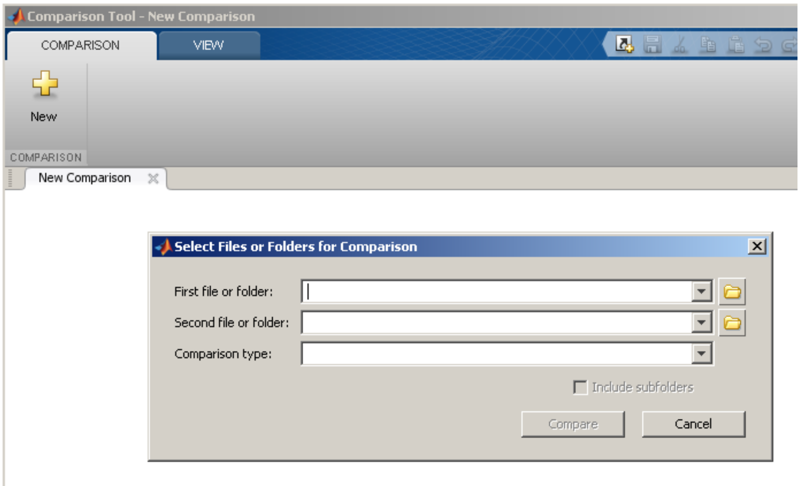

Рис. 1.2. Окно импорта файлов данных

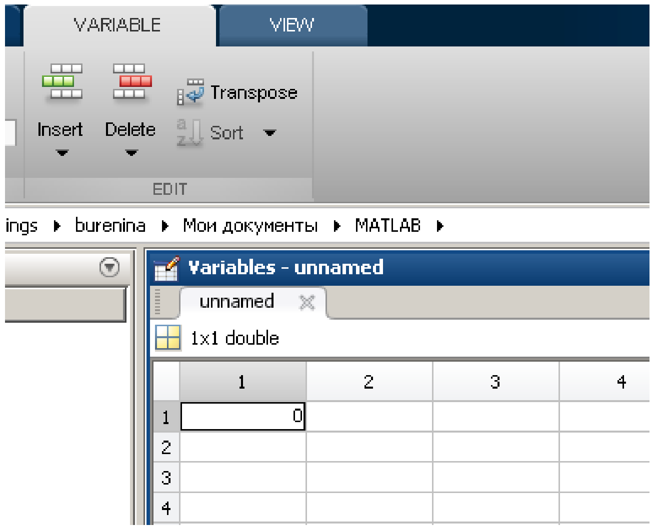

Рис. 1.3

2. Настройка элементов интерфейса

Команда *Preferences* выводит окно детальной настройки элементов интерфейса (рис. 1.4).

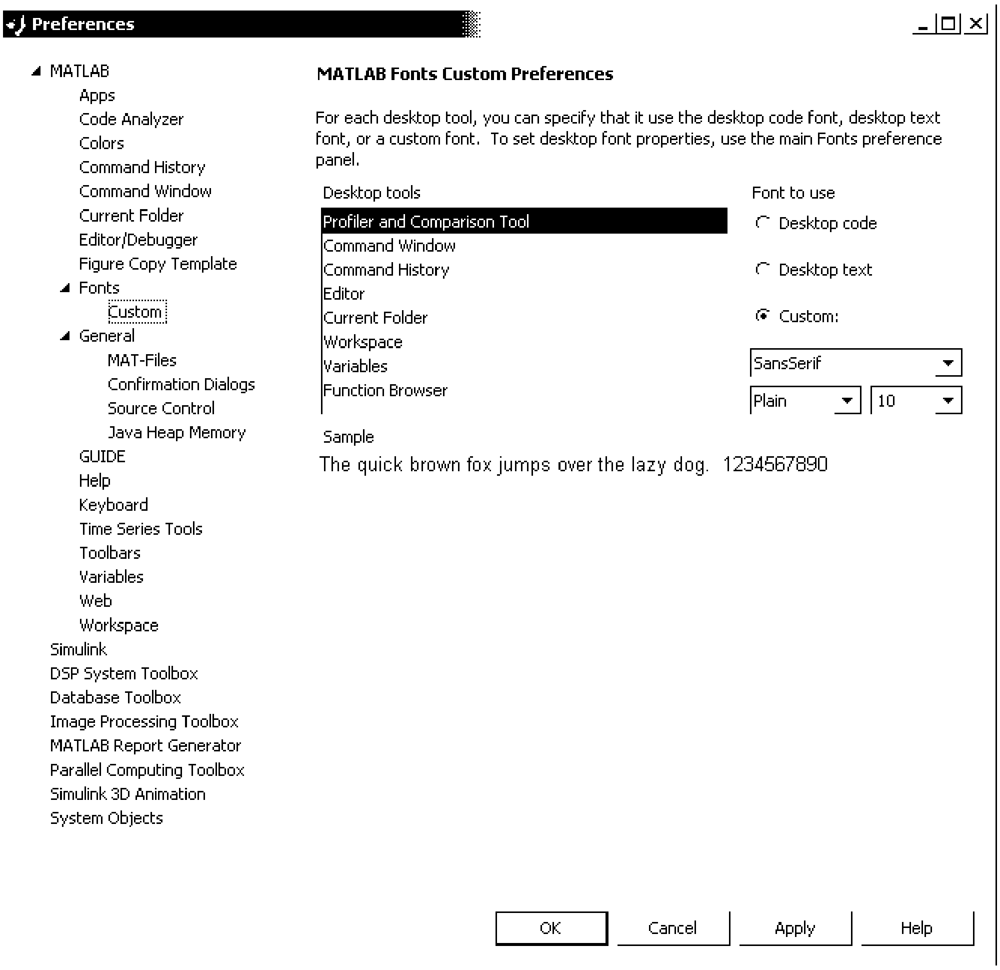

Рис. 1.4. Окно настройки элементов интерфейса

В левой части этого окна имеется древообразный список элементов интерфейса системы, а в правой части – поле задания параметров для выбранного типа элементов. Число параметров и видов этого окна велико, пользователь выбирает нужные ему параметры для настройки.

3. Интерфейс редактора/отладчика m-файлов

Программы в системе MatLab представлены m-файлами. Для подготовки редактирования и отладки m-файлов (а в MatLab R2013b и файлов различных языков программирования) служит специальный многооконный редактор. Он выполнен как типичное приложение Windows. Редактор можно вызвать командой *Edit* из командной строки или командой

New script из меню *New* (или кнопка *New script*). После этого в окне редактора можно создавать свой файл, пользоваться средствами его отладки и запуска. Перед запуском файла его необходимо записать на диск, используя команду *Save as* в меню редактора *Save*.

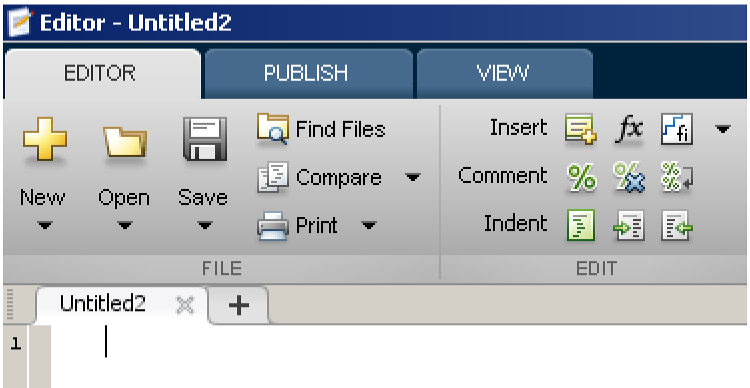

Рис. 1.5. Editor – средство редактирования документов (редактор)

На рисунке 1.5 показано окно редактора/отладчика MatLab R2013b с именем простого файла *Untitled2.m* в окне редактирования и отладки.

После записи файла на диск можно заметить, что команда *Run* в меню редактора позволяет произвести запуск файла. Запустив команду *Run*, можно наблюдать исполнение m-файла.

В правом верхнем углу меню редактора находится панель быстрого доступа (рис. 1.6).

Рис. 1.6. Панель быстрого доступа

Это меню имеет следующие операции и команды:

Save (ctrl + S) – сохранение содержимого m-файла;

Cut (ctrl+ X) – вырезание выделенного фрагмента и перенос его в буфер;

Copy (ctrl + C) – копирование выделенного фрагмента в буфер;

Past (ctrl + V) – вставка фрагмента из буфера в текущую позицию курсора;

Undo (ctrl +Z) – отмена результата предшествующей операции;

Redo (ctrl + V) – отмена действия последней операции Undo;

Switch windows – переключение окон редактора.

В MATLAB можно использовать контекстное меню, появляющееся при нажатии правой кнопки мыши. Например, установив курсор мыши на выделенный фрагмент матрицы, можно увидеть меню, которое дублирует позиции с командой *Copy* (Копировать). Есть и ряд других доступных в данный момент команд.

Для удобства работы с редактором/отладчиком строки программы в нем нумеруются в последовательном порядке. Редактор является многооконным. Окно каждой программы оформляется как вкладка. Редактор-отладчик позволяет легко просматривать значения переменных. Для этого достаточно подвести к имени переменной курсор мыши и задержать его – появится всплывающая подсказка с именем переменной и ее значением.

4. Цветовые выделения и синтаксический контроль

Редактор/отладчик m-файлов выполняет синтаксический контроль программного кода по мере ввода текста. При этом используются следующие цветовые выделения:

• ключевые слова языка программирования – синий цвет;

• операторы, константы и переменные – черный цвет;

• комментарии после знака % – зеленый цвет;

• символьные переменные (в апострофах) – зеленый цвет;

• синтаксические ошибки – красный цвет.

Благодаря цветовым выделениям вероятность синтаксических ошибок резко снижается.

Однако далеко не все ошибки диагностируются. Ошибки, связанные с неверным применением операторов или функций (например, применение оператора – вместо + или функции *cos(x)* вместо *sin(x)* и т. д.), не способна обнаружить ни одна система программирования. Устранение такого рода ошибок (их называют семантическими) – дело пользователя, отлаживающего свои алгоритмы и программы.

При работе с системой возможны два режима работы: **режим прямых вычислений**, когда пользователь на клавиатуре вводит вычисляемое выражение, нажимает «Enter», а система выдает результат; и **режим программы**, когда пользователем последовательность необходимых вычислений предварительно оформляется в виде программы, а затем данная программа запускается на выполнение.

Здесь полезно отметить, что m-файлы, создаваемые редактором/отладчиком делятся на два класса:

- *файлы-сценарии*, не имеющие входных параметров;
- *файлы-функции*, имеющие входные параметры полноценные процедуры.

Файл-сценарий или *Script-файл* не имеет списка входных параметров и является примером простой процедуры без параметров. Он использует глобальные переменные, то есть такие переменные, значения которых могут быть изменены в любой момент сеанса работы и в любом месте программы. Для запуска файла-сценария из командной строки MATLAB достаточно указать его имя в этой строке.

Файл-функция отличается от файла-сценария, прежде всего тем, что созданная им функция имеет входные параметры, список которых указывается в круглых скобках. Используемые в файле-функции переменные являются локальными переменными, изменение значений которых в теле функции никоим образом не влияет на значения, которые те же самые переменные могут иметь за пределами функции.

В режиме прямых вычислений команда вводится после приглашающего символа «>>». Ввод команды завершается клавишей «Enter».

В арифметических выражениях можно использовать следующие знаки арифметических операций: + (сложение), – (вычитание), * (умножение), / (деление), ^ (возведение в степень).

Например,

```
>> 2+2*2
    ans=
          6
```

здесь ans – переменная, хранящая результат последней операции. Кроме этого система содержит следующие системные переменные: pi – число «пи», inf – значение машинной бесконечности. В случае возникновения неопределенности типа 0/0 или inf/inf система выдает результат NaN (not a number).

Если используется операция присваивания

Имя_переменной = Выражение [;]

то результат вычисления выражения помещается не в системную переменную ans, а в переменную с именем «Имя_переменной». Символ «;» указывает на то, что результат вычисления не будет выводиться на экран. Этот символ используется также и для разделения нескольких операторов в одной строке.

Имена переменных должны формироваться из букв латинского алфавита, цифр, символа подчеркивания и могут начинаться только с буквы. Например,

>> k=2+2*2 k= 6	>> km2=2+2*2 km2= 6	>> k_1=2+2*2 k_1= 6

<u>Замечание:</u> **заглавные и строчные буквы системой различаются!**

Система MatLab может работать как с действительными, так и с комплексными числами, которые записываются следующим образом

```
>>5+6*i
   ans =
         5.0000 + 6.0000i
```

5. Математические функции системы MatLab

В MatLab имеются следующие математические функции (Z – в общем случае число комплексное, X и Y – только действительные числа):

abs(Z) – вычисление модуля комплексного или действительного числа,

angle(Z) – вычисление аргумента,

sqrt(Z) – вычисление квадратного корня,

real(Z) – вычисление действительной части числа,

imag(Z) – вычисление мнимой части числа,

round(X) – округление до целого;

fix(X) – округление до ближайшего целого в сторону нуля,

floor(X) – округление до ближайшего целого в сторону отрицательной бесконечности,

sign(X) – вычисление функции знака $\begin{cases} -1, & \text{при } X < 0, \\ 0, & \text{при } X = 0, \\ 1, & \text{при } X > 0, \end{cases}$

rem(X, Y) – вычисление остатка от деления X на Y,

exp(Z) – вычисление экспоненты,

log(Z) – вычисление натурального логарифма,

log10(Z) – вычисление десятичного логарифма,

sin(Z) – вычисление синуса,

cos(Z) – вычисление косинуса,

tan(Z) – вычисление тангенса,

asin(Z) – вычисление арксинуса,

acos(Z) – вычисление арккосинуса,

atan(Z) – вычисление арктангенса,

sinh(Z) – вычисление гиперболического синуса,

cosh(Z) – вычисление гиперболического косинуса,

tanh(Z) – вычисление гиперболического тангенса,

asinh(Z) – вычисление гиперболического арксинуса,

acosh(Z) – вычисление гиперболического арккосинуса,

atanh(Z) – вычисление гиперболического арктангенса.

Например,

```
>> Z=20+20i; asin(Z)
    ans =
           0.7851 + 4.0355i
```

6. Операторы отношения и логические операции

В системе могут применяться следующие операторы отношения (сравнения):

<	– меньше чем,	>=	– больше или равно,
<=	– меньше или равно,	==	– равно,
>	– больше чем,	~=	– не равно.

Эти операторы используются, как правило, в режиме программы (а не прямых вычислений) для сравнения двух величин или арифметических выражений. Они возвращают значение «Неверно» или «Верно» в виде чисел 0 или 1 соответственно. Например,

```
>> Z=9; Z==0
    ans =
           0
```

Возможно также выполнение логических операций:

&	– логическое умножение (И),
|	– логическое сложение (ИЛИ),
~	– логическое отрицание (НЕ).

Таблица 1.1. Варианты заданий

Вариант	Вычислить значение выражения	
	Задача 1	**Задача 2**
1	**2**	**3**
1	$y=(a+b+c)/e^{2a}+ab+3c$ при $a=1{,}78$, $b=2{,}5$, $c=4{,}5$ $y=\ln\left(\frac{a\sqrt[3]{z}}{c}\right)+b$ при $a=10{,}4$, $b=-3{,}2$, $c=2{,}5$, $z=0{,}1$	c = a & b при a = 2, b = 4
2	$y=x^z+a/\left(b^2+c^2\right)$ при $a=4{,}8$, $b=2{,}3$, $c=6{,}95$, $x=1{,}3$; $z=5$ $y=\frac{a^2+\ln(b)}{\sqrt{cd}}$ при $a=-3{,}1$, $b=2{,}9$, $c=4{,}09$, $d=0{,}875$	c = a | b при a = 2, b = 4
3	$y=a^2+b^2c^2/\left(a+z^c\right)$ при $a=1{,}8$, $b=4{,}65$, $c=7{,}8$; $z=0{,}3$ $y=\frac{\sqrt{a}+e^b}{\sin^3(cd)}$ при $a=9{,}4$, $b=1{,}8$, $c=0{,}395$, $d=2{,}344$	c = a < b при a = 0; b = 1
4	$y=z^2+z^3+a^2$ при $z=1{,}96$, $a=8{,}4$ $y=\sin^2\left(\frac{\pi x}{b+1{,}5}\right)-z^a$ при $a=2{,}5$, $b=4{,}1$, $x=5$, $z=0{,}85$	c = a > b при a = 2; b = 4
5	$y=\sin\left(a^2+bx\right)/\cos\left(b^2+ax\right)$ при $a=2{,}6$, $b=1{,}8$, $x=4{,}4$ $y=x\ln(xz)/\left(ax+\sqrt{bx^2}\right)$ при $a=7{,}5$, $b=3{,}9$, $x=4{,}1$, $z=0{,}9$	c = a ~ = b при a = 1, b = 1
6	$y=\ln\left(ax+b+cx^2\right)/\ln\left(x^2\right)$ при $a=3{,}5$, $b=7{,}5$, $x=4{,}1$, $c=7{,}8$ $y=\sin^3(a+b+c)/\cos(a+b)$ при $a=0{,}2$, $b=4{,}3$, $c=3{,}8$	c = a & b при a = 1, b = 1
7	$y=\sqrt{x^2+bx+c}/\sqrt[4]{a^2+b^2}$ при $x=2{,}85$, $b=3{,}4$, $c=5$, $a=9{,}1$ $y=(a+b+c)/e^{2a}+ab+3c$ при $a=1{,}78$, $b=2{,}5$, $c=4{,}5$	c = a | b при a = 1, b = 2

Продолжение таблицы 1.1. Варианты заданий

1	2	3
8	$y = x^2 + \sqrt{b}\big/\left(b^2 + c^2 + a^2\right)$ при $a = 1{,}84$, $b = 13{,}5$, $c = 2{,}4$, $x = 1$ $y = x^z + a\big/\left(b^2 + c^2\right)$ при $a = 4{,}8$, $b = 2{,}3$, $c = 6{,}95$, $x = 1{,}3$, $z = 5$	c = a & b при a = 0, b = 0
9	$y = \arcsin(x)\arccos\left(x^2\right)\big/x^3$ при $x = 0{,}3$ $y = a^2 + b^2c^2\big/\left(a + z^c\right)$ при $a = 1{,}8$, $b = 4{,}65$, $c = 7{,}8$, $z = 0{,}3$	c = a | b при a = 0, b = 0
10	$y = \sqrt[5]{(a+b+c)^2}\big/\left(c^2 + e^c\right)$ при $a = 0{,}8$, $b = 4{,}4$, $c = 7{,}5$ $y = \sin\left(a^2 + bx\right)\big/\cos\left(b^2 + ax\right)$ при $a = 2{,}6$, $b = 1{,}8$, $x = 4{,}4$	c = a = = b при a = 2, b = 4
11	$y = \sin\left(ax + bx^2 + c\right)\big/\cos^2(a+b+c)$ при $a = 0{,}2$, $b = 4{,}3$, $c = 3{,}8$, $x = 2$. $y = (a+b+c)\big/e^{2a} + ab + 3c$ при $a = 1{,}78$, $b = 2{,}5$, $c = 4{,}5$	c = a > = b при a = 1, b = 1
12	$y = \sin^3(a+b+c)\big/\cos(a+b)$ при $a = 0{,}2$, $b = 4{,}3$, $c = 3{,}8$ $y = x\ln(xz)\Big/\left(ax + \sqrt{bx^2}\right)$ при $a = 7{,}5$, $b = 3{,}9$, $x = 4{,}1$, $z = 0{,}9$	c = a < = b при a = 0, b = 1
13	$y = a^2 + b^2 + \sqrt[4]{c}\big/e^{ax+b}$ при $a = 4$, $b = 5{,}2$, $c = 0{,}5$, $x = 0{,}8$ $y = \sin^2\left(\dfrac{\pi x}{b + 1{,}5}\right) - z^a$ при $a = 2{,}5$, $b = 4{,}1$, $x = 5$, $z = 0{,}85$	c = a & b при a = 1, b = 10
14	$y = x\ln(xz)\Big/\left(ax + \sqrt{bx^2}\right)$ при $a = 7{,}5$, $b = 3{,}9$, $x = 4{,}1$, $z = 0{,}9$ $y = \sin^2\left(\dfrac{\pi x}{b + 1{,}5}\right) - z^a$ при $a = 2{,}5$, $b = 4{,}1$, $x = 5$, $z = 0{,}85$	c = a < > b при a = 2, b = 4
15	$y = \sin^3(a+b+c)\big/\cos(a+b)$ при $a = 0{,}2$, $b = 4{,}3$, $c = 3{,}8$ $y = \ln\left(\dfrac{a\sqrt[3]{z}}{c}\right) + b$ при $a = 10{,}4$, $b = -3{,}2$, $c = 2{,}5$, $z = 0{,}1$	c = a & b при a = 2, b = 2

Лабораторная работа № 2

РАБОТА С МАТРИЦАМИ В СИСТЕМЕ MATLAB

Цель работы: получить навыки работы с матрицами и векторами в системе MatLab; ознакомиться с основными матричными и векторными функциями.

Задание. Выполнить операции с векторами и матрицами, решить систему линейных алгебраических уравнений по заданному варианту.

Методические указания

Поскольку MatLab – матрично-ориентированная система, то в ней даже обычная переменная представляет собой матрицу размерностью 1×1. Чтобы задать вектор (матрицу из одной строки), состоящий, например, из трех элементов, то их значения необходимо перечислить в квадратных скобках, разделяя пробелами или запятыми:

```
>> P=[12 25 31]
   P =
       12    25    31
```

Получить значение каждого элемента вектора можно по его индексу, например, для второго элемента

```
>> P(2)
   ans =
         25
```

Задание матрицы требует указания различных строк. Для различения строк используется знак «;», например:

```
>> Q=[3 6;4 10]
   Q =
        3    6
        4   10
>> Q(2,1)
   ans =
```

```
            4
```

Обращение к элементам матрицы при помощи одного индекса, поясним на следующем примере.

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

>> A(2) ans = 4	>> A(6) ans = 8

Можно осуществлять вывод одной строки или одного столбца матрицы. Для этого используется знак «:», например,

Вывод 2-ой строки >> A(2,:) ans = 4 5 6	Вывод 3-го столбца >> A(:,3) ans = 3 6 9

Для преобразования матрицы в столбец используется также символ «:». Например,

```
>> A(:)
ans =
   1
   4
   7
   2
   5
```

Помимо поэлементного ввода векторов и матриц можно задавать значения в виде арифметической прогрессии с шагом 1:

```
>> x=[1:5]
x =
    1    2    3    4    5
```

с произвольным шагом

```
>> x=[1:0.5:3]
x =
    1.0000    1.5000    2.0000    2.5000    3.0000
```

или с шагом, заданным при помощи арифметического выражения:

```
>> x=[0:pi/4:pi]
x =
    0    0.7854    1.5708    2.3562    3.1416
```

Следующие матричные функции обеспечивают генерацию некоторых наиболее распространенных видов матриц размерностью M строк на N столбцов:

zeros(M, N) – генерация матрицы с нулевыми элементами,

ones(M, N) – генерация матрицы с единичными элементами,

rand(M, N) – генерация матрицы с элементами, имеющими случайные значения,

eye(M, N) – генерация матрицы с единичными диагональными элементами.

Например,

```
>> Q=eye(3,4)
Q =
     1    0    0    0
     0    1    0    0
     0    0    1    0
```

Транспонирование квадратной матрицы осуществляется при помощи символа апостроф («'»):

```
>> A=[1 2;3 4]
    A =
         1    2
         3    4
>> B=A'
    B =
         1    3
         2    4
```

Для сложения и вычитания матриц используются обычные знаки «+» и «–». Умножение двух матриц по правилам линейной алгебры осуществляется с использованием знака «*». В случае, когда требуется перемножить поэлементно две матрицы, используется знак «.*». Например,

```
>> A=[1 2;3 4]; B=[3 2;0 1]; C=A*B
    C =
         3    4
         9   10
>> C=A.*B
    C =
         3    4
         0    4
```

Аналогично, знаки «./», «.^» означают поэлементное деление и поэлементное возведение каждого элемента матрицы в степень соответственно.

Пример. Решение системы линейных алгебраических уравнений.

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 4 \\ x_1 + x_2 - x_3 = 1 \\ x_1 - 2x_2 + x_3 = 3 \end{cases} \quad \text{или} \quad \begin{pmatrix} 3 & 2 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 3 \end{pmatrix}$$

В матричной форме система записывается следующим образом $A \times X=B$, где A – матрица линейных коэффициентов, X – искомый вектор неизвестных, B – вектор свободных членов. Тогда решение данной системы – $X=A^{-1} \times B$, где A^{-1} – обратная матрица. В MatLab обращению матрицы соответствует возведение ее в степень –1.

Выполним решение заданной системы уравнений:

```
>> A=[3 2 1;1 1 -1;1 -2 1]; B=[4;1;3]; X=A^(-1)*B
 X =
     1.7000
    -0.6000
     0.1000
```

Для вычисления определителя квадратной матрицы M используется функция det(M), например:

```
>> M=eye(3,3)
    M =
        1     0     0
        0     1     0
        0     0     1
>> det(M)
    ans =
          1
```

Для определения размерности вектора-строки используется функция length, а размерности матрицы – size. Например,

```
>> A=[3 2 1];B=[1 2; 3 4;5 6];k=length(A)
k =
    3
>> [m,n]=size(B)
m =
```

```
    3
n =
    2
```

Функции min (max) для матрицы возвращают вектор-строку минимальных (максимальных) значений каждого столбца матрицы, например,

```
>> A=[3 2 1; 1 2 3]; min(A)
    ans =
         1    2    1
```

Применение этих функций к векторам-строкам позволяет определить наибольший (наименьший) элемент в строке, например,

```
>> A=[3 2 1]; max(A)
    ans =
         3
```

а также его порядковый номер в строке

```
>> [M,m]=max(A)
    M =
        3
     m =
        1
```

Функции sum, prod, mean и std для матриц возвращают вектор-строку сумм, произведений, средних значений и стандартных квадратичных отклонений каждого столбца матрицы, а для вектора-строки возвращают сумму, произведение, среднее значение и стандартное квадратичное отклонение соответственно.

Для матрицы функция sort возвращает матрицу, у которой отсортированы в порядке возрастания элементы каждого столбца, а для вектора-строки - упорядочивает все его элементы, например,

```
>> A=[10 2 11; 23 1 0; 1 2 3]; B=sort(A)
    B =
         1     1     0
        10     2     3
        23     2    11
>> A=[10 2 11]; B=sort(A)
     B =
          2    10    11
```

Матричная функция find (A) возвращает индексы ненулевых элементов матрицы A. Например,

```
>>A=[1 0 3; 0 0 4; 2 2 2]; find(A)
   ans =
           1
           3
           6
           7
           8
           9
```

Если в скобках указано логическое условие, то с помощью этой функции осуществляется поиск элементов матрицы удовлетворяющих этому условию. При этом возвращаются индексы найденных элементов. Например,

>> k=find(A>2)	>> k=find(A==0)
k =	k =
7	2
8	4
	5

Варианты заданий

Вариант №1

1. Задать вектор X из 6 элементов случайным образом в диапазоне значений от –5 до 5. Изменить 3-ий элемент на 0. Для вектора X найти минимальный элемент и его порядковый номер.

2. Задать две матрицы A и B размерностью 5 x 5 с клавиатуры. Найти определитель матрицы A, произведение матриц A и B, а также разделить поэлементно A на B.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} x_1 + 2x_2 - x_3 = 2; \\ 2x_1 - x_2 + 2x_3 = 11; \\ 4x_1 + x_2 - 2x_3 = 1. \end{cases}$$

Вариант №2

1. Задать вектор X, элементы которого представляют собой арифметическую последовательность от $\pi/16$ до π с шагом $\pi/16$, и вектор Y - от $\pi/8$ до 2π с шагом $\pi/8$. Найти произведение максимального элемента вектора X на минимальный элемент вектора Y. Найти вектор попарных произведений элементов векторов X и Y.

2. Задать матрицу A размерностью 6 x 7 случайным образом в диапазоне значений от 1 до 10. Транспонировать матрицу A и выполнить сортировку столбцов матрицы A^T по возрастанию. Изменить 3-ий элемент 4-ой строки на 1.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 4; \\ 4x_1 + x_2 - x_3 = 4; \\ x_1 + x_2 + 10x_3 = 12. \end{cases}$$

Вариант №3

1. Задать вектор X из 8 значений с клавиатуры. Для вектора X найти максимальный элемент и его порядковый номер. Преобразовать вектор-строку X в вектор-столбец.

2. Задать матрицу A размерностью 4 x 4 с клавиатуры. Прибавить к ней единичную матрицу. Возвести все элементы полученной матрицы в квадрат. Обнулить 4-ый элемент 3-ей строки полученной матрицы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 2; \\ 2x_1 + 3x_2 - 4x_3 = 4; \\ x_1 - 3x_2 + 4x_3 = -3. \end{cases}$$

Вариант №4

1. Задать вектор Y из 8 элементов случайным образом в диапазоне значений от –5 до 0. Увеличить 4-ий элемент вектора Y на 1. Выполнить сортировку элементов вектора и найти произведение его элементов.

2. Задать две матрицы A и B размерностью 5 x 5 с клавиатуры. Найти определитель матрицы B и максимальный элемент для обеих матриц. Найденное значение определителя матрицы B поместить на место 2-го элемента 3 строки матрицы A.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} -2x_1+2x_2-4x_3=1; \\ 3x_1-3x_2+5x_3=4; \\ x_1-3x_2+x_3=2. \end{cases}$$

Вариант №5

1. Задать вектор X, элементы которого представляют собой арифметическую последовательность от 1 до 10 с шагом 1, и вектор Y - от 0.5 до 5 с шагом 0.5. Найти вектор попарных произведений упорядоченных векторов X и Y.

2. Задать матрицу A размерностью 5 x 5 случайным образом в диапазоне значений от 1 до 2. Найти сумму максимальных элементов каждого столбца. Вычислить обратную матрицу. В обратной матрице обнулить первый и последний элементы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} 10x_1+x_3=-11; \\ 2x_1+8x_2-3x_3=5; \\ 10x_1-2x_2-5x_3=-2. \end{cases}$$

Вариант №6

1. Задать векторы X и Y, состоящие из 7 значений, с клавиатуры. Упорядочить вектор X-Y. Найти сумму элементов вектора X и произведение элементов вектора Y. Увеличить на 1 вторые элементы векторов X и Y.

2. Задать матрицу A размерностью 7 x 7 случайным образом в диапазоне значений от 0 до 100. Отнять от нее единичную матрицу. Найти определитель полученной матрицы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} 2x_1+3x_2-x_3=4; \\ 4x_1+x_2-x_3=4; \\ x_1+x_2+10x_3=12. \end{cases}$$

Вариант №7

1. Задать векторы X и Y размерностью 6 случайным образом в диапазоне значений от –10 до 10. Найти минимальный элемент для обоих векторов. Произвести сложение векторов.

2. Задать матрицу A размерностью 4 x 4 с клавиатуры. Обнулить первый и последний элементы 3-ей строки матрицы A. Найти определитель матрицы A^T. Рассчитать сумму всех элементов матрицы A.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} x_1+2x_2-x_3=2; \\ 2x_1-x_2+2x_3=11; \\ 4x_1+x_2-2x_3=1. \end{cases}$$

Вариант №8

1. Задать вектор Y из 9 элементов случайным образом в диапазоне значений от –3 до 6. Изменить 8-ой элемент на 2. Для вектора Y найти максимальный элемент и его порядковый номер.

2. Задать матрицу A размерностью 6 x 6 с клавиатуры. Обнулить второй элемент 4-ой строки матрицы A. Найти определитель обратной матрицы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 2; \\ 2x_1 + 3x_2 - 4x_3 = 4; \\ x_1 - 3x_2 + 4x_3 = -3. \end{cases}$$

Вариант №9

1. Задать векторы X и Y размерностью 7 случайным образом в диапазоне значений от –1 до 1. Найти сумму всех элементов обоих векторов. Произвести попарное деление векторов.

2. Задать две матрицы A и B размерностью 4 x 4 с клавиатуры. Найти определители матриц A и B, произведение матриц A и B, а также сумму всех элементов обеих матриц.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} -2x_1 + 2x_2 - 4x_3 = 1; \\ 3x_1 - 3x_2 + 5x_3 = 4; \\ x_1 - 3x_2 + x_3 = 2. \end{cases}$$

Вариант №10

1. Задать вектор X, элементы которого представляют собой арифметическую последовательность от $\pi/6$ до π с шагом $\pi/12$, и вектор Y - от 1 до 11 с шагом 1. Найти произведение минимального элемента вектора X на максимальный элемент вектора Y. Найти вектор суммы элементов векторов X и Y.

2. Задать матрицу A размерностью 5 x 5 случайным образом в диапазоне значений от -10 до 0. Обнулить второй элемент 3-ей строки. Прибавить к матрице A единичную матрицу. Найти вектор средних значений каждого столбца полученной матрицы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} 10x_1 + x_3 = -11; \\ 2x_1 + 8x_2 - 3x_3 = 5; \\ 10x_1 - 2x_2 - 5x_3 = -2. \end{cases}$$

Вариант №11

1. Задать векторы X и Y, состоящие из 7 значений, с клавиатуры. Упорядочить вектор попарных произведений X на Y. Найти произведение средних значений элементов векторов X и Y. Увеличить на 2 третьи элементы векторов X и Y.

2. Задать матрицу A размерностью 7 x 6 случайным образом в диапазоне значений от 10 до 100. Транспонировать матрицу A и найти минимальный элемент матрицы A^T. Изменить 2-ой элемент 6-ой строки на 3.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 2; \\ 2x_1 + 3x_2 - 4x_3 = 4; \\ x_1 - 3x_2 + 4x_3 = -3. \end{cases}$$

Вариант №12

1. Задать вектор X из 9 значений с клавиатуры. Для вектора X найти минимальный элемент и его порядковый номер. Преобразовать вектор-строку X в вектор-столбец. Обнулить его 3-ий элемент.

2. Задать матрицу B размерностью 6 x 6 случайным образом в диапазоне значений от 1 до 5. Найти произведения средних значений каждого столбца. Транспонировать матрицу.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} -2x_1 + 2x_2 - 4x_3 = 1; \\ 3x_1 - 3x_2 + 5x_3 = 4; \\ x_1 - 3x_2 + x_3 = 2. \end{cases}$$

Вариант №13

1. Задать вектор X, элементы которого представляют собой арифметическую последовательность от 0 до 100 с шагом 10, и вектор Y - от 0 до 100π с шагом 10π. Найти вектор поэлементного деления вектора X на вектор Y.

2. Задать матрицу A размерностью 5 x 6 случайным образом в диапазоне значений от 0.5 до 1. Транспонировать матрицу A и найти вектор средних значений для матрицы A^T. Увеличить 2-ой элемент 3-ей строки на 1.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 4; \\ 4x_1 + x_2 - x_3 = 4; \\ x_1 + x_2 + 10x_3 = 12. \end{cases}$$

Вариант №14

1. Задать вектор Y из 8 элементов случайным образом в диапазоне значений от –5 до 5. Увеличить 3-ий элемент вектора Y на 3. Выполнить сортировку элементов вектора и найти сумму его элементов.

2. Задать матрицу A размерностью 5 x 5 с клавиатуры. Отнять от нее единичную матрицу. Возвести все элементы полученной матрицы в куб. Обнулить 3-ый элемент 2-ой строки полученной матрицы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} -2x_1 + 2x_2 - 4x_3 = 1; \\ 3x_1 - 3x_2 + 5x_3 = 4; \\ x_1 - 3x_2 + x_3 = 2. \end{cases}$$

Вариант №15

1. Задать с клавиатуры вектор X и вектор Y, состоящие из 10 элементов каждый. Найти произведение суммы элементов вектора X на среднее значение элементов вектора Y. Разделить поэлементно вектор X на вектор Y.

2. Задать матрицу A размерностью 6 x 6 случайным образом в диапазоне значений от -50 до 100. Найти определитель матрицы A. Удвоить первый и последний элементы 3-ей строки матрицы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} x_1 - 3x_3 = -13; \\ -2x_1 + 3x_2 + x_3 = 10; \\ 3x_1 - 5x_2 + 2x_3 = 1. \end{cases}$$

Вариант №16

1. Задать вектор Y из 8 элементов случайным образом в диапазоне значений от –30 до -10. Увеличить первый элемент вектора на 2. Для вектора Y найти сумму, произведение и среднее значение его элементов.

2. Задать матрицу A размерностью 3 x 4 и матрицу B размерностью 4 x3 с клавиатуры. Найти матрицу C=A·B. Рассчитать сумму и произведение всех элементов матрицы C. Найти определитель матрицы C.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} -4x_1 + 5x_2 - 3x_3 = -8; \\ 3x_2 - x_3 = 4; \\ 3x_1 + 2x_2 - 2x_3 = 2. \end{cases}$$

Вариант №17

1. Задать вектор Z из 7 значений с клавиатуры. Для вектора Z найти максимальный элемент и его порядковый номер. Транспонировать вектор Z. Обнулить 5-ый элемент вектора Z.

2. Задать матрицу B размерностью 6 x 5 случайным образом в диапазоне значений от 100 до 200. Транспонировать матрицу B. Найти сумму произведений столбцов матрицы. Добавить к матрице B единичную матрицу размерностью 6 x 5.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} 10x_1 - 5x_2 - 3x_3 = 4; \\ -x_1 + x_2 - x_3 = -2; \\ 7x_1 - 2x_2 - 3x_3 = 4. \end{cases}$$

Вариант №18

1. Задать векторы X и Y, состоящие из 8 значений, с клавиатуры. Упорядочить вектор попарных произведений X на Y. Найти произведение средних значений элементов векторов X и Y. Увеличить на 3 третьи элементы векторов X и Y.

2. Задать матрицу A размерностью 4 x 4 случайным образом в диапазоне значений от -3 до -2. Найти сумму минимальных элементов каждого столбца. Вычислить обратную матрицу. В обратной матрице обнулить первый и последний элементы.

3. Решить систему линейных уравнений матричным способом.

$$\begin{cases} x_1 + 2x_2 - x_3 = 2; \\ 2x_1 - x_2 + 2x_3 = 11; \\ 4x_1 + x_2 - 2x_3 = 1. \end{cases}$$

Лабораторная работа № 3

РАБОТА С ГРАФИЧЕСКОЙ ПОДСИСТЕМОЙ MATLAB

Цель работы: получить навыки работы с двумерными и трехмерными графиками; освоить приемы оформления графиков.

Задание

Разбить графическое окно MatLab на 6 частей. В каждом из окон построить следующие графики:

а) $y = f(x)$, $z = f(x)$ в обычном масштабе; нанести сетку, легенду, название графика и осей координат;

б) $y = f(x)$ в логарифмическом масштабе по X и Y;

в) гистограмму значений функции z;

г) график функции $\rho(\varphi)$ в полярных координатах, $\varphi = [0..2\pi]$, $\Delta\varphi = \pi/16$;

д) график функции $y = f(x_1, x_2)$ в виде прозрачной сетки (mesh); нанести названия всех осей x_1, x_2 и y;

е) $y = f(x_1, x_2)$ в виде контурных линий (contour).

Методические указания

Построение двумерного графика зависимости вектора y от вектора x осуществляется посредством функции plot(x,y), например,

```
>> x=[0:pi/10:4*pi]; y=sin(x); plot(x,y,'b')
```

При этом построение графика производится в отдельном окне Windows.

При работе с графическими средствами в программе MATLAB специализированные функции помещают графики, а также специальные знаки и надписи на них в отдельные графические окна. Типичный вид такого окна представлен на рисунке 3.1.

Кнопки управления видом графического окна и их назначение представлены на рисунке 3.2.

Программа MatLab позволяет редактировать вид графического окна с помощью кнопок управления графическим окном. Кроме того, можно строить различные графики в режиме реального времени. Для задания этого режима можно ввести команду plottools или нажать кнопку “Отображение графика” на предыдущем рисунке.

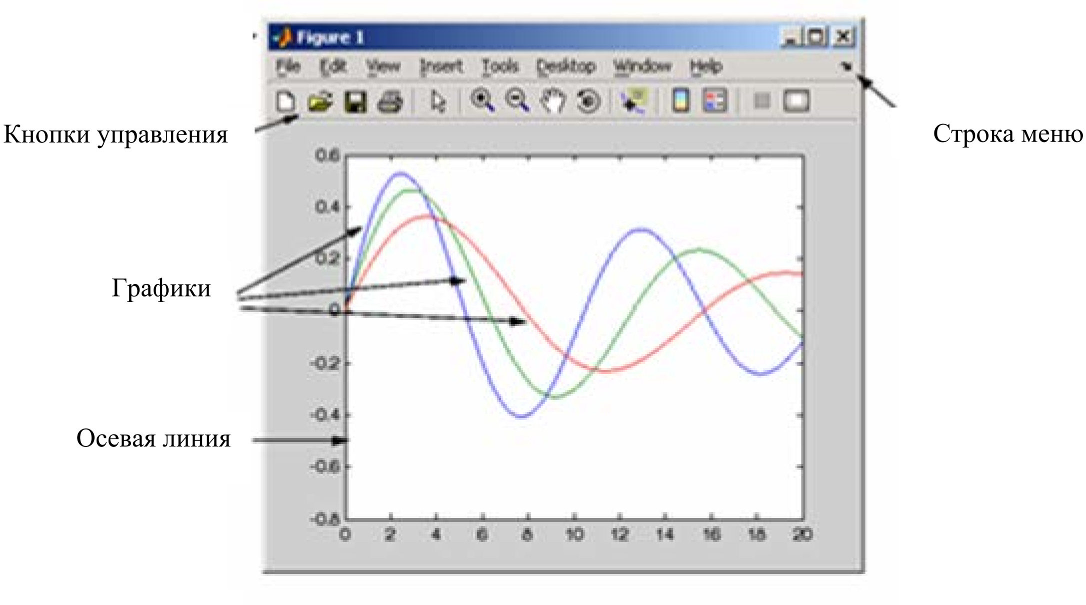

Рис. 3.1. Графическое окно Matlab

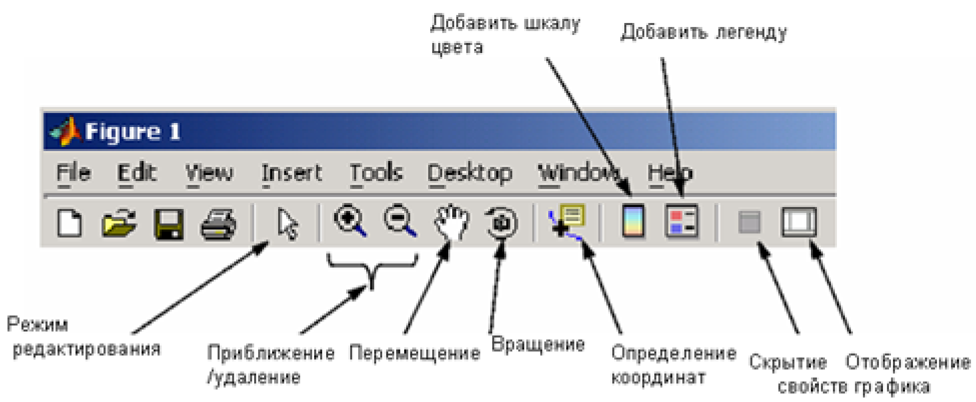

Рис. 3.2. Кнопки управления в системе Matlab

Интерфейс режима управления графиками включает три панели (рис. 3.3).

1. Палитра графиков (Figure Palette) – применяется для редактирования осевых линий графиков, вида рабочей области графика, а также для добавления пояснений. Для

отображения палитры графиков можно также использовать специальную команду figurepalette.

2. Браузер графиков (Plot Browser) – применяется для выбора и управления графиками. Также можно добавить данные путем нажатия кнопки Add Data. Для вызова браузера графиков можно использовать команду plotbrowser.

3. Редактор свойств (Property Editor) – применяется для установления одинаковых свойств различным объектам. Для вызова редактора можно применить команду propertyeditor.

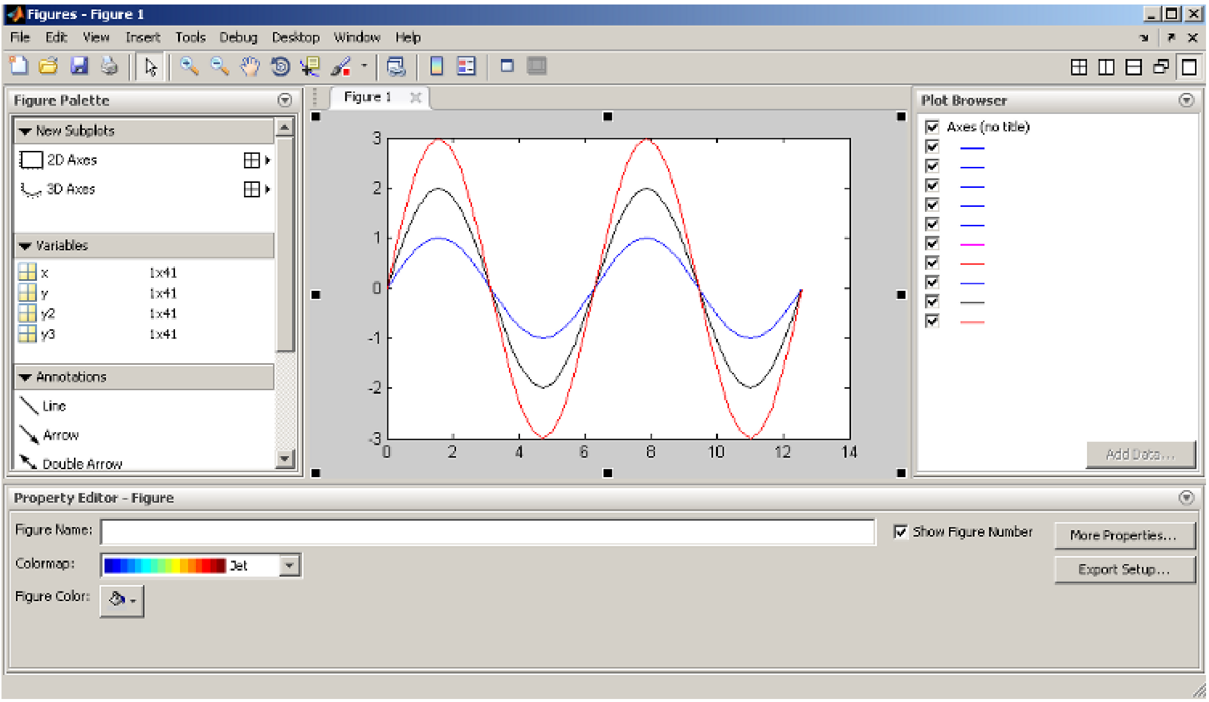

Рис. 3.3. Интерфейс режима управления графиками

Рассмотрим алгоритм построения графика в интерактивном режиме.

1) Зададим два вектора, используя команды:

```
x=-pi:pi/20:pi;
y=sin(x);
```

2) Задаем команду plottools в результате чего появится окно, представленное на рисунке 4.

3) Выбираем двухмерный вид графика на палитре графиков, нажав кнопку 2D Axes.

4) Нажимаем кнопку Add Data в окне браузера графиков и выбираем источники данных так, как показано на рисунке.

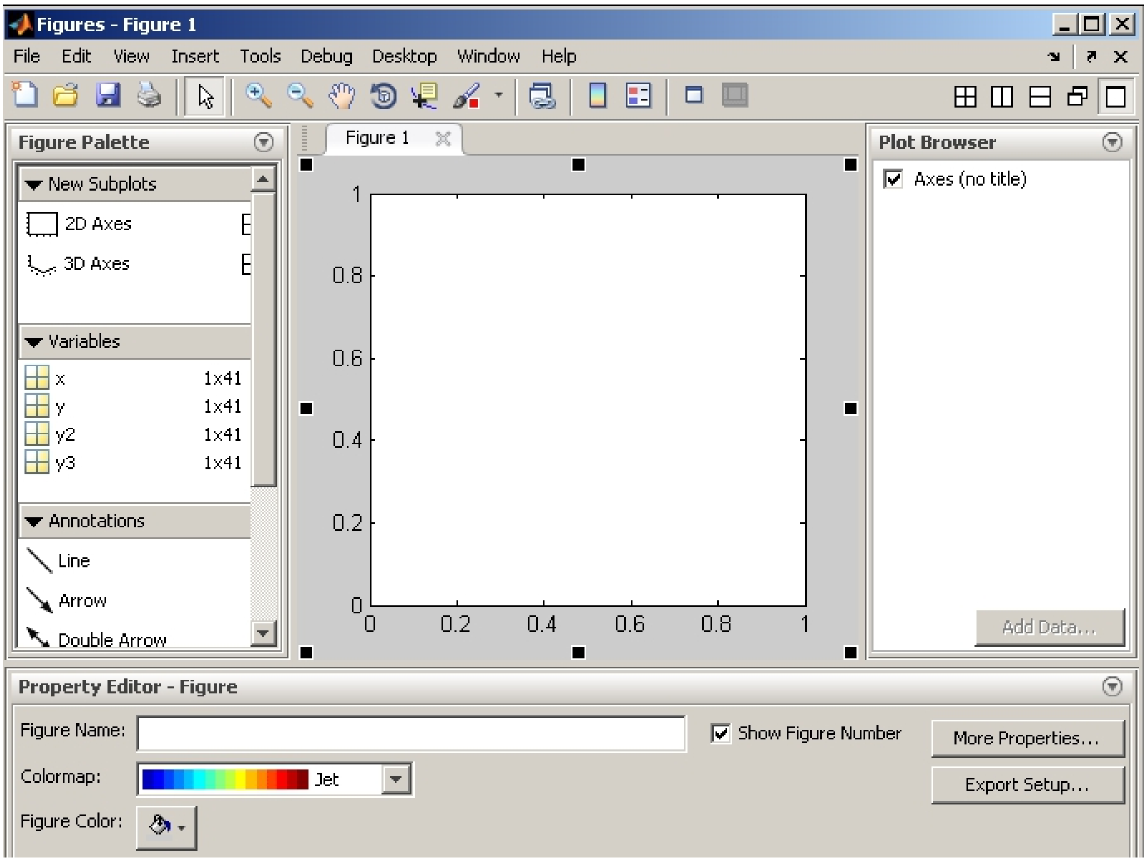

Рис. 3.4. Окно управления графиками

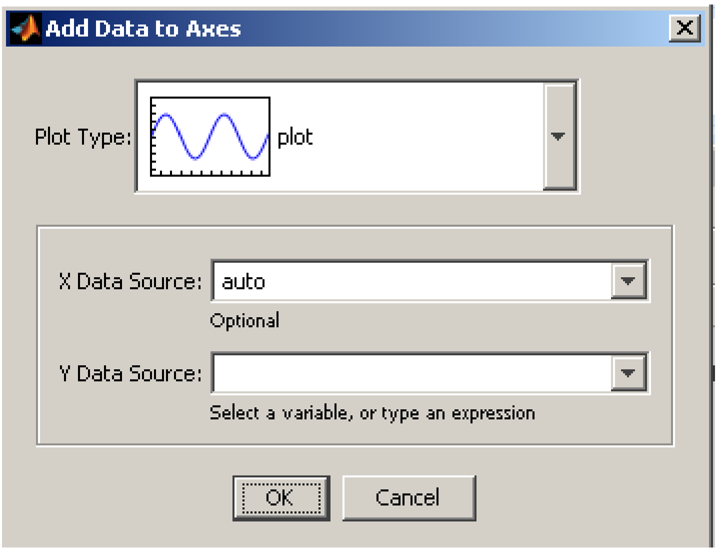

Рис. 3.5.

После нажатия кнопки OK появится график, представленный на рисунке 3.6.

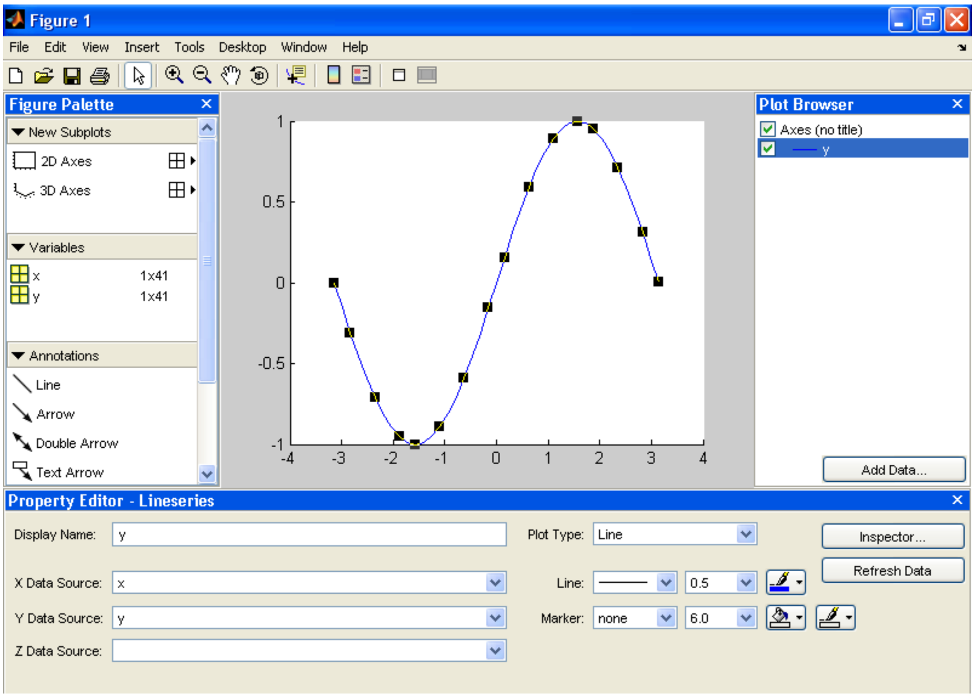

Рис. 3.6.

В режимы программы команда plot(x,y,s) позволяет при помощи строкового параметра s указать способ отображения точек графика. Возможные значения элементов параметра s указаны в таблице 3.1.

Таблица 3.1

Цвет		Тип точки		Тип линии	
y	– желтый	.	– точка	-	– сплошная линия
m	– сиреневый	o	– кружок	--	– пунктирная линия
c	– голубой	x	– крестик	-.	– штрих-пунктирная линия
r	– красный	*	– звездочка	:	– штриховая линия
b	– синий	+	– плюс		
g	– зеленый	s	– квадрат		
w	– белый	d	– ромб		
k	– черный	v	– треугольник (вниз)		
		^	– треугольник (вверх)		
		<	– треугольник (влево)		
		>	– треугольник (вправо)		
		p	– 5-угольные звездочки		
		h	– 6-угольные звездочки		

Например,

>> plot(x,y,'r') – нарисовать красным цветом (линия сплошная без точек);

>> plot(x,y,'k*') – нарисовать черным цветом звездочками (линии нет);

```
>> plot(x,y,'g--') – нарисовать зеленым цветом пунктирной линией (точек
                    нет);
>> plot(x,y,'y-.h') – нарисовать желтым цветом штрих-пунктирной линией 6-
угольными звездочками.
```

Команда plot(x1,y1,s1,x2,y2,s2,x3,y3,s3,...) позволяет объединить на одном графике несколько функций y1(x1), y2(x2), y3(x3), ..., определив для каждой из них свой способ отображения посредством строковых переменных s1, s2, s3,... Например,

```
>> x=[1:10]; plot(x, sqrt(x), 'b:', x, x.^2, 'c-.')
```

Аналогичную функцию выполняет и команда hold on, которая включает режим сохранения текущего графика и обеспечивает добавление к нему новых графиков.

Команда hold off выключает режим накопления графиков.

Для окончательного оформления графика служат следующие операторы.

Оператор text(x, y, 'текст') выводит надпись "текст" в точке, определяемой координатами x и y.

Оператор title ('Заголовок') выводит текст "Заголовок" вверху графического окна в качестве заголовка. Текст автоматически центрируется в строке.

Операторы xlabel('Ось X') и ylabel('Ось Y') обеспечивают вывод на графике названия осей «Ось X» и «Ось Y».

Команда grid on наносит пунктирную координатную сетку на текущие оси, а команда grid off – удаляет координатную сетку.

Оператор legend('текст1', 'текст2', 'текст3', ...) добавляет к графику легенду (пояснение), в котором параметры 'текст1', 'текст2', 'текст3', ... являются поясняющими надписями к зависимостям, представленным на общем графике.

По умолчанию при построении графиков используется автоматическое масштабирование осей координат. Принудительное задание масштабов по осям X и Y обеспечивает оператор axis ([x_min x_max y_min y_max]), где вектор из четырех параметров определяет минимальные и максимальные значения по X и по Y. Оператор axis ('square') устанавливает равные масштабы по обеим осям.

Для очистки экрана используются функции: clc – очистка текстового экрана от предыдущих вычислений и clf – очистка графического экрана от ранее построенных графиков.

Пример программы построения графиков функций $y_1 = \sqrt{x}$ и $y_2 = x^2$ (рис. 3.7).

```
clf
x=[1:10]; plot(x,sqrt(x), 'r--')
hold on
axis([1 10 1 100])
plot(x,x.^2, 'b-')
grid  on
title('Графики корня и квадрата числа')
xlabel('X');ylabel('Y')
legend('корень', 'квадрат')
text(1,1,'точка пересечения')
hold off
```

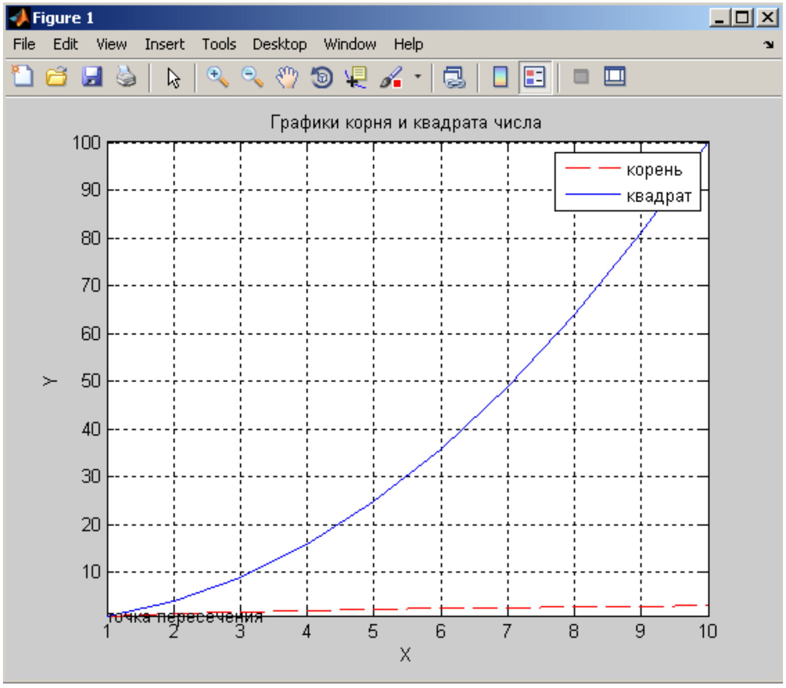

Рис. 3.7. Пример оформления графика

Если вместо функции plot (x, y, s) использовать функций loglog (x, y, s), то построение двумерного графика будет производиться в логарифмическом масштабе и по оси X, и по оси Y. Например,

```
>> x=1:0.5:100; y=x.^2; loglog (x, y, 'w+')
```

Построение графика в полулогарифмическом масштабе производится функциями semilogx (x, y, s) (log по оси X) и semilogy (x, y, s) (log по оси Y).

Для построения графика функции в полярных координатах применяется функция polar (phi, ro, s), где phi – угол в радианах, ro – полярный радиус. Например (рис. 3.8),

```
>> clf; phi=[0:pi/100:2*pi];
>> polar (phi, 1-(sin(phi).^2), 'r')
```

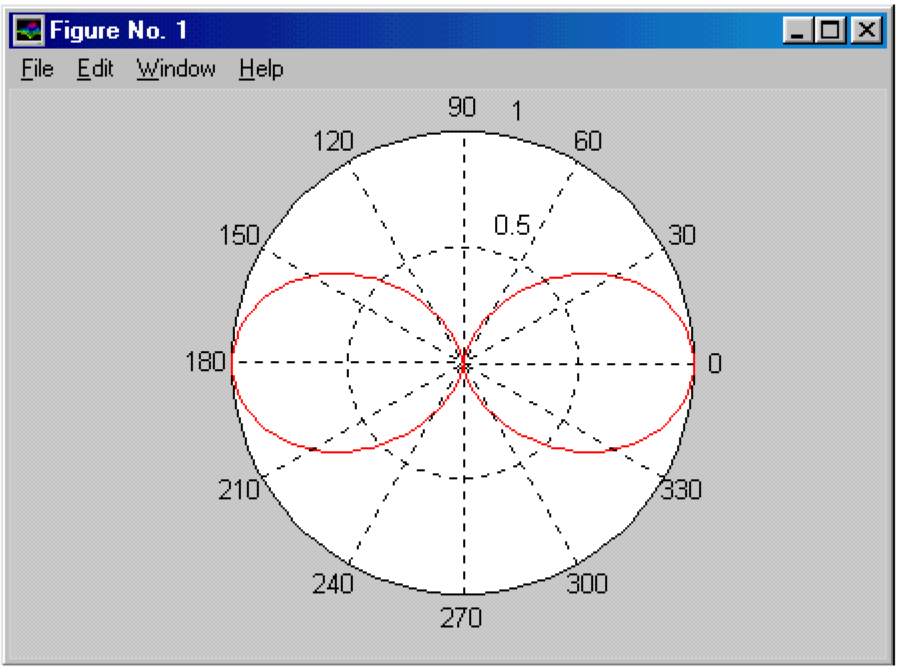

Рис. 3.8. Пример построения графика в полярных координатах

Для построения гистограмм в MatLab можно воспользоваться оператором bar (z). Он строит прямоугольники, высота которых пропорциональна значениям компонент вектора z, а их число определяется числом компонент этого вектора.

MatLab предоставляет возможность разбиения графического экрана на несколько частей и вывода в каждой из них отдельного графика или гистограммы. Для этого применяется команда subplot (m, n, p), где m определяет, на сколько частей разбивается экран по вертикали, n – по горизонтали, p – номер окна, куда будет выводиться очередной график. Например (рис. 3.9),

```
clc; clf
x=[1:10];
subplot(2,1,1); plot(x, sqrt(x))
subplot(2,1,2); bar(x)
```

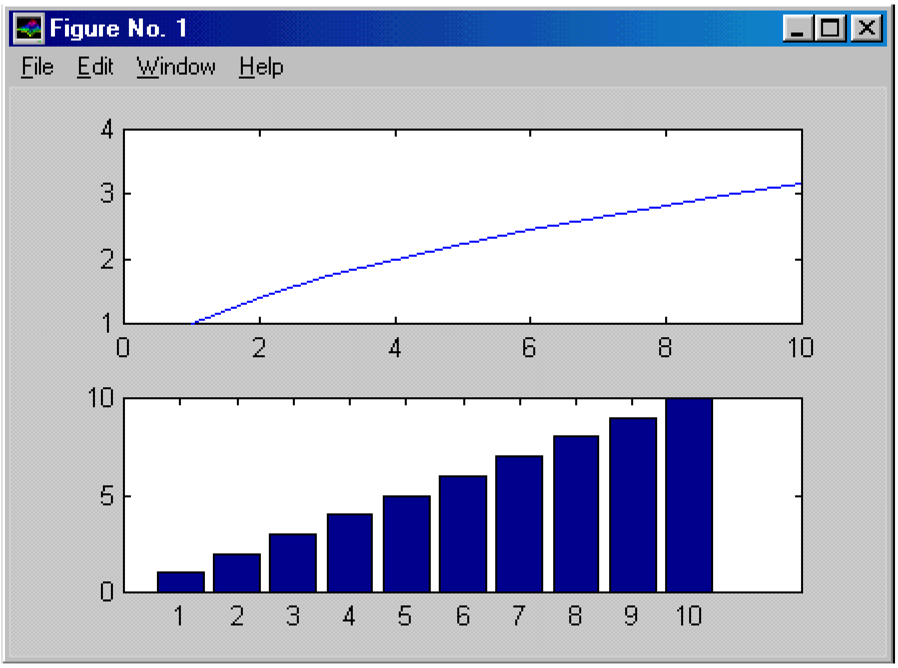

Рис. 3.9. Пример использования команды subplot

Таким образом, команда subplot (m, n, p) осуществляет переключение между частями экрана. Так, например, чтобы вывести координатную сетку в первой части экрана, необходимо выполнить следующие команды:

```
>> subplot(2,1,1); grid on
```

при этом во второй части экрана сетка не появится.

Для построения трехмерных графиков функций $z = f(x, y)$ сначала необходимо задать диапазоны изменения координат x и y, а затем определить сетку, представляющую собой совокупность всех возможных парных сочетаний значений x и y. Это осуществляется при помощи функции meshgrid. Например, пусть переменные $x1$ и $y1$ принимают значения 0 или 1, тогда в результате мы получим сетку из четырех пар значений:

```
>> x1=[0 1]; y1=[0 1];
>> [x, y]=meshgrid(x1, y1)
   x =
       0    1
       0    1
   y =
```

0	0
1	1

После этого на полученной координатной сетке рассчитывается функция *z* = *f*(*x*, *y*), а окончательный вывод трехмерного графика производится функцией mesh(x,y,z). Например (рис. 3.10), требуется построить график функции $z = x^2 - y^2$, в области изменения $x = [-2;2]$ и $y = [-2;2]$.

```
>> x1=[-2:0.1:2]; y1=[-2:0.1:2];
>> [x, y]=meshgrid (x1, y1);
>> z=x.^2-y.^2;
>> mesh (x, y, z)
```

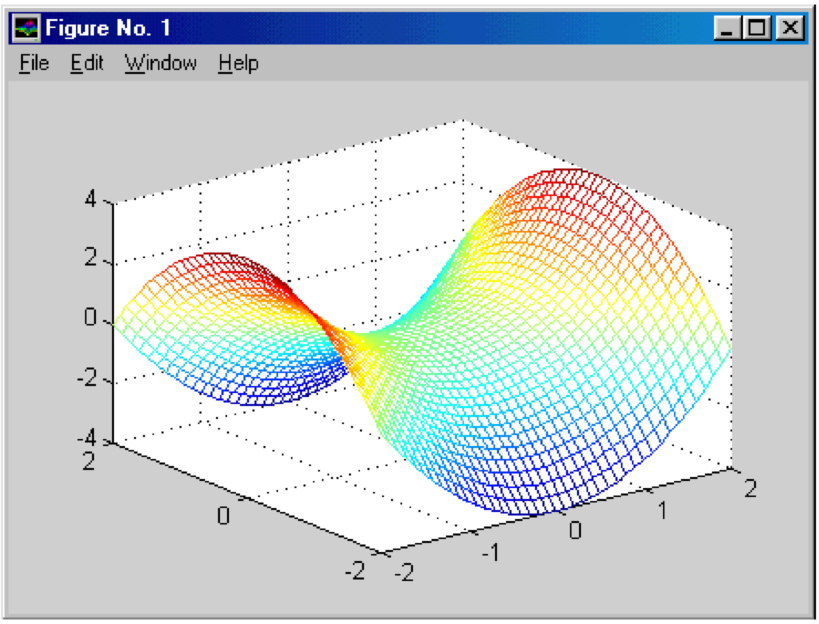

Рис. 3.10. Пример построения трехмерного графика

Поворот построенного трехмерного графика осуществляется при помощи оператора view(az,el), где az – угол горизонтального поворота (по часовой стрелке), el – угол вертикального возвышения (оба – в градусах.)

По умолчанию az = -37.5, el = 30. Для возврата к этому режиму используется оператор view(3).

Поскольку трехмерный график не имеет одного определенного цвета, то в нем производится изменение целиком цветовой гаммы. Для этого применяется оператор colormap(map), где map – имя конкретной цветовой гаммы. В MatLab возможны следующие значения параметра map: hsv, hot, gray, bone, copper, pink, white, flag, lines, colorcube, jet, prism, cool, autumn, spring, winter, summer. Например,

```
>> colormap(gray)
```

Оператор mesh(x,y,z) строит трехмерную поверхность в виде прозрачной сетки, а аналогичный ему оператор surf(x,y,z) – в виде непрозрачной сетки. Для построения контурных линий трехмерных поверхностей, часто именуемых линиями уровня, используется оператор contour(x,y,z). Построение контурных линий особенно полезно для многоэкстремальных функций. Обычный трехмерный график в этом случае иногда не нагляден, так как один пик фигуры может скрывать другой, и он будет невидим. Совмещение контурных линий с самой поверхностью производится оператором meshc(x,y,z). Пример использования этих четырех операторов для одной функции приведен на рис. 3.11.

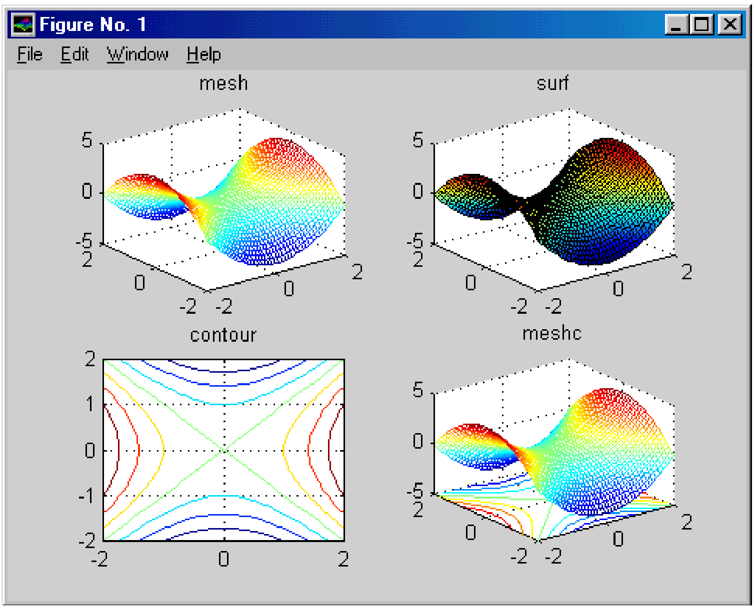

Рис. 3.11. Графики функции $z = x^2 - y^2$, построенные при помощи операторов mesh, surf, contour и meshc

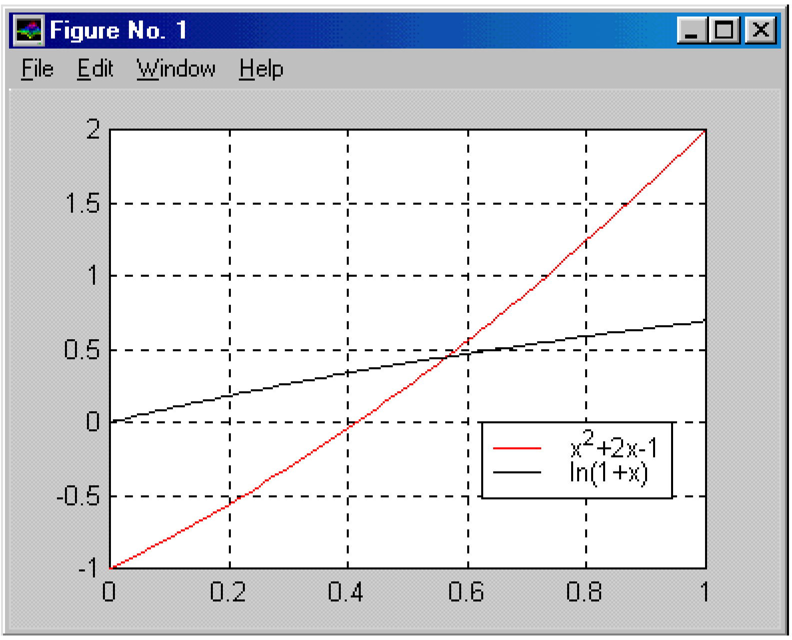

Рис. 3.12. Графическое решение уравнения $x^2+2x-1-\ln(x+1)=0$

Варианты заданий

Таблица 3.2 (окна а, б, в, г)

№	**Функции** $y=f(x)$ **и** $z=f(x)$	**Диапазон и шаг изменения аргумента** x	**Функция** $\rho(\varphi)$
1	$y=\sin(x);\ z=\sin(2x+10)$	$x\in[0..3\pi],\ \Delta x=\pi/6$	$\rho(\varphi)=1-\cos^2\varphi\ \sin^2\varphi$
2	$y=2x+10;\ z=\sqrt{x}+2$	$x\in[0..10],\ \Delta x=0{,}5$	$\rho(\varphi)=1-\sin^2\varphi$
3	$y=e^{-x};\ z=e^{x}$	$x\in[0..2],\ \Delta x=0{,}05$	$\rho(\varphi)=1-\cos^2\varphi$
4	$y=\cos(3x);\ z=\sin(3x)$	$x\in[0..2\pi],\ \Delta x=\pi/8$	$\rho(\varphi)=1-2\sin^2\varphi$
5	$y=e^{x/10};\ z=e^{(x/10-1)}$	$x\in[10..20],\ \Delta x=0{,}1$	$\rho(\varphi)=1-\cos^2(\varphi/2)\ \sin^2\varphi$
6	$y=\sqrt{10x};\ z=2\sqrt{5x}$	$x\in[10..20],\ \Delta x=0{,}1$	$\rho(\varphi)=1-\sin^2(\varphi/2)$
7	$y=2x^2-3x+1;$ $z=x^2-2$	$x\in[0..10],\ \Delta x=0{,}1$	$\rho(\varphi)=1-\cos^2(\varphi/2)$
8	$y=\mathrm{tg}(x);\ z=\mathrm{tg}(x/2)$	$x\in[0..\pi/4],\ \Delta x=\pi/64$	$\rho(\varphi)=1-\sin^2(2\varphi)$
9	$y=1/\sqrt[3]{x};\ y=1/\sqrt[4]{x}$	$x\in[5..10],\ \Delta x=0{,}2$	$\rho(\varphi)=\varphi\ \sin^2\varphi$
10	$y=\log(x);\ z=e^{-x/10}$	$x\in[1..6],\ \Delta x=0{,}2$	$\rho(\varphi)=\varphi\ \cos^2\varphi$
11	$y=\mathrm{tg}(x);\ z=\mathrm{th}(x)$	$x\in[0..\pi/4],\ \Delta x=\pi/32$	$\rho(\varphi)=1-\varphi\ \sin^2\varphi$
12	$y=\ln(x);\ z=\log_{10}(x)$	$x\in[1..2],\ \Delta x=0{,}05$	$\rho(\varphi)=2\sin^2\varphi\ -1$
13	$y=\mathrm{sh}(x);\ z=\mathrm{ch}(x)$	$x\in[0..4\pi],\ \Delta x=\pi/16$	$\rho(\varphi)=1-2\cos^2\varphi\ \sin^2\varphi$
14	$y=\lvert x\rvert;\ z=2x+4$	$x\in[-10..10],\ \Delta x=0{,}1$	$\rho(\varphi)=1-4\sin^2\varphi$
15	$y=x^2-3x+2;$ $z=x^3-x^2$	$x\in[-5..5],\ \Delta x=0{,}01$	$\rho(\varphi)=\varphi^2+\cos\varphi$
16	$y=\dfrac{e^{-x^2}}{x+2};\ z=\dfrac{x}{6}$	$x\in[0..3],\ \Delta x=0{,}005$	$\rho(\varphi)=\varphi-\sin\varphi$
17	$y=\dfrac{1}{x-2};\ z=-x-2$	$x\in[-10..0],\ \Delta x=0{,}01$	$\rho(\varphi)=\sqrt{\cos\varphi}$
18	$y=\dfrac{1}{2x+4};\ z=1+x$	$x\in[-10..10],\ \Delta x=0{,}02$	$\rho(\varphi)=\sqrt[3]{\sin\varphi}$

Таблица 3.3 (окна д, е)

№	**Функции** $y = f(x_1, x_2)$	**Диапазон и шаг изменения аргументов** x_1 **и** x_2
1	$[1{,}5 - x_1(1 - x_2)]^2 + \left[2{,}25 - x_1\left(1 - x_2^2\right)\right]^2 + \left[2{,}625 - x_1\left(1 - x_2^3\right)\right]$	$x_1 \in [0..5]$, $x_2 \in [-2..2]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
2	$x_1^3 + x_2^2 - 3x_1 - 2x_2 + 2$	$x_1 \in [-3..3]$, $x_2 \in [-3..3]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
3	$1 - 2x_1 - 2x_2 - 4x_1x_2 + 10x_1^2 + 2x_2^2$	$x_1 \in [-2..2]$, $x_2 \in [-2..2]$, $\Delta x_1 = 0{,}05$, $\Delta x_2 = 0{,}05$
4	$x_1^4 + x_2^4 + 2x_1^2x_2^2 - 4x_1 + 3$	$x_1 \in [-2..3]$, $x_2 \in [-2..2]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
5	$\left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$	$x_1 \in [0..5]$, $x_2 \in [0..5]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
6	$x_1^3 + x_2^2 - 3x_1 - 2x_2 + 2$	$x_1 \in [-5..5]$, $x_2 \in [-5..5]$, $\Delta x_1 = 0{,}2$, $\Delta x_2 = 0{,}2$
7	$100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$	$x_1 \in [-2..3]$, $x_2 \in [-2..3]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
8	$\left[\left(e^{-0{,}1x_1} - e^{-0{,}1x_2}\right) - \left(e^{-0{,}1} - e^{-1}\right)\right]^2$	$x_1 \in [-3..3]$, $x_2 \in [5..15]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}2$
9	$[1{,}5 - x_1(1 - x_2)]^2 + \left[2{,}25 - x_1\left(1 - x_2^2\right)\right]^2 + \left[2{,}625 - x_1\left(1 - x_2^3\right)\right]$	$x_1 \in [-1..4]$, $x_2 \in [-3..3]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
10	$1 - 2x_1 - 2x_2 - 4x_1x_2 + 10x_1^2 + 2x_2^2$	$x_1 \in [-4..4]$, $x_2 \in [-3..3]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
11	$x_1^4 + x_2^4 + 2x_1^2x_2^2 - 4x_1 + 3$	$x_1 \in [-3..3]$, $x_2 \in [-2..3]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
12	$\left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$	$x_1 \in [1..6]$, $x_2 \in [1..6]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
13	$100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$	$x_1 \in [-3..3]$, $x_2 \in [-3..3]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$
14	$\left[\left(e^{-0{,}2x_1} - e^{-0{,}2x_2}\right) - \left(e^{-0{,}2} - e^{-2}\right)\right]^2$	$x_1 \in [-4..4]$, $x_2 \in [3..15]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}2$
15	$(x_1 - 2)^2 + (x_2 - 5)^2$	$x_1 \in [0..5]$, $x_2 \in [0..10]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}1$

Продолжение таблицы 3.3 (окна д, е)

№	**Функции** $y = f(x_1, x_2)$	**Диапазон и шаг изменения аргументов** x_1 **и** x_2
16	$x_1^2 + 4x_1x_2 + 5x_2^2$	$x_1 \in [-5..5]$, $x_2 \in [-5..5]$, $\Delta x_1 = 0{,}05$, $\Delta x_2 = 0{,}05$
17	$3(x_1 - 4)^2 + 5(x_2 + 3)^2$	$x_1 \in [0..10]$, $x_2 \in [-5..1]$, $\Delta x_1 = 0{,}1$, $\Delta x_2 = 0{,}05$
18	$\frac{x_1^2}{5} + \frac{x_2^2}{3}$	$x_1 \in [-5..5]$, $x_2 \in [-7..7]$, $\Delta x_1 = 0{,}01$, $\Delta x_2 = 0{,}05$

Лабораторная работа № 4

ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЯЮЩЕЙСЯ И ЦИКЛИЧЕСКОЙ СТРУКТУРЫ В MATLAB

Цель работы: получить практические навыки разработки и программирования вычислительного процесса линейных, разветвляющихся и циклических структур в системе Matlab; получить навыки по отладке и тестированию программ.

Задание

1. Вычислить значение функции, заданной в таблице 4.1 (в соответствии с вариантом). Осуществить вывод значений вводимых исходных данных и результат вычисления значения функции, сопровождая вывод наименованиями переменных.

2. Написать программу для вычисления значения суммы членов бесконечного ряда (см. табл. 4.2) с заданной точностью. На печать вывести значение суммы и число членов ряда, вошедших в сумму.

Методические указания

Для решения сложных задач режима прямых вычислений недостаточно. Возникает необходимость сохранять используемые последовательности вычислений и модифицировать их. Иными словами, нужно программировать решение задачи.

Программа в MatLab представляет собой запись в m-файле (файл с расширением ".m") всей той последовательности команд, которые потребовались бы для решения задачи в режиме прямых вычислений. Запуск на выполнение m-файла осуществляется введением в строке его имени без расширения и нажатием клавиши "Enter".

Ввод данных с клавиатуры в программе организуется при помощи оператора **input**, а вывод на дисплей – **disp**.

Например,

```
r = input ('Введите r');
```

```
disp (r)
```

Кроме того, управление выводом информации на экран может осуществляться посредством символа ";".

В системе MatLab оператор ветвления имеет следующий сокращенный вид:

```
if Условие
    Операторы
end
```

а в общем случае записывается следующим образом:

```
if Условие
    Операторы1
else
    Операторы2
end
```

В первом варианте "Операторы" выполняются только в том случае, если условие истинно, и не выполняются если ложно. Во втором варианте если условие истинно, то выполняется группа операторов "Операторы1", в противном случае, выполняется группа операторов "Операторы2".

Для организации циклов в программах используются операторы **for ... end** (со счетчиком) и **while ... end** (с предварительной проверкой условия). Первый из них записывается следующим образом:

```
for Счетчик = Нач_знач : Шаг : Кон_знач
      Тело цикла (операторы)
end
```

где *Счетчик* – имя переменной параметра цикла (не обязательно целочисленный тип);

Нач_знач и *Кон_знач* – начальное и конечное значение переменной цикла;

Шаг – задает величину приращения переменной *Счетчик* при ее изменении от *Нач_знач* до *Кон_знач*.

Например, задать случайным образом вектор X из 10 элементов и подсчитать в нем число положительных элементов.

```
X = rand(1, 10)-0.5
n = 0;
for i=1:1:10
  if X(i)>0
    n=n+1;
  end
end
disp (n)
```

Конструкция цикла **while ... end** имеет вид:

```
while Условие
  Тело цикла (операторы)
end
```

В данном случае операторы цикла будут выполняться, пока будет истинно условие "Условие".

Центральным моментом программирования в среде MatLab является задание новых функций. Для этого используется конструкция

```
function Имя_переменной = Имя_функции (Список_аргументов)
% Комментарий, выводимый при использовании команды help
     Операторы % Комментарии
      Имя_переменной=Выражение
```

Все переменные, используемые внутри блока функции, являются локальными, т.е. область их действия распространяется только на блок функции.

Для того, чтобы создать свою функцию необходимо войти в любой текстовый редактор (например, в FAR – клавиши "Shift"+"F4"), ввести весь текст с описанием функции и сохранить его в файле с именем "Имя_функции.m". Например, создав запись

```
function y =f unc(x);
  y = sqrt(2*x+1);
```

и сохранив ее в файле "func.m", мы получаем доступ к этой функции ($y=\sqrt{2x+1}$), как, скажем, к стандартной функции sin(x), т.е. следующая команда даст результат:

```
>> z = func(4)
  z =
     3
```

Пример. Вычисление факториала числа. В файле с именем "fact.m" создаем запись:

```
function f = fact(x)
% fact(x) - функция вычисления факториала x!
  f = 1;
  if x > 0
    for i = 1:x
      f = f*i;
    end
  end
```

Первая строка с комментарием (%) будет выводиться на экран в результате выполнения команды

```
>> help Имя_функции
то есть
>> help fact
  fact(x) - функция вычисления факториала x!
```

Пример определения функции трех переменных $y=\frac{x_1^2}{2}+\frac{x_2^3}{3}+\frac{x_3^3}{4}$.

```
function y = func3(x)
% Файл func3.m
    y = x(1)^2/2+x(2)^3/3+x(3)^4/4;
```

Тогда вызов функции может быть осуществлен следующими способами:

```
>> x=[1 2 3]; func3(x)
    ans =
          23.4167
```

или

```
>> func3([1 2 3])
    ans =
          23.4167
```

Для изменения формата представления действительных чисел используется команда

```
>> format Тип_формата
```

В MatLab доступны следующие форматы:

short	– короткое представление в фиксированном формате (4 знака после десятичной запятой)
short e	– короткое представление в экспоненциальном формате (4 знака после запятой и 3 знака порядка)
long	– длинное представление в фиксированном формате (14 знаков после запятой)
long e	– длинное представление в экспоненциальном формате (14 знаков после запятой и 3 знака порядка)

Например,

```
>> format short; x = 4/3
    x =
        1.3333
>> format short e; x = 4/3
    x =
        1.3333e+000
```

Примеры загрузки файлов данных

Часто в практике применения программ на языке MatLab требуется обрабатывать большие массивы данных, созданных и записанных в файлы. Для обработки данных, содержащихся в файлах, их необходимо сначала загрузить в вектора MatLab. Это осуществляется путем добавления в код программы следующего оператора

```
T = load('heat1.txt');
```

В функции **load** в кавычках указано имя загружаемого файла данных **'heat1.txt'**. Следует помнить, что загружаемые данные должны быть соответствующим образом подготовлены. Например, если данные содержат несколько строк, в каждой из которых содержится несколько столбцов, то число столбцов в каждой строке должно совпадать.

Как видно из предыдущего примера, данные будут загружены в переменную с именем Т. После загрузки к данным можно обращаться обычным способом. Например, пусть загружаемый файл 'heat1.txt' содержал 5 столбцов данных. Тогда после его загрузки данные первого столбца можно считать следующим образом T(:,1), а данные 5-го столбца T(:,5).

Рассмотрим пример программы, которая в исходном текстовом файле отыскивает число строк, число символов в строке, а также не пустую строку с минимальным числом символов.

Вначале программы в блоке

```
if(nargin~=1)
    error('Число входных параметров не равно 1')
end
if ~ischar(filename)
    error('Входной параметр должен быть строкой')
end
```

осуществляются проверки числа входных параметров в функцию, а также, является ли входной параметр (имя файла) строкой.

Функция fopen(filename,'r') предназначена для открытия файла в режиме чтения (обратите внимание, что данные не загружаются в программу в отличие от функции load()).

Функция feof(fid) предназначена для проверки факта достижения конца файла.

Внимательно проанализируйте представленную ниже программу и зарисуйте алгоритм ее выполнения.

Программа

```
function [line_number,char_number]=empty(filename)
if(nargin~=1)
    error('Число входных параметров не равно 1')
end
if ~ischar(filename)
    error('Входной параметр должен быть строкой')
end
[fid, mes]=fopen(filename,'r');
if(fid==-1)
    disp('ошибка открытия файла')
    error(mes)
else
    line_number=0;
    char_number=0;
    line_empty=0;
    line_index=1;
    char_old=1000;
    char_tek=0;
        while ~(feof(fid))
        string=fgetl(fid);
        line_number=line_number+1;
        char_number=char_number+size(string,2);
```

```
            char_tek=size(string,2);
            if char_tek<char_old && char_tek~=0
                char_old=char_tek;
                line_index=line_number;
            end
                   if strmatch(string,'')
                disp(' ')
                line_empty=line_empty+1;
            else
                disp(string)
            end
        end
        fclose(fid);
        disp('Lines empty=')
        disp(line_empty)
        disp('Minimum number=')
        disp(char_old)
    end
```

Варианты заданий

Таблица 4.1

Вариант	Функция	Условие	Исходные данные	Диапазон и шаг изменения аргумента
1	2	3	4	5
1	$Y=\begin{cases} at^2\cdot\ln t \\ 1 \\ e^{at}\cdot\cos bt \end{cases}$	$1\le t\le 2$ $t<1$ $t>2$	$a=-0.5$ $b=2$	$t\in[0;3]$ $\Delta t=0.15$
2	$Y=\begin{cases} \pi x^2-\dfrac{7}{x^2} \\ ax^3+7\sqrt{x} \\ \lg\left(x+7\sqrt{x}\right) \end{cases}$	$x<1.3$ $x=1.3$ $x>1.3$	$a=1.5$	$x\in[0.8;2]$ $\Delta x=0.1$
3	$W=\begin{cases} ax^2+bx+c \\ \dfrac{a}{x}+\sqrt{x^2+1} \\ \dfrac{(a+bx)}{\sqrt{x^2+1}} \end{cases}$	$x<1.2$ $x=1.2$ $x>1.2$	$a=2.8$ $b=-0.3$ $c=4$	$x\in[1;2]$ $\Delta x=0.05$
4	$Q=\begin{cases} \pi x^2-\dfrac{7}{x^2} \\ ax^3+7\sqrt{x} \\ \ln\left(x+7\sqrt{\lvert x+a\rvert}\right) \end{cases}$	$x<1.4$ $x=1.4$ $x>1.4$	$a=1.65$	$x\in[0.7;2]$ $\Delta x=0.1$
5	$Y=\begin{cases} 1{,}5\cdot\cos^2 x \\ 1{,}8\cdot ax \\ (x-2)^2+6 \\ 3\cdot tg\, x \end{cases}$	$x<1$ $x=1$ $1<x\le 2$ $x>2$	$a=2.3$	$x\in[0.2;2.8]$ $\Delta x=0.2$
6	$W=\begin{cases} e^{-ax}\cdot\cos ax \\ x\cdot\sin ax \\ x\cdot\sqrt[3]{x-a} \end{cases}$	$x<a$ $x=a$ $x>a$	$a=2.5$	$x\in[1;5]$ $\Delta x=0.5$

Продолжение таблицы 4.1

1	2	3	4	5
8	$Y=\begin{cases}\sin x\cdot\lg x\\ \cos^2 x\end{cases}$	$x>3.5$ $x\le 3.5$		$x\in[2;5]$ $\Delta x=0.25$
9	$F=\begin{cases}\lg(x+1)\\ \sin^2\sqrt{ax}\end{cases}$	$x>1$ $x\le 1$	$a=20.3$	$x\in[0.5;2]$ $\Delta x=0.2$
10	$Z=\begin{cases}\cos x+t\cdot\sin^2 x\\ \dfrac{1}{x}+\sqrt{x+t}\\ \dfrac{\left(\ln^3 x+x^2\right)}{\sqrt{x+t}}\end{cases}$	$x>0.5$ $x=0.5$ $x<0.5$	$t=2.2$	$x\in[0.2;2]$ $\Delta x=0.2$
11	$S=\begin{cases}\dfrac{a+b}{e^x+\cos x}\\ \dfrac{a+b}{x+1}\\ e^x+\sin x\end{cases}$	$x<2.8$ $2.8\le x<6$ $x\ge 6$	$a=2.6$ $b=-0.39$	$x\in[0;7]$ $\Delta x=0.5$
12	$Y=\begin{cases}a\cdot\lg x+\sqrt[3]{\lvert x\rvert}\\ 2a\cdot\cos x+3x^2\end{cases}$	$x>1$ $x\le 1$	$a=0.9$	$x\in[0.8;2]$ $\Delta x=0.1$
13	$W=\begin{cases}\dfrac{a}{i}+bi^2+c\\ ai+bi^3\end{cases}$	$i<4$ $4\le i\le 6$ $i>6$	$a=2.1$ $b=1.8$ $c=-20.5$	$i\in[0;12]$ $\Delta i=1$
14	$Z=\begin{cases}a\cdot\sin\left(\dfrac{i^2+1}{n}\right)\\ \cos\left(i+\dfrac{1}{n}\right)\end{cases}$	$\sin\left(\dfrac{i^2+1}{n}\right)>0$ $\sin\left(\dfrac{i^2+1}{n}\right)<0$	$a=0.3$ $n=10$	$i\in[1;10]$ $\Delta i=1$
15	$Q=\begin{cases}\sqrt{at^2+b\cdot\sin t+1}\\ at+b\\ \sqrt{at^2+b\cdot\cos t+1}\end{cases}$	$t<0.1$ $t=0.1$ $t>0.1$	$a=2.5$ $b=0.4$	$t\in[-1;1]$ $\Delta t=0.2$

Таблица 4.2

Вариант	Сумма членов ряда	Точность
1	2	3
1	$S=-\frac{(2x)^2}{2}+\frac{(2x)^4}{24}+.....+(-1)^n\cdot\frac{(2x)^{2n}}{2n!}+.....$	10^{-4}
2	$S=x-\frac{x^3}{3}+.....+(-1)^{n+1}\cdot\frac{x^{2n-1}}{2n-1}+.....$	10^{-5}
3	$S=\frac{x^3}{5}-\frac{x^5}{17}+....+(-1)^{n+1}\cdot\frac{x^{2n+1}}{4n^2+1}+...$	10^{-4}
4	$S=1+\cos\left(\frac{\pi}{4}\right)\cdot\left(\frac{x}{1!}\right)+.....+\cos\left(\frac{n\pi}{4}\right)\cdot\left(\frac{x^n}{n!}\right)+.....$	10^{-6}
5	$ch\,x=1+\frac{x^2}{2!}+\frac{x^4}{4!}+....+\frac{x^{2n}}{2n!}+....$	10^{-5}
6	$\pi=4\cdot\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+...+\frac{(-1)^{n+1}}{2n-1}+....\right)$	10^{-8}
7	$arctg\,x=\frac{1}{x}-\frac{1}{3x^3}+\frac{1}{5x^5}-...+\frac{(-1)^n}{((2n+1)\cdot x^{2n+1})}+...$	10^{-6}
8	$\cos\left(\frac{\pi}{6}\right)=-\frac{\left(\frac{\pi}{6}\right)^2}{2!}+\frac{\left(\frac{\pi}{6}\right)^4}{4!}-....+(-1)^n\cdot\frac{\left(\frac{\pi}{6}\right)^{2n}}{(2n)!}+....$	10^{-4}
9	$sh\,x=x+\frac{x^3}{3!}+\frac{x^5}{5!}+....+\frac{x^{2n+1}}{(2n+1)!}+....$	10^{-5}
10	$\sin\left(\frac{\pi}{3}\right)=\frac{\pi}{3}-\frac{\left(\frac{\pi}{3}\right)^3}{3!}+\frac{\left(\frac{\pi}{3}\right)^5}{5!}+....+(-1)^n\cdot\frac{\left(\frac{\pi}{3}\right)^{2n+1}}{(2n+1)!}+...$	10^{-4}
11	$S=1+\frac{x^2}{2!}-\frac{3x^4}{4!}+....+(-1)^n\cdot\frac{(2n-1)\cdot x^{2n}}{(2n)!}+...$	10^{-5}
12	$S=\frac{2}{3}\cdot\sin 2x-\frac{3}{8}\cdot\sin 3x+....+(-1)^n\cdot n\cdot\sin\frac{nx}{n^2-1}+...$	10^{-6}
13	$S=1+\frac{\cos x}{1!}+\frac{\cos 2x}{2!}+....+\frac{\cos nx}{n!}+....$	10^{-5}
14	$S=\frac{x^3}{3}-\frac{x^5}{15}+....+(-1)^{n+1}\cdot\frac{x^{2n+1}}{4n^{2-1}}+...$	10^{-4}
15	$S=x\cdot\cos\left(\frac{\pi}{3}\right)+x^2\cdot\cos\frac{\left(\frac{2\pi}{3}\right)}{2}+....+x^n\cdot\cos\frac{\left(\frac{n\pi}{3}\right)}{n}+..$	10^{-5}

Лабораторная работа № 5

ОДНОМЕРНЫЕ И МНОГОМЕРНЫЕ МАССИВЫ В СИСТЕМЕ MATLAB

Цель работы: получение практических навыков работы с массивами в системе Matlab.

Задание. В системе автоматизированных расчетов Matlab обработать массив в соответствии с вариантом задания (см. табл. 5.1).

Методические указания

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

1. Введены явно с помощью списка элементов.
2. Сгенерированы встроенными операторами или функциями.
3. Созданы в m-файлах.
4. Загружены из внешнего файла данных.

Явное задание матриц

Например, любой из приведенных далее операторов

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

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

создает матрицу 3 × 3 и присваивает ее значение переменной

```
A =
    1 2 3
    4 5 6
    7 8 9.
```

Элементы внутри строки матрицы могут отделяться друг от друга не только пробелами, но и запятыми. При вводе чисел в экспоненциальной форме (например, 2,34e – 94) нельзя использовать пробелы. Ввод больших матриц лучше выполнять с помощью m-файлов, в которых легко находить и исправлять ошибки. В MATLAB встроен ряд функций, позволяющих создавать функции специального вида, например, rand, magic, zeros и ones позволяют легко сгенерировать матрицы.

Функция rand(n) создает матрицу размером $n \times n$, каждый элемент которой – случайное число с равномерным законом распределения в диапазоне [0, 1].

Функция rand(m,n) создает матрицу размера $m \times n$, каждый элемент которой – случайное число с равномерным законом распределения в диапазоне [0, 1].

Функция magic(n) создает матрицу размером $n \times n$, которая является магическим квадратом (суммы элементов по строкам и столбцам равны).

Функция zeros(m,n) создает нулевую матрицу размера $m \times n$.

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

Матрицы могут быть сгенерированы также с помощью цикла for.

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

Подматрицыи использование двоеточия

Для записи алгоритмов сложной обработки данных в компактной форме в системе MATLAB используются векторы и подматрицы. Использование нотации с двоеточием (которая используется и для генерации векторов и подматриц) и векторов вместо индексов является ключом к эффективной манипуляции этими объектами. Эффективное использование этих возможностей позволяет минимизировать число явных циклов, использование которых существенно замедляет работу MATLAB, и делает написанную программу простой и легко читаемой. (Правда, овладение данной технологией требует от пользователя пакета определенных усилий.) Например, выражение 1 : 5 фактически является вектором-строкой [1 2 3 4 5]. Отметим, элементы вектора могут быть не только целыми, но и действительными.

Например, команда

>> x=0.2:0.2:1.2

создает вектор

x = [0.2, 0.4, 0.6, 0.8, 1.0, 1.2],

а команда

>> 5:-1:1

создает вектор

[5 4 3 2 1].

Для создания и вывода на экран таблицы синусов необходимо выполнить следующую последовательность команд:

>> x = [0.0:0.1:2.0]'; % транспонирование вектора

>> y = sin(x); % вычисление вектора, содержащего значения sin

>> [x y] % вывод таблицы значений.

Отметим, что так как sin является скалярной функцией (то есть функцией, действующей поэлементно), поэтому результатом ее применения к вектору *x* будет вектор *y*.

Для доступа к подматрицам может быть использовано двоеточие. Например, A(1:2,3) является вектором-столбцом, состоящим из двух первых элементов третьего столбца матрицы A:

>> A(2:3,3)

ans =

6

9.

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

>> A(:,3)

ans =

3

6

9

>> A(3,☺;

ans =

7 8 9.

В качестве индекса подматрицы может использоваться произвольный целый вектор:

>> A(1,[2 3])

ans =

2 3.

Описанные способы индексирования могут использоваться с обеих сторон знака присваивания. Например, после выполнения команды

>> A(:,[2 4 5]) = B(:,1:3)

2, 4 и 5-й столбцы матрицы A будут заменены на первые три столбца матрицы B.

Функции построения матриц

В MATLAB имеются следующие стандартные функции для построения матриц:

eye(m,n) – создание единичной матрицы размером $m \times n$;

zeros(m,n) – создание нулевой матрицы размером $m \times n$;

ones(m,n) – создание матрицы размером $m \times n$, каждый элемент которой равен единице;

diag(x) – создание матрицы, у которой на главной диагонали стоят элементы вектора x;

diag(A) – создание матрицы, у которой на главной диагонали стоят диагональные элементы матрицы A;

triu(A) – создание верхней треугольной матрицы, элементы которой, расположенные на главной диагонали и выше главной диагонали, равны соответствующим элементам матрицы A;

tril(A) – создание нижней треугольной матрицы, элементы которой, расположенные на главной диагонали и ниже главной диагонали, равны соответствующим элементам матрицы A.

Операции, выражения и переменные

Пакет MATLAB является интерпретирующим языком непосредственных вычислений. Это означает, что выражения, которые вы вводите, интерпретируются и вычисляются. Операторы пакета MATLAB обычно имеют форму:

```
>> имя_переменной = выражение
```

или просто

```
>> выражение
```

Выражение, как правило, формируется из операторов, функций и имен переменных. После выполнения выражения генерируется матрица, которая выводится на экран и присваивается соответствующей переменной для последующего использования. Если имя переменной в левой части и знак = отсутствуют, автоматически генерируется переменная ans (answer – ответ), которой присваивается результат вычислений.

Обычно оператор завершается клавишей <Enter>. Однако при необходимости оператор может быть продолжен на следующей строке. Для этого его необходимо завершить тремя или более точками, после которых следует <Enter>. С другой стороны, в одной строке может быть несколько операторов, разделенных запятой или точкой с запятой.

Если последним символом в строке является точка с запятой, то вывод значений результата не производится, но присвоение значения выражения переменной выполняется. Это помогает подавить вывод ненужных промежуточных результатов. Важно помнить, что MATLAB различает строчные и прописные буквы в именах команд, функций и переменных.

Для получения списка всех переменных, расположенных в рабочем пространстве, используется команда who. Переменная может быть удалена из рабочего пространства командой clear <имя_переменной>. Команда clear без аргументов очищает все непостоянные переменные, расположенные в рабочем пространстве. Примером постоянной переменной является переменная eps (epsilon), значение которой по умолчанию равно 10^{-6}. Данная переменная используется для оценки точности вычислений итеративных процессов.

Вывод на дисплей или вычисления могут быть прерваны на большинстве компьютеров, не покидая MATLAB, с помощью комбинации клавиш <Ctrl>+<C>

(<Ctrl>+<Break> на PC).

Матричные операции

В MATLAB доступны следующие матричные операции:

< + > – сложение;

< - > – вычитание;

< * > – умножение;

< ^ > – возведение в степень;

< ′ > – транспонирование;

< \ > – левое деление;

< / > – правое деление.

Данные матричные операции применимы, конечно, и к скалярам – матрицам 1 × 1. Если размерность матриц не соответствует используемой операции, то система генерирует сообщение об ошибке, за исключением случаев, когда одним из операндов является скаляр, потому что в этом случае операция выполняется между скаляром и каждым элементом матрицы второго операнда.

Отметим, что операция возведения в степень применима только для квадратных матриц.

Если A является невырожденной квадратной матрицей, а b – вектор-столбец или вектор-строка соответственно, тогда вектор x = A \ b является решением уравнения A x = b, а x = b / A является решением уравнения x A = b. Если A – квадратная матрица, то при левом делении для факторизации используется метод исключения Гаусса. Если матрица не квадратная, то для ее факторизации используется метод ортогонализации Хаусхольдера с ведущим столбцом, а приведенная матрица используется для решения переопределенной системы уравнений в смысле наименьших квадратов. Правое деление определяется в терминах левого деления по формуле b / A = (A' \ b')'.

Операции с массивами

Матричные операции сложения и вычитания действуют поэлементно, а остальные приведенные выше операции – нет, они являются матричными операциями. Следует отметить, что приведенные выше операции *, ., \, / могут стать поэлементными, если перед ними поставить точку. Например, команды

```
>> [1,2,3,4].*[1,2,3,4]
```

или

```
>> [1,2,3,4].^2
```

дадут один и тот же результат

ans=

[1,4,9,16].

Функции MATLAB

В MATLAB существует большое количество функций, созданных разработчиками системы, большинство из которых предоставлено в виде m-файлов, содержащих исходные тексты. Можно подойти к классификации данных функций различными способами, например по областям их использования (тригонометрические, спецфункции, функции линейной алгебры и т. д.). Далее мы используем подход, основанный на виде аргумента функции: скаляр, вектор, матрица.

<u>Скалярные функции</u>

Скалярные функции MATLAB действуют только на скаляры. Если аргументом данных функций является матрица, то они действуют поэлементно. К таким функциям,

например, относятся

sin asin exp abs round

cos acos log (натуральный логарифм) sqrt floor

tan atan rem (остаток от деления двух чисел) sign ceil.

<u>Векторные функции</u>

Аргументами векторных функций являются векторы (строки или столбцы). Если в качестве аргумента функции указана матрица размером $m \times n$ (m >= 2), то данная функция действуют постолбцово, то есть результатом действия является векторстолбец, каждый элемент которой является результатом действия этой функции на соответствующий столбец. Построчное действие такой функции (если необходимо) может быть достигнуто использованием операции транспонирования.

Названия некоторых из этих функций приведены ниже:

max sum median any

min prod mean all

sort std.

Например, максимальный элемент прямоугольной матрицы находится с помощью команды max(max(A)), а не с помощью max(A), так как результат, возвращенный функцией max(A), – вектор, каждая компонета которого есть максимальный элемент соответствующего столбца.

Матричные функции

Наибольшую мощь системе MATLAB дают матричные функции, наиболее употребительные из которых приведены ниже:

Eig – собственные значения и собственные векторы;

Chol – факторизация Холецкого;

Svd –сингулярная декомпозиция;

Inv – обратная матрица;

Lu – LU-факторизация;

Qr – QR-факторизация;

Hess – вычисление формы Хессенберга;

Schur – декомпозицияШура;

Rref – приведение к треугольной форме методом Гаусса;

Expm – матричная экспонента;

Sqrtm – матричный корень квадратный;

Poly –характеристический полином;

Det –определитель;

Size – размерность;

Norm – норма вектора или матрицы;

cond– число обусловленности;

rank – ранг матрицы;

Функции MATLAB могут возвращать одновременно несколько переменных. Например, функция

y = eig(A),

или просто

eig(A)

генерирует вектор-столбец, содержащий собственные значения матрицы A, в то время как оператор

[U,D] = eig(A)

генерирует матрицу U, чьи столбцы являются собственными векторами A, а диагональная матрица D содержит на главной диагонали собственные значения этой матрицы.

Варианты заданий

Таблица 5.1

Вар. зад.	Массив	Действия	Условия и ограничения
1	$x(100)$	Вычислить сумму элементов массива *X*, значения которых больше 0,5.	$0 < x_i < 1$
2	$A(80)$	Вычислить среднее арифметическое значение элементов массива *A*.	$0 < a_i < 100$
3	$X(15)$	Переписать положительные элементы массива *X* в массив *Y* и подсчитать их количество. Вывести массивы на экран.	$-1 < x_i < 1$
4	$C(10)$	Определить минимальный элемент массива *C* и его номер.	$x_i < 0$
5	$X(N)$	Вычислить среднее гармоническое значение элементов массива *X*, кратных двум.	$x_i > 0, N < 30$
6	$Y(20)$	Вычислить среднее геометрическое элементов массива	$y_i > 0$
7	$Z(30)$	Расположить в массиве *R* сначала положительные, а затем отрицательные элементы.	-
8	$N(50)$	Определить сумму элементов массива *N* кратных трем.	$Int(n_i / 3) + 3 = n_i$
9	$D(80)$	Найти максимальный элемент массива *D*.	-
10	$A(N)$	Найти среднее геометрическое элементов массива *A*.	$a_i > 0, N < 50$
11	$X(N)$	Переписать в массив *Y* положительные элементы массива *X*. Массивы вывести на экран.	$x_i > 0, N < 40$
12	$X(N)$	Переписать подряд в массив *Y* отрицательные элементы массива *X*. Массивы вывести на экран.	$N < 40$
13	$B(K)$	Определить максимальный элемент массива *B* и его порядковый номер.	$x_i < 0, K < 40$
14	$C(K)$	Определить среднее квадратичное элементов массива *C*.	$-1 < x_i, K < 20$

Справочный материал

Даны n чисел: $x_1, x_2, \ldots\ldots, x_n$.

Среднее арифметическое этих чисел

$$A = \frac{x_1 + x_2 + \ldots + x_n}{n}.$$

Среднее геометрическое

$$G = \sqrt[n]{x_1 x_2 \ldots x_n}, (x_i > 0).$$

Среднее гармоническое

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n}}, (x_i > 0).$$

Среднее квадратичное

$$K = \sqrt{\frac{x_1^2 + x_2^2 + \ldots + x_n^2}{n}}.$$

Лабораторная работа № 6

ПРОГРАММИРОВАНИЕ И ОБРАБОТКА ДАННЫХ В СИСТЕМЕ MATLAB

Цель работы: изучение принципов программирования в MatLab; освоение приемов обработки данных.

Задание

1. Аппроксимировать полиномом таблично заданную функцию $y = f(x)$. Степень полинома выбрать с учетом критерия:

$$\text{а) } \sum_{i=1}^{n}(y_i - y_{ia})^2 \rightarrow \min \qquad \text{б) } \sum_{i=1}^{n}|(y_i - y_{ia})| \rightarrow \min$$

где n - количество точек заданной функции, y_{ia} - значения функции аппроксимирующего полинома в точках x_i.

2. Найти корни полученного полинома.

3. Вывести график заданной функции и аппроксимирующей полиномиальной функции.

Варианты заданий даны в таблице 6.1.

Методические указания

Для решения сложных задач режима прямых вычислений недостаточно. Возникает необходимость сохранять используемые последовательности вычислений и модифицировать их. Иными словами, нужно программировать решение задачи.

Программа в MatLab представляет собой запись в m-файле (файл с расширением ".m") всей той последовательности команд, которые потребовались бы для решения задачи в режиме прямых вычислений. Запуск на выполнение m-файла осуществляется введением в строке его имени без расширения и нажатием клавиши "Enter".

Ввод данных с клавиатуры в программе организуется при помощи оператора input, а вывод на дисплей – disp. Например,

```
r=input ('Введите r');
disp (r)
```

Кроме того, управление выводом информации на экран может осуществляться посредством символа ";".

В системе MatLab оператор ветвления имеет следующий сокращенный вид:

```
if Условие
   Операторы
end
```

а в общем случае записывается следующим образом:

```
if Условие
   Операторы1
else
   Операторы2
end
```

В первом варианте "Операторы" выполняются только в том случае, если условие истинно, и не выполняются если ложно. Во втором варианте если условие истинно, то выполняется группа операторов "Операторы1", в противном случае, выполняется группа операторов "Операторы2".

Для организации циклов в программах используются операторы for ...end (со счетчиком) и while ...end (с предварительной проверкой условия). Первый из них записывается следующим образом:

```
for Счетчик=Нач_знач:Шаг:Кон_знач
     Тело цикла (операторы)
end
```

где Счетчик – имя переменной параметра цикла (не обязательно целочисленный тип), Нач_знач и Кон_знач – начальное и конечное значение переменной цикла, Шаг задает величину приращения переменной Счетчик при ее изменении от Нач_знач до Кон_знач.

Например, задать случайным образом вектор X из 10 элементов и подсчитать в нем число положительных элементов.

```
X=rand(1, 10)–0.5
```

```
n=0;
for i=1:1:10
  if X(i)>0
    n=n+1;
  end
end
disp (n)
```

Конструкция цикла while ...end имеет вид:

```
while Условие
  Тело цикла (операторы)
end
```

В данном случае операторы цикла будут выполняться, пока будет истинно условие "Условие".

Центральным моментом программирования в среде MatLab является задание новых функций. Для этого используется конструкция

```
function Имя_переменной=Имя_функции (Список_аргументов)
% Комментарий, выводимый при использовании команды help
     Операторы % Комментарии
      Имя_переменной=Выражение
```

Все переменные, используемые внутри блока функции, являются локальными, т.е. область их действия распространяется только на блок функции.

Для того, чтобы создать свою функцию необходимо войти в любой текстовый редактор (например, в FAR – клавиши "Shift"+"F4"), ввести весь текст с описанием функции и сохранить его в файле с именем "Имя_функции.m". Например, создав запись

```
function y=func(x);
  y=sqrt(2*x+1);
```

и сохранив ее в файле "func.m", мы получаем доступ к этой функции ($y=\sqrt{2x+1}$), как, скажем, к стандартной функции sin(x), т.е. следующая команда даст результат:

```
>> z=func(4)
   z =
      3
```

Пример. Вычисление факториала числа. В файле с именем "fact.m" создаем запись:

```
function f=fact(x)
% fact(x) - функция вычисления факториала x!
  f=1;
  if x>0
    for i=1:x
      f=f*i;
    end
  end
```

Первая строка с комментарием (%) будет выводиться на экран в результате выполнения команды

```
>> help Имя_функции
```

то есть

```
>> help fact
   fact(x) - функция вычисления факториала x!
```

Пример определения функции трех переменных $y=\frac{x_1^2}{2}+\frac{x_2^3}{3}+\frac{x_3^3}{4}$.

```
function y=func3(x)
% Файл func3.m
  y=x(1)^2/2+x(2)^3/3+x(3)^4/4;
```

Тогда вызов функции может быть осуществлен следующими способами:

```
>> x=[1 2 3]; func3(x)
```

```
ans =
      23.4167
```

или

```
>> func3([1 2 3])
ans =
      23.4167
```

Для изменения формата представления действительных чисел используется команда

```
>> format Тип_формата
```

В MatLab доступны следующие виды форматы:

short – короткое представление в фиксированном формате (4 знака после десятичной запятой)

short e – короткое представление в экспоненциальном формате (4 знака после запятой и 3 знака порядка)

long – длинное представление в фиксированном формате (14 знаков после запятой)

long e – длинное представление в экспоненциальном формате (14 знаков после запятой и 3 знака порядка)

Например,

```
>> format short; x=4/3
x =
    1.3333
>> format short e; x=4/3
x =
    1.3333e+000
```

Операции с многочленами

Многочлен, определяемый следующим выражением:

$$P(x) = a_n x^n + ... + a_2 x^2 + a_1 x + a_0,$$

задается в виде вектора, хранящего коэффициенты от a_n до a_0. Для нахождения корней полинома в MatLab используется функция roots (P), возвращающая вектор, элементы которого являются корнями многочлена, заданного вектором P. Например, пусть требуется найти корни уравнения

$$x^5 + 8x^4 + 31x^3 + 80x^2 + 94x + 20 = 0$$

```
>> P=[1 8 31 80 94 20];
>> roots (P)
    ans =
          -1.0000 + 3.0000i
          -1.0000 - 3.0000i
          -3.7321
          -2.0000
          -0.2679
```

Обратная функция poly (R), наоборот, восстанавливает по корням коэффициенты полинома:

```
>> R = [1 -1];
>> P = P=poly(R)
    P =
        1    0    -1
```

т.е. для корней $x = \pm 1$, мы получили полином $P = x^2 - 1$.

Для проведения полиномиальной аппроксимации используется функция polyfit(x,y,n), которая реализует вычисление коэффициентов полинома n-ой степени, аппроксимирующего зависимость y(x). Для вычисления значений аппроксимирующей полиномиальной функции в точках сетки используется функция polyval(P,x), где P – вектор коэффициентов аппроксимирующего полинома, x – точки сетки. Например (рис. 6.1),

```
>> x = [1 2 3 4 5]; y = [3 4 7 10 11];
>> P = polyfit(x,y,2)
```

```
P =

    0.0000    2.2000    0.4000

>> yp = polyval(P,x)

yp =

    2.6000    4.8000    7.0000    9.2000   11.4000

>> plot (x, y, 'r-', x, yp, 'm:')
```

В результате мы получили уравнение аппроксимирующей полиномиальной зависимости $y_p(x) = a_2 x^2 + a_1 x + a_0$, где $a_2 = 0$; $a_1 = 2{,}2$; $a_0 = 0{,}4$.

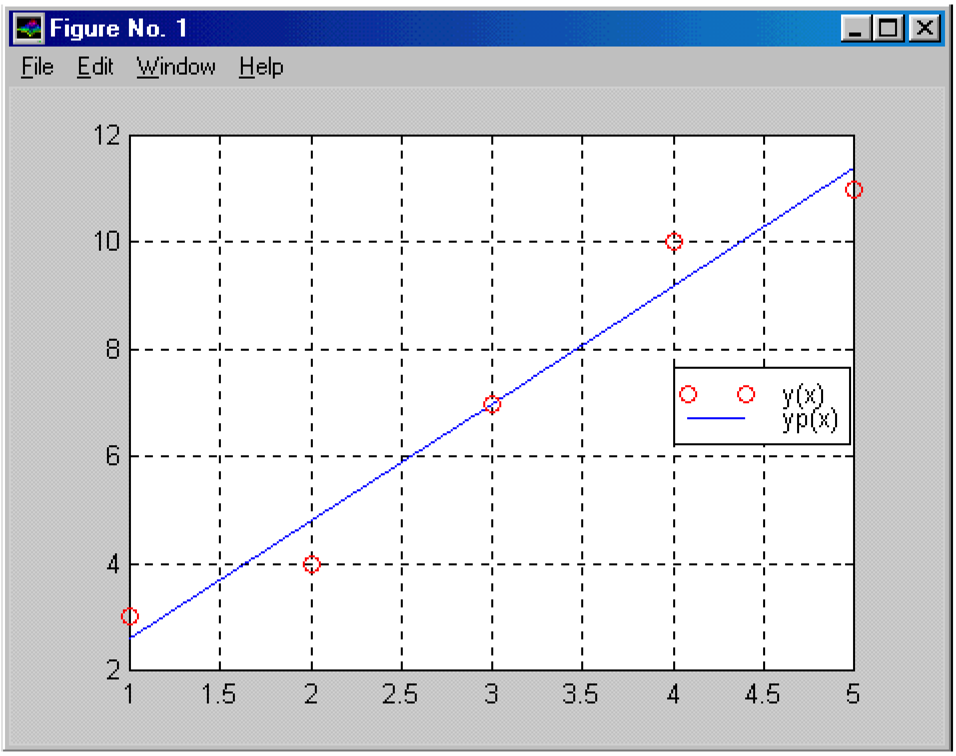

Рис. 6.1. Графики таблично заданной функции $y(x)$ и аппроксимирующей полиномиальной зависимости $y_p(x)$

Варианты заданий

Таблица 6.1

№	Критерий	Данные y=f(x)								
1	а	x	1	2	3	4	5	6	7		
		y	2,45	2,56	2,35	2,48	2,51	2,66	2,98		
2	б	x	1	2	3	4	5	6	7	8	9
		y	10,56	10,98	10,54	10,78	10,32	10,58	10,91	10,25	10,56
3	а	x	11	22	33	44	55	66	77	88	
		y	121,2	132,6	129,6	130,5	127,6	126,4	135,1	128,2	
4	б	x	1,2	1,3	1,4	1,5	1,6	1,7			
		y	18,0	17,5	16,8	17,4	18,6	18,1			
5	а	x	1	2	3	4	5	6	7		
		y	121	131	141	125	135	138	129		
6	б	x	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
		y	4.52	4.35	4.28	4.68	4.55	4.70	4.29	4.56	4.85
7	а	x	5	5.5	6	6.5	7	7.5	8	8.5	
		y	21.3	22.1	21.9	22	22.6	23.1	22.4	21.8	
8	б	x	17	18	19	20	21	22	23		
		y	0.56	0.63	0.58	0.49	0.63	0.55	0.59		
9	а	x	14	15	16	17	18	19			
		y	121	124	126	128	130	124			
10	б	x	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	
		y	14.23	14.56	14.28	14.98	14.11	14.78	15.11	14.32	
11	а	x	120	130	140	150	160	170	180		
		y	6.33	6.52	6.42	6.81	6.12	6.35	6.45		
12	б	x	1	2	3	4	5	6	7	8	9
		y	55.23	54.96	53.89	56.36	51.69	54.32	55.31	52.89	54.56
13	а	x	0.5	1	1.5	2	2.5	3	3.5	4	
		y	5	7.3	6.4	8.1	4.9	6.3	5.6	7.2	
14	б	x	71	75	79	83	87	91			
		y	2.36	2.45	2.8	2.12	2.65	2.74			
15	а	x	12	13	14	15	16	17	18	19	
		y	281	254	266	239	255	263	278	245	
16	б	x	65	70	75	80	85	90	95	100	
		y	16.36	15.98	16.21	16.54	16.83	17.23	15.99	16.87	
17	а	x	7	8	9	10	11	12	13	14	15
		y	7.41	7.23	7.56	7.12	7.9	8.1	6.98	7.56	7.49
18	б	x	11	11.5	12	12.5	13	13.5			
		y	33.3	12.4	0.5	44.5	12	27.6			

Лабораторная работа № 7

ОБРАБОТКА ДАННЫХ В СИСТЕМЕ MATLAB

Цель работы: освоение приемов обработки данных; ознакомление с численным интегрированием в системе MatLab.

Задание

1. Аппроксимировать определенную заданным количеством точек на указанном интервале функцию периодической функцией (с использованием быстрого преобразования Фурье) на более мелкой сетке x.

Провести кубическую интерполяцию и интерполяцию сплайнами функции внутри указанного интервала на более мелкой сетке x.

Вывести зависимости заданной, аппроксимирующей и интерполяционных функций на одном графике. Варианты заданий даны в таблице 1.

2. Вычислить значение определенного интеграла $\int_a^b f(x)dx$ методами трапеций и квадратур. Определить относительные погрешности численного интегрирования, сравнив полученные результаты с точным значением интеграла $F(b)-F(a)$, F - первообразная функции $f(x)$. Варианты заданий даны в таблице 7.1.

Методические указания

Аппроксимация и интерполяция

Для аппроксимации периодических функций на основе быстрого преобразования Фурье используется оператор interpft (y, n), возвращающий одномерный массив чисел, который является периодической функцией, определенной в n точках и аппроксимирующей одномерный массив y.

Интерполяция функции одной переменной осуществляется с помощью оператора interp1 (x, y, xi, 'способ'), где xi – сетка, на которой производится интерполяция зависимости y(x), заданной на более крупной сетке. При этом значения xi должны лежать внутри области определения функции y(x). Строковый параметр 'способ' определяет способ интерполяции. Возможны следующие способы:

'linear' – линейная интерполяция;

'spline' – интерполяция сплайнами;

'cubic' – интерполяция кубическими полиномами.

Пример. Выполнить аппроксимацию функции y=sin(x), которая задана 10 точками на интервале [0; 3π] и провести интерполяцию (кубическую и сплайновую) данной функции на интервале [π/2; π] (рис. 5.2).

```
clf
x=0:pi/3:3*pi; y=sin(x);
xa=0:pi/10:3*pi;
ya=interpft(y, length(xa));
plot(x, y, 'bo', xa, ya, 'r')
xi=pi/2:pi/10:pi;
yi=interp1(x, y, xi, 'cubic');
hold on
plot (xi, yi, 'k+')
yi=interp1(x, y, xi, 'spline');
plot (xi, yi, 'g')
legend('исход','аппрокс','кубич','сплайн')
grid
hold off
```

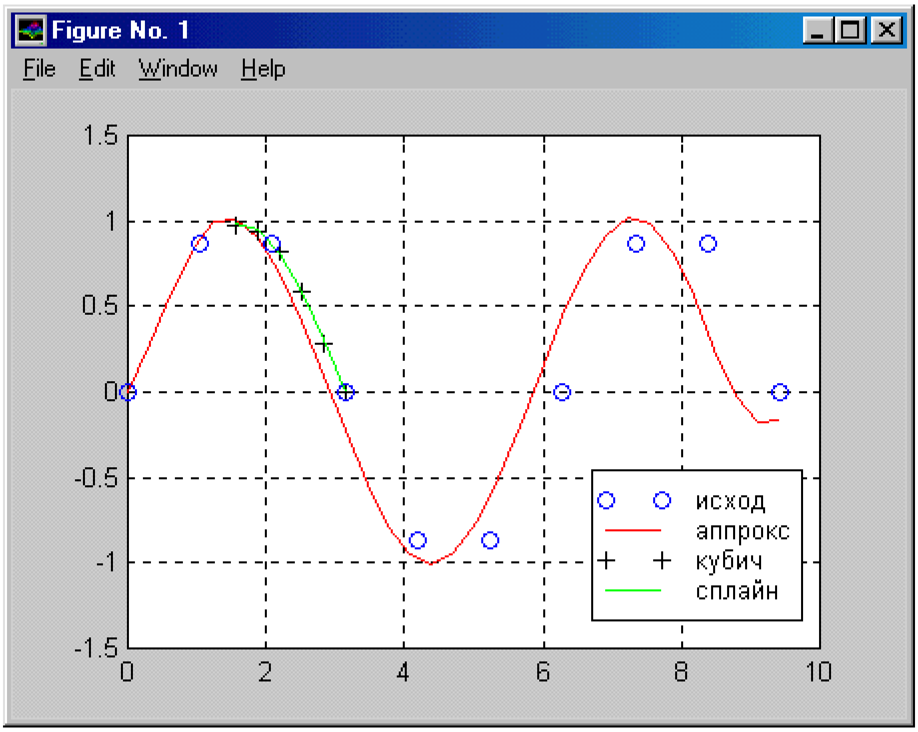

Рис. 7.1. Пример выполнения аппроксимации и интерполяции функции

Численное интегрирование в MatLab

Для выполнения операций численного интегрирования в системе MatLab используются две функции: реализующая метод трапеций – trapz и реализующая методом квадратур – quad.

Синтаксис функции trapz имеет вид:

trapz (x, y)

где x и y – векторы одинаковой размерности: x – переменная интегрирования, y – значения интегрируемой функции в точках x. Например, требуется определить значение интеграла $\int_0^{\pi} \sin(x)dx$. Его точное значение равно двум.

```
>> x=0:pi/100:pi; y=sin(x);
>> I=trapz(x,y)
  I =
      1.9998
```

Функция вычисления интеграла по методу квадратур имеет следующий вид:

quad ('имя_функции', ниж_пред, верх_пред, точность)

где "имя_функции" – имя функции (m-файла), в котором определена интегрируемая функция, "ниж_пред" и "верх_пред" – нижний и верхний пределы интегрирования, точность – задаваемая относительная погрешность вычисления интеграла (по умолчанию точность равна 0.001).

Например, для предыдущего примера

```
>> I=quad ('sin', 0, pi, 0.01)
   I =
        2.0003
```

Если осуществить вызов функции quad следующим образом:

```
>> [I, n]=quad('sin',0,pi,0.0001)
   I =
        2.0000
   n =
        33
```

то в первой переменной (I) будет возвращено значение интеграла, а во второй (n) – число выполненных итераций, обеспечивших расчет интеграла с заданной точностью.

Варианты заданий

Таблица 7.1

№	**Функция**	**Заданный интервал**	**Кол-во точек**	**Интервал интерполяции**
1	**2**	**3**	**4**	**5**
1	$\frac{1}{\sin x \cdot \cos x}$	π/20, 2π/5	10	π/4, π/3
2	$tgx \cdot ctg2x$	π/20, 2π/5	8	π/4, π/3
3	$\frac{\sin x}{\sin 2x}+1$	π/15, 3π/7	12	π/5, π/4
4	$4\sin^3 x + 2\sin 3x$	0, 2π	6	π/2, π
5	$tgx + \cos x$	π/20, 2π/5	8	π/5, π/4
6	$\sin 2x + \sin 4x + \sin$	0, 2π	10	π,3π/2

Продолжение таблицы 7.1

1	2	3	4	5
7	$\frac{1}{\sin x}+\frac{2}{\cos x}$	π/20, 2π/5	15	π/5, π/4
8	$\cos x \cos 4x$	-π, π	8	0, π/2
9	$8\cos^3 x+3\sin 3x$	-π, 2π	12	-π/2, 0
10	$\sin x \cos 3x$	π, 4π	7	2π, 5π/2
11	$\frac{tg 3x}{\sin x}-1$	π/10, 9π/10	8	π/2, 2π/3
12	$\cos x+\cos 3x-\sin$	π, 3π	9	2π, 5π/2
13	$tg 2x-\cos x$	π/15, 3π/7	15	π/5, π/4
14	$2\sin^2 2x+1$	0, 2π	10	0, 2π/3
15	$\cos x \sin 4x$	0, 2π	14	0, 2π/5
16	$\sin 3x-\cos x$	0, π	6	0, 2π/7
17	$\frac{\sin 2x}{\mathrm{tg} x}$	π/15,3π/7	7	π/15,4π/15
18	$\cos x \cdot \sin x+\cos^2 3$	-2π,2π	12	-π/2,π

Таблица 7.2

№	Подынтегральная функция $f(x)$	Первообразная $F(x)$	Интервал Интегрирования $[a,b]$
1	2	3	4
1	$\frac{1}{\sqrt{9+x^2}}$	$\ln\left\|x+\sqrt{x^2+9}\right\|$	[0, 2]
2	$x^x(1+\ln x)$	x^x	[1, 3]
3	$\sqrt{x^2+16}$	$x\sqrt{x^2+16}+8\ln\left\|x+\sqrt{x^2+16}\right\|$	[2, 5]
4	$\frac{1+2x}{x^2(1+x^2)}$	$-\frac{1}{x}+2\ln x-\ln(1+x^2)-\mathrm{arctg}\, x$	[6, 8]
5	$e^x \cos^2 x$	$e^x(0{,}4+0{,}2\cos^2 x+0{,}2\sin 2x)$	[0, π]
6	$\sqrt{1-x^2}$	$0.5\left(\arcsin x+x\sqrt{1-x^2}\right)$	[-0.5, 0.5]
7	$x^3/(3+x)$	$x^3/3-3x^2/2+9x-27\ln(x+3)$	[1, 2]
8	$1/\cos^2 3x$	$\mathrm{tg} 3x/3$	[-π/8, 0]
9	$\frac{\ln^2 x}{x}$	$\frac{\ln^3 x}{3}$	[1, 4]
10	$5\sin x+\cos x$	$-5\cos x+\sin x$	[0, π/2]
11	$\sqrt{2x-1}$	$\sqrt{(2x-1)^3}/3$	[2, 5]

Продолжение таблицы 7.2

1	2	3	4
12	$\dfrac{1}{\sin(1/x)\cdot x^2}$	$-\ln\left(\dfrac{1}{\sin(1/x)}-ctg(1/x)\right)$	[1, 2.5]
13	$\dfrac{1}{(x+2)(x+4)}$	$-\dfrac{1}{2}\ln\left\|\dfrac{x+4}{x+2}\right\|$	[1, 3]
14	$\cos(\ln\|x\|)$	$x[\sin(\ln\|x\|)+\cos(\ln\|x\|)]/2$	[3, 6]
15	$\dfrac{x^2+x+2}{x^4-5x+4}$	$\dfrac{1}{3}\ln\left\|\dfrac{(x+1)^2(x-2)}{(x-1)(x+2)^2}\right\|$	[3, 7]
16	$x\,\mathrm{arctg}x$	$\dfrac{\mathrm{arctg}(x^2+1)-x}{2}$	[0, 3]
17	xe^{3x}	$\dfrac{e^{3x}(3x-1)}{9}$	[0.7, 3.1]
18	$\dfrac{x}{x^4+3x^2+2}$	$\dfrac{1}{2}\ln\dfrac{-x^2-1}{x^2+2}$	[2, 4]

Литература

1. Симонович, С.В. Информатика. Базовый курс: учебное пособие для втузов. / С.В. Симонович. – СПб.: Питер, 2009. – 640 с.
2. Информатика: учебник для вузов. / А.Н.Гуда [и др.] – М.: Дашков и К, 2008. - 400 с.
3. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебное пособие для вузов. / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2007. - 958 с.
4. Левин, А.Ш. Самоучитель работы на компьютере / А.Ш. Левин. – СПб.: Питер, 2009. – 672 с.
5. Левин, А.Ш. Word и Excel / А.Ш. Левин. – СПб.: Питер, 2009. – 224 с.
6. Грызлов, В.И. Турбо Паскаль 7.0 [Электронный ресурс] / В.И. Грызлов, Т.П. Грызлова – М.: ДМК Пресс, 2006. – 400 с. – Режим доступа: http://e.lanbook.com/books/pdf.php?book_id=1217&p_id=25&bookid=1007
7. Зеленяк, О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения [Электронный ресурс] / О.П. Зеленяк – М.: ДМК Пресс, 2009. – 320 с. – Режим доступа: http://e.lanbook.com/books/pdf.php?book_id=1249&p_id=25&bookid=1218
8. Кудрявцев, Е.М. Mathcad 11: Полное руководство по русской версии[Электронный ресурс]/Е.М. Кудрявцев.- М.:ДМК-Пресс, 2009. – 592 с. - Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1172.
9. Дьяконов, В.П. MATLAB 7.*/R2006/R2007: Самоучитель [Электронный ресурс]/ В.П. Дьяконов.- М.:ДМК-Пресс, 2009. – 768 с. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1178.