МИНИСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

(ФГБОУ ВО «ВятГУ»)

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Теория принятия решений

Методические указания
к лабораторным и самостоятельным работам

4 курс

Киров, 2020

УДК 004.891(07)

Р783

Ростовцев В.С. Теория принятия решений: методические указания к самостоятельным и лабораторным работам.- Киров: Изд-во ВятГУ, 2020.-49 с.

Пособие может быть полезно студентам других специальностей при знакомстве и изучении MatLab.

Составитель к.т.н., доцент кафедры ЭВМ В.С. Ростовцев

1. Лабораторная работа №2. Основы работы в MatLab

1.1. Цель лабораторной работы

Цель лабораторной работы: приобретение навыков основы работы с программой MATLAB.

Название MATLAB является сокращением от Matrix Laboratory, и первоначально разрабатывался как средство для матричных вычислений. При помощи MATLAB и его расширений (Toolbox) выполняется матричный анализ, обработка сигналов и изображений, задачи математической физики, оптимизационные задачи, финансовые задачи, обработка и визуализация данных. моделирование нейронных сетей, нечёткой логики и многое другое. Более 40 специализированных Toolbox могут быть выборочно установлены вместе с MATLAB.
При запуске MATLAB на экране открывается рабочая среда, приведённая на рис.1.1.

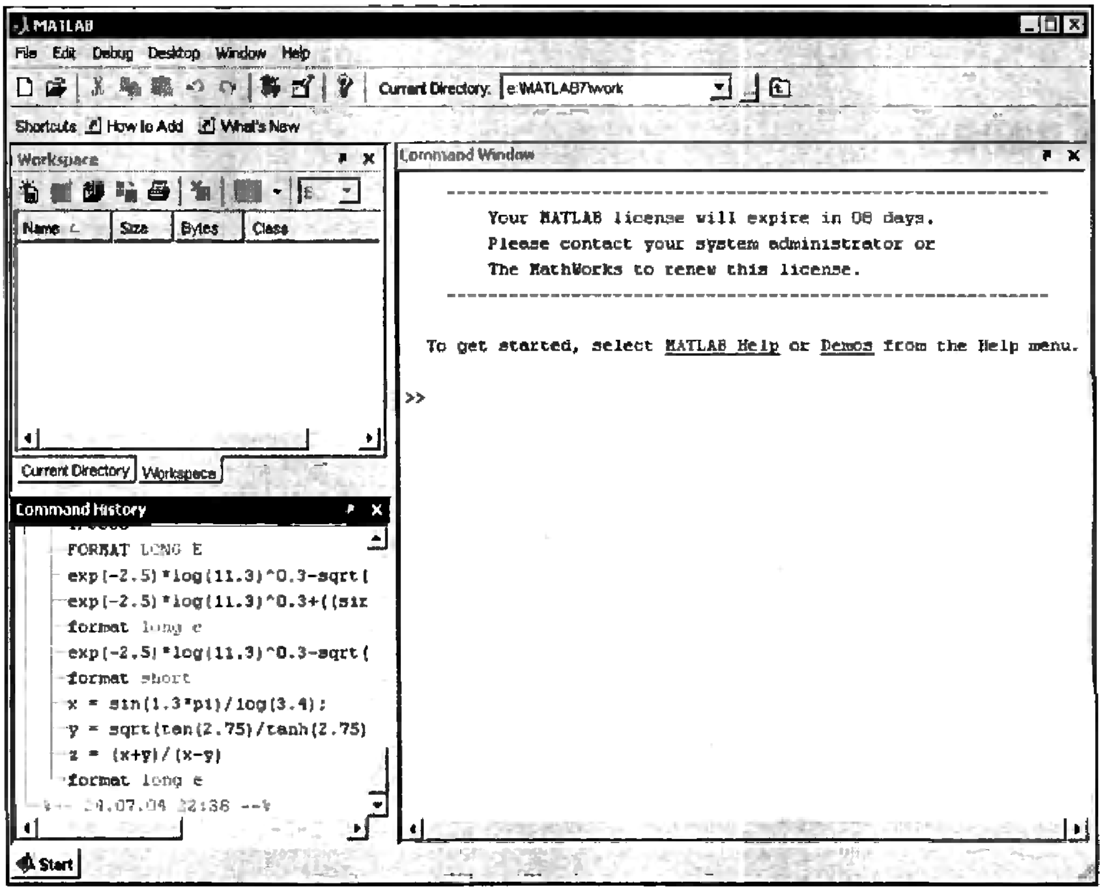

Рис. 1.1. Рабочая среда MATLAB

Для ввода команды в окне Command Window появляется приглашение в виде символов «>>».

Перечень задач по лаб. работе

Формат представления данных

>> 1/3000

ans=3.3333e-004

Запись 3.3333e-004 обозначает 3.3333 $*10^{-4}$ или 0.00033333

<u>Пробел между цифрой и символом «е» не допускается.</u>

Формат можно задать из командной строки. Например, для задания формата с плавающей точкой необходимо ввести команду только строчными буквами

>> format long e

>> 1.25/3.11

ans=4.019292604501608e-001

1.2. Использование элементарных функций

Вычислить выражение

$e^{-2.5}*(\ln 11.3)^{0.3} - \sqrt{}(\sin 2.45\pi+\cos 3.48\,\pi)/tg3.3$

Ввести в командной строке

>> exp2.5* ln11.3^$^{0.3}$ – sqrt((sin(2.45*pi)+cos(3.48*pi))/tan(3.3))

Ответ выводится в командное окно

\>> ans=

-3.2105

Для ввода ранее введённого выражения необходимо нажать клавишу стрелку вверх(↑).

1.3. Тригонометрические, гиперболические и обратные к ним функции

sin, cos, tan, cot – синус, косинус, тангенс, котангенс

sec, csc, - секанс, косеканс   sec(x)=1/cos(x); csc(x)=1/sin(x)

asin, acos, atan, acot – арксинус, арккосинус, арктангенс, арккотангенс

Гиперболические функции синус, косинус, тангенс, котангенс

sinh, cosh, tanh, coth – синус, косинус, тангенс, котангенс.

1.4. Экспоненциальные функции и логарифмы

exp-экспоненциальная функция

log-натуральный логарифм

log10-десятичный логарифм

log2-логарифм по основанию 2

pow2-возведение в степень числа 2

sqrt-квадратный корень

1.5. Округление и остаток

fix- округление до ближайшего целого в направлении к нулю

\>> fix(1.8)

ans=

1

round

round

\>> round(4.5)

ans=

```
>> round(4.3)
ans=
4
```

mod- –остаток от целочисленного деления со знаком второго аргумента

```
>>mod(5,2)
ans=
1
>>mod(5,-2)
ans=
-1
```

rem-остаток от целочисленного деления (со знаком первого аргумента)

```
>> rem(5,2)
ans=
1
>> rem(5,-2)
ans=
1
```

1.6. Использование переменных

Присваивание переменной

```
>> z=1.45
Z=
```

1.45

>> x=sin(1.3*pi)/log(3.4);

>> y=sqrt(tan(2.75)/tanh(2.75));

>>z=(x+y)/(x-y)

z=

0.0243-0.9997i

Самый простой способ сохранения переменных- использовать в меню *File* пункт *SaveWorkspace As*.

При этом появляется диалоговое окно *Save to MAT-File*, в котором надо указать каталог и имя файла. По умолчанию сохранение файла производится в подкаталоге work каталога MatLab.

Сохранить историю команд перед завершением можно следующим образом

Save history file on quit.

1.7. Работа с массивами

Массив-это упорядоченная, пронумерованная совокупность однородных данных. Массивы различаются по размерности: одномерные, двумерные, многомерные. Размером массива называют число элементов вдоль каждого из измерений.

1.7.1 Вектор-столбцы и вектор-строки

Рассмотрим вычисление суммы векторов

$$a=\begin{bmatrix}1.3\\5.4\\6.9\end{bmatrix} \quad b=\begin{bmatrix}7.1\\3.5\\8.2\end{bmatrix}$$

Сначала вводятся массивы a и b.

Введите массив в командной строке в квадратных скобках, разделяя элементы точкой с запятой.

>> a=[1.3; 5.4;6.9]

a=

1.3000

5.4000

6.9000

Аналогично вводится массив b.

>> b=[7.1; 3.5;8.2]

Для вычисления суммы двух векторов используется знак +.

>>c=a+b

c=

8.4000

8.9000

15.1000

Просмотрите появившуюся в окне Workspace информацию. Векторы *a* и *b* хранятся в двумерных массивах.

Пусть требуется вычислить sin всех элементов вектора c. Если аргумент является массивом, то и результат функции будет массивом того же размера.

>> d=sin c

d=

0.8546

0.5010

0.5712

Ввод вектор-строки отличается от ввода вектора-столбца. Элементы разделяются пробелами или запятой.

s1=[3 4 9 2]

s1=

3 4 9 2

>> s2=[5 3 3 2]

s2=

5 3 3 2

>> s3=s1+s2

s3=

8 7 12 4

>>s4=log(s3)

s4=

2.0794 1.9459 2.4849 1.3863

Для определения длины вектор-столбца или вектор-строки используется команда length

>>L= length(s1)

L=

4

Умолчанию все элементы массива хранятся с двойной точностью и занимают 8 байтов.

Для уменьшения объёма занимаемой массивами памяти можно применять другие способы хранения элементов массива: single для вещественных чисел, требующих 4 байта, и int8, int16, int32 – для целых чисел, требующих 1,2 и 4 байта соответственно.

Изменение способа хранения выполняется следующим образом

>>g4=single(s4);

>>>g3=int32(s3);

>>g2=int16(s2);

>>g1=int8(s1);

Попытка выполнить арифметические операции над целыми числами разных типов приводит к ошибке.

Из нескольких вектор-столбцов можно составить один

```
>> v1=[1; 2];
>> v2=[3; 4; 5];
>> v=[v1;v2]
v=
1
2
3
4
5
```

Для сцепления векторов-строк применяются квадратные скобки

```
>> v1=[pi pi/2];
>> v1=[pi/3 pi/4 pi/5];
>> v=[v1+v2]
v=
   1.1416  1.5708  1.0472  0.7854  0.6283
```

1.7.2 Обращение к элементам вектора

Пусть имеется вектор-строка v

```
>>v=[1.3  3.6  7.4  8.2  0.9]
```

Для обращения к четвёртому элементу

>> h=v(4)

h=

8.2000

Можно изменить элемент массива

>>v(2)=777

v=

1.3000 777.0000 7.4000 8.2000 0.9000

Из элементов массива можно сформировать новые массивы

>>u=[v(3); v(2); v(1)]

u=

7.4000

777.0000

1.3000

Предположим, что в вектор-строке w необходимо заменить нулями элементы со второго по шестой. Для этого служит индексация при помощи двоеточия.

>> w=[0.1 2.9 3.3 5.1 2.6 7.1 9.8];

>> w=(2:6)=0;

>> w

w=

0.1000 0 0 0 0 0 0 9.8000

Присваивание w(2:6)=0 эквивалентно последовательности команд w(2)=0;

w(3)=0; w(4)=0; w(5)=0; w(6)=0.

Для выделения части из массива можно снова использовать двоеточие

>> w=[0.1 2.9 3.3 5.1 2.6 7.1 9.8];

w1=w(3:5)

w1=

3.3000 5.1000 2.6000

Вектор-строка содержит элементы кроме четвёртого

w2=w[(1:3) w(5:7)]

w2=

0.1000 2.9000 3.3000 2.6000 7.1000 9.8000

1.7.3 Применение функций обработки данных к векторам

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

>>z=[3; 2; 1; 4; 6 ;5]

>>p= prod(z)

p=

720

Функция *sum* предназначена для суммирования элементов вектора.

>>sum(z)/length(z)

ans=

3.5000

Среднее арифметическое находит функция *mean*.

>> mean(z)

ans=

3.5000

Для нахождения минимума или максимума служат встроенные функции min, max.

>>M=max(z)

M=

6

>>m=min(z)

m=

1

При обращении функции *min* с двумя векторами в качестве входных аргументов получается вектор, каждый элемент которого есть минимум из двух элементов исходных векторов

>> p=[3 12 8]

>> s=[4 10 7]

>> min(p,s)

ans=

3 10 7

Часто требуется знать не только минимальное и ли максимальное значение, но и его индекс(порядковый номер).

>>[m, k]=min(z)

m=

1

k=

3

В число основных функций при работе с векторами входит сортировка

по возрастанию его элементов.

>>r=[9.4 -2.3 -5.2 7.1 0.8 1.3]

>>R=sort(r)

R=

-5.2000 -2.3000 0.8000 1.3000 7.1000 9.4000

Сортировка по убыванию

>>R1=-sort(-r)

R1=

9.4000 7.1000 1.3000   0.8000 -2.3000 -5.2000

>>R2=sort(abs(r))

R2=

0.8000 1.3000 2.3000 5.2000 7.1000 9.4000

1.7.4 Поэлементные операции с векторами

Умножение векторов одинаковой длины

>>v1=[2 -3 4 1]

>>v2=[7 5 -6 9]

>>u=v1.*v2

u=

14 -15 -24 9

Между точкой и звёздочкой нет пробела (.*).

При помощи функции точка и крышечка (.^) *производится поэлементное возведение в степень.*

>>p=v1.^2

p=

4 9 16 1

Деление соответствующих элементов векторов одинаковой длины выполняется с помощью функции ./

>>d=v1./v2

d=

0.2857 -0.6000 -0.6667 0.1111

Обратное поэлементное деление (деление элементов второго вектора на соответствующие элементы первого вектора) выполняется с помощью функции .\

>>dinv=v1.\v2

dinv=

3.5000 -1.6667 -1.5000 9.0000

Операции сложения с числом

```
>> v=[4  6  8 10];
>> s=v+1.2
s=
5.2000  7.2000  9.2000 11.2000
>>r=1.2 – v
 r=
-2.8000  -4.8000  -6.8000  -8.8000
```

Умножение вектора на число

```
>>v=[4  6  8  10]
>>p=v*2
p=
8  12  16  20
```

Деление вектора на число

```
>>p=v/2
p=
2  3  4  5
```

Деление числа на вектор-строку недопустимо. Если требуется разделить число на каждый элемент вектора и записать новый вектор. То следует использовать операцию ./.

```
>>w=[4  2  6];
>>d=12./w
```

d=

3 6 2

Все описанные операции применимы как к вектор-строкам, так и к вектор-столбцам.

1.8. Построение таблицы значений функции

Отображение функции в виде таблицы удобно, если число значений функции невелико.

Пусть требуется вывести в командное окно значения функции

$y(x)=[\sin^2 x/(1+\cos x)]+e^{-x}*\ln x$.

Задача решается в два этапа.

1. Создайте вектор-строку x, содержащую координаты заданных точек.
2. Вычислите функцию y(x) для каждого элемента вектора и запишите полученные значения в вектор-строку y.

```
>>x=[0.2  0.3  0.5  0.8  1.3  1.7  2.5]

x=

0.2000  0.3000  0.5000 0.8000 1.3000 1.7000 2.5000

>>y=sin(x).^2./(1+cos(x))+exp(-x).*log(x)

y=

-1.2978  -0.8473  -0.2980  0.2030  0.8040  1.2258  1.8764
```

Таблице можно придать удобный вид, расположив значения x и y.

```
 >> x

x=

0.2000  0.3000  0.5000 0.8000 1.3000 1.7000 2.5000

>>y

y=
```

-1.2978 -0.8473 -0.2980 0.2030 0.8040 1.2258 1.8764

В MatLab предусмотрено простое создание векторов, отличающихся на одинаковую величину.

x = начальное значение: шаг: конечное значение

```
>>x=1.2:0.2:2

x=

1.2000  1.4000 1.6000 1.8000 2.0000
```

В случае отрицательного шага начальное значение должно быть больше или равно конечному значению.

```
>>x=1.9:-0.2:1

x=

1.9000  1.7000  1.5000 1.3000 1.1000
```

При заполнении вектора-столбца в начале заполняется вектор-строка, а затем производится её транспонирование (обозначается символом апострофа).

```
>>x=(0: 0.1: 0.5)`

x=

0

0.1000

0.2000

0.3000

0.4000

0.5000
```

1.9. Умножение векторов

Вектор можно умножить на другой вектор скалярно, векторно и образовывать матрицу.

Скалярное произведение векторов a, b длины N, определяется формулой

$$a*b=\sum_{k=1}^{N} a_k*b_k$$

Рассмотрим вычисление скалярного произведения векторов

$$a=\begin{bmatrix}1.2\\-3.2\\0.7\end{bmatrix} \quad b=\begin{bmatrix}4.1\\6.5\\-2.9\end{bmatrix}$$

```
>>a=[1.2; -3.2; 0.7];
>>b=[4.1; 6.5; -2.9];
>>sum(a.*b)
>>s=
-17.9100
```

Скалярное произведение можно вычислить, применив функцию *dot*

```
>>dot(a,b);
```

Векторное произведение $a\times b$ определено только для векторов из трёхмерного пространства, т.е. состоящих из трёх элементов. Результатом является вектор из трёхмерного пространства. Для вычисления векторного произведения служит функция *cross*.

```
>>a=[1.2; -3.2; 0.7];
>>b=[4.1; 6.5; -2.9];
>>c=cross(a,b)
c=
4.7300
6.3500
20.9200
```

Смешанное произведение векторов *a,b,c* определяется по формуле *abc=a*(b×c)*. Модуль смешанного произведения векторов равен объёму параллепипеда, построенного на этих векторах.

```
>>a=[3.5; 0;0];

>>b=[0.5; 2.1; 0];

>>c=[-0.2; -1.9; 2.8];

>>v=abs(dot(a, cross(b,c)))

v=

20.5800
```

Внешнее произведение

Внешним произведением векторов *a,b* называется матрица.

```
>>a=[1;2;3];

>>b=[5;6;7];

>>c=a*b’

c=

5 6 7

10 12 14

15 18 21
```

1.10. Двумерные массивы, матрицы

1.10.1. Ввод матриц

Ввод матрицы размером два на три. Матрицу можно рассматривать как вектор-столбец из двух элементов, каждый из которых является вектор-строкой длиной три. Поэтому строки при наборе разделяются точкой с запятой.

A=[3 1 -1; 2 4 3];

A=

3 1 -1

2 4 3

Другие способы ввода квадратной матрицы.

>>B= [4 3 -1

2 7 0

-5 1 2]

По окончанию набора нажмите Enter.

Ещё один способ ввода матриц состоит в том, что матрицу можно трактовать как вектор-строку, каждый элемент которой является вектором-столбцом.

Рассмотрим матрицу C размером два на три

$$\begin{matrix} 3 & -1 & 7 \\ 4 & 2 & 0 \end{matrix}$$

>>C=[[3;4] [-1;2] [7;0]]

C=

$$\begin{matrix} 3 & -1 & 7 \\ 4 & 2 & 0 \end{matrix}$$

Для проверки откройте окно Workspace или наберите в командной строке whos.

1.10.2. Обращение к элементам матриц

Доступ к элементам матриц производится с помощью двух индексов- номера строки и номера столбца, заключённых в круглые скобки.

>>C(2,3)

ans=

0

Элементы матриц могут входить в состав выражений:

>>C(1,1)+C(2,2)+C(2,3)

ans=

5

1.10.3 Сложение, вычитание, умножение, транспонирование и возведение в степень

При использовании матричных операций следует помнить, что для сложения или вычитания матрицы должны быть одного размера, а при перемножении число столбцов первой матрицы обязательно равняется числу строк второй матрицы. Сложение и вычитание осуществляется с помощью знаков плюс и минус.

>>S=A+B

S=

6 0 6

6 6 3

>>S=A-B

S=

0 -2 8

2 -2 -3

Для умножения матриц предназначена «звёздочка».

>>P=A*B

P=

-25 9 11

20 26 -4

Умножение матрицы на число.

>>P=A*3

P=

9 3 -3

6 12 9

Транспонирование матрицы, как и вектора, производится с помощью апострофа (‘). Для вещественных чисел можно использовать точку с апострофом (.‘).

$$B = \begin{matrix} 4 & 3 & -1 \\ 2 & 7 & 0 \\ -5 & 1 & 2 \end{matrix}$$

>>B’

$$\begin{matrix} 4 & 2 & -5 \\ 3 & 7 & 1 \\ -1 & 0 & 2 \end{matrix}$$

Возведение квадратной матрицы в целую степень производится с помощью команды ^.

>>B2=B1^2

B2=

$$\begin{matrix} 27 & 32 & -6 \\ 22 & 55 & -2 \\ -28 & -6 & 9 \end{matrix}$$

Проверьте полученный результат, умножив матрицу саму на себя.

Задание. Вычислите выражение $(A+C)B^3(A-C)^T$

Учтите приоритет операций, сначала выполняется транспонирование, затем возведение в степень, потом умножение, а сложение и вычитание производится в последнюю очередь.

>>(A+C)*B^3*(A-C)’

1.10.4 Перемножение матрицы и вектора

Поскольку вектор-столбец или вектор-строка в MatLab являются матрицами, у которых один из размеров равен единице, то все вышеописанные операции применимы и для умножения матрицы на вектор, или вектор-строки на матрицу.

Например, вычисление выражения

$$[1\ 3\ {-2}]\begin{vmatrix} 2 & 0 & 1 \\ -4 & 8 & -1 \\ 0 & 9 & 2 \end{vmatrix}\begin{bmatrix} -8 \\ 3 \\ 4 \end{bmatrix}$$

Можно осуществить следующим образом:

```
>>a=[1  3  -2];
>>B=[2  0  1; -4  8  -1; 0  9  2];
>>c=[-8; 3; 4];
>>a*B*c
ans=
74
```

1. **Варианты задания. Вычисление произведения матриц и векторов**

```
>> a=[5   4   7];
>> B=[7   4   2;  2   4   4;  5   4   3];
>> c=[2;  6;  8];
```

Вариант	Вектор-строка A	Матрица B	Вектор-столбец C	P	Вычисляемое выражение
1	A=[5 4 7]	B=[7 4 4; 2 4 4; 5 4 3]	C=[2;6;8]	13	(A*B*C)/13
2	A=[2 4 9]	B=[8 4 3; 5 4 4; 5 9 8]	C=[3;7;8]	33	(A*B*C)/33

3	A=[3 4 9]	B=[7 4 5; 7 4 4; 8 4 3]	C=[7;6;9]	27	(A*B*C)/27
4	A=[6 4 7]	B=[7 4 2; 2 4 4; 5 4 3]	C=[2;6;8]	11	(A*B*C)/11
5	A=[7 4 9]	B=[8 2 5; 5 4 4; 5 9 8]	C=[3;7;8]	25	(A*B*C)/25
6	A=[8 4 9]	B=[7 3 6; 7 4 4; 8 4 3]	C=[4;6;9]	44	(A*B*C)/44
7	A=[5 4 1]	B=[1 4 3; 2 4 4; 5 4 3]	C=[2;6;8]	13	(A*B*C)/13
8	A=[2 4 2]	B=[2 4 3; 5 4 4; 5 9 8]	C=[3;7;8]	23	(A*B*C)/13
9	A=[3 4 3]	B=[3 4 3; 7 4 4; 8 4 3]	C=[1;6;9]	27	(A*B*C)/27
10	A=[6 4 4]	B=[4 1 3; 2 4 4; 5 4 3]	C=[4;6;8]	11	(A*B*C)/11
11	A=[7 4 5]	B=[5 2 4; 5 4 4; 5 9 8]	C=[5;7;8]	25	(A*B*C)/25
12	A=[8 4 6]	B=[6 3 4; 7 4 4; 8 4 3]	C=[6;6;9]	44	(A*B*C)/44
13	A=[10 4 7]	B=[1 4 4; 2 4 4; 5 4 3]	C=[2;6;8]	14	(A*B*C)/14
14	A=[12 4 9]	B=[2 4 4; 5 4 4; 5 9 8]	C=[3;7;8]	15	(A*B*C)/15
15	A=[11 4 9]	B=[3 4 5; 7 4 4; 8 4 3]	C=[7;6;9]	16	(A*B*C)/16

16	A=[13 4 7]	B=[4 1 5; 2 4 4; 5 4 3]	C=[2;6;8]	17	(A*B*C)/17
17	A=[14 4 9]	B=[5 2 5; 5 4 4; 5 9 8]	C=[3;7;8]	18	(A*B*C)/18
18	A=[15 4 9]	B=[6 3 5; 7 4 4; 8 4 3]	C=[4;6;9]	19	(A*B*C)/19
19	A=[25 4 1]	B=[7 4 5; 2 4 4; 5 4 3]	C=[2;6;8]	20	(A*B*C)/20
20	A=[16 4 2]	B=[8 4 3; 5 4 4; 5 9 8]	C=[3;7;8]	21	(A*B*C)/21
21	A=[17 4 3]	B=[9 4 4; 7 4 4; 8 4 3]	C=[1;6;9]	22	(A*B*C)/22
22	A=[18 4 4]	B=[10 1 4; 2 4 4; 5 4 3]	C=[4;6;8]	23	(A*B*C)/23
23	A=[19 4 5]	B=[11 2 5; 5 4 4; 5 9 8]	C=[5;7;8]	24	(A*B*C)/24
24	A=[20 4 6]	B=[12 3 5; 7 4 4; 8 4 3]	C=[6;6;9]	45	(A*B*C)/45
25	A=[13 4 7]	B=[4 1 5; 2 4 4; 5 4 3]	C=[2;6;8]	26	(A*B*C)/26
26	A=[14 4 9]	B=[5 2 5; 5 4 4; 5 9 8]	C=[3;7;8]	27	(A*B*C)/27
27	A=[15 4 9]	B=[6 3 5; 7 4 4; 8 4 3]	C=[4;6;9]	28	(A*B*C)/28
28	A=[25 4 1]	B=[7 4 3; 2 4 4; 5 4 3]	C=[2;6;8]	29	(A*B*C)/29

29	A=[16 4 2]	B=[8 4 3; 5 4 4; 5 9 8]	C=[3;7;8]	30	(A*B*C)/30
30	A=[17 4 3]	B=[9 4 3; 7 4 4; 8 4 3]	C=[1;6;9]	31	(A*B*C)/31
31	A=[18 4 4]	B=[10 1 3; 2 4 4; 5 4 3]	C=[4;6;8]	32	(A*B*C)/32
32	A=[19 4 5]	B=[11 2 4; 5 4 4; 5 9 8]	C=[5;7;8]	33	(A*B*C)/33

2. Решение систем линейных алгебраических уравнений методом исключения Гаусса

К решению систем линейных уравнений сводятся многочисленные практические задачи, например, различные краевые задачи для обыкновенных и в частных производных дифференциальных уравнений. Можно с полным основанием утверждать, что данная проблема является одной из самых распространенных и важных задач вычислительной математики.
Пусть задана система *n* линейных алгебраических уравнений с *n* неизвестными:

2.2. Решение систем линейных алгебраических уравнений методом исключения Гаусса

К решению систем линейных уравнений сводятся многочисленные практические задачи, например различные краевые задачи для обыкновенных и в частных производных дифференциальных уравнений. Можно с полным основанием утверждать, что данная проблема является одной из самых распространенных и важных задач вычислительной математики.

Пусть задана система *n* линейных алгебраических уравнений с *n* неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ ---------------- \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases} . \qquad (2.1)$$

Система уравнений (2.1) в матричной форме представляется следующим образом:

$$\mathbf{AX} = \mathbf{B}, \qquad (2.2)$$

где **A** – квадратная матрица коэффициентов, размером $n \times n$ строк и столбцов;

X – вектор-столбец неизвестных;

B – вектор-столбец правых частей.

Систему уравнений (2.2) можно решить различными методами. Один из наиболее простых и эффективных методов является метод исключения Гаусса и его модификации. Алгоритм метода основан на приведении матрицы **A** к треугольному виду (прямой ход) и последовательном вычислении неизвестных (обратный ход). Эти процедуры можно выполнять над невыраженными матрицами, в противном случае метод Гаусса неприменим.

Недостатком метода является накапливание погрешностей в процессе округления, поэтому метод Гаусса без выбора главных элементов используется обычно для решения сравнительно небольших ($n \leq 100$) систем уравнений с плотно заполненной матрицей и не близким к нулю определителем. Если матрица **A** сильно разрежена, а ее определитель не близок к нулю, то метод Гаусса пригоден для решения больших систем уравнений. В MATLAB имеется обширный арсенал методов решения систем уравнений (2.2) методом исключения Гаусса. Для этого применяются следующие операторы

| / | – правое деление;

| \ | – левое деление;

| **^-1** | – возведение в степень –1;

| inv(A) | – обращение матрицы **A**.

Выражения

```
X = B/A
X = B* A^ – 1
X = B* inv(A)
X = A\B
```

дают решения ряда систем линейных уравнений **AX = B**, где **A** – матрица размером $m \times n$, **B** – матрица размером $n \times k$. Более сложные случаи решения систем уравнений (2.2) с плохо обусловленной матрицей **A** освещены в работе [1].

Пример 3.

Решить систему 4-х линейных уравнений:

$$\begin{cases}1.1161x_1 + 0.1397x_2 + 0.1254x_3 + 0.1490x_4 = 1.5471;\\ 0.1582x_1 + 0.1768x_2 + 1.1675x_3 + 0.1871x_4 = 1.6471;\\ 0.1968x_1 + 1.2168x_2 + 0.2071x_3 + 0.2271x_4 = 1.7471;\\ 0.2368x_1 + 0.2568x_2 + 0.2471x_3 + 1.2671x_4 = 1.8471.\end{cases}$$

Протокол программы (в М-файле)

```
a   =  [1.1161   0.1397   0.1254   0.1490 ;
         0.1582   0.1768   1.1675   0.1871 ;
         0.1968   1.2168   0.2071   0.2271 ;
         0.2368   0.2568   0.2471   1.2671] ;
b   =  [1.5471 ; 1.6471 ; 1.7471 ; 1.8471] ;

X4 = a \ b
```

Эта программа выдает решение заданной системы с помощью четвертого оператора в виде матрицы – столбца

```
X4   =
        1.0406
        0.9351
        0.9870
        0.8813
```

Внимание. В М-файле матрица ***a*** *набирается по строкам,* а *элементы* матрицы правых частей ***b*** *отделяются символом* **;** , т. е. *тоже набираются по строкам*. Решение другими операторами системы уравнений (2.2) требует набора матрицы ***a*** *по столбцам*, а *элементы* правых частей ***b*** *отделяются только пробелом*!

```
a   =  [1.1161   0.1582   0.1968   0.2368 ;
         0.1397   0.1768   1.2168   0.2568 ;
         0.1254   1.1675   0.2071   0.2471 ;
         0.1490   0.1871   0.2271   1.2671] ;
b   =  [1.5471   1.6471   1.7471   1.8471] ;

X1 = b/a
X2 = b * a ^ – 1
X3 = b * inv(a)
```

Результаты решения

X1 =

1.0406 0.9351 0.9870 0.8813

X2 =

1.0406 0.9351 0.9870 0.8813

X3 =

1.0406 0.9351 0.9870 0.8813

Варианты заданий. Решить систему линейных алгебраических уравнений с помощью 4-х операторов. Данные взять из таблицы 2.2.

Таблица 2.2

1	$a_{ij} = \begin{bmatrix} 2 & -4 & 3 & 1 \\ -1 & 5 & -7 & -3 \\ 10 & -2 & 4 & 4 \\ -1 & 1 & -1 & -1 \end{bmatrix}; \quad b_i = \begin{bmatrix} 7 \\ -24 \\ 34 \\ -6 \end{bmatrix}$	**2**	$a_{ij} = \begin{bmatrix} 5 & -3 & 4 & -2 \\ 10 & 3 & -4 & 2 \\ 7 & -5 & 8 & -10 \\ 4 & 5 & -8 & 10 \end{bmatrix}; \quad b_i = \begin{bmatrix} 4 \\ 11 \\ 0 \\ 11 \end{bmatrix}$
3	$a_{ij} = \begin{bmatrix} 2 & 2 & -1 & 1 \\ 4 & 3 & -1 & 2 \\ 8 & 5 & -3 & 4 \\ 3 & 3 & -2 & 2 \end{bmatrix}; \quad b_i = \begin{bmatrix} 4 \\ 6 \\ 12 \\ 6 \end{bmatrix}$	**4**	$a_{ij} = \begin{bmatrix} 1 & 1 & -6 & 4 \\ 3 & -1 & -6 & -4 \\ 2 & 3 & 9 & 2 \\ 3 & 2 & 3 & 8 \end{bmatrix}; \quad b_i = \begin{bmatrix} 6 \\ 2 \\ 6 \\ -7 \end{bmatrix}$
5	$a_{ij} = \begin{bmatrix} 1 & 5 & 3 & 4 \\ 7 & 14 & 20 & 27 \\ 5 & 10 & 16 & 19 \\ 3 & 5 & 6 & 13 \end{bmatrix}; \quad b_i = \begin{bmatrix} 0 \\ 0 \\ -2 \\ 5 \end{bmatrix}$	**6**	$a_{ij} = \begin{bmatrix} 2 & 4 & 3 & 4 \\ 4 & -2 & 5 & 6 \\ 6 & -3 & 7 & 8 \\ 8 & -4 & 9 & 10 \end{bmatrix}; \quad b_i = \begin{bmatrix} 5 \\ 7 \\ 9 \\ 11 \end{bmatrix}$
7	$a_{ij} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 1 & 2 \\ -1 & 2 & 4 & 1 \\ 2 & -1 & 1 & -1 \end{bmatrix}; \quad b_i = \begin{bmatrix} 1 \\ 1 \\ 18 \\ -12 \end{bmatrix}$	**8**	$a_{ij} = \begin{bmatrix} 5 & -3 & 2 & 4 \\ 4 & -2 & 3 & 7 \\ 8 & -6 & -1 & -5 \\ 7 & -3 & 7 & 17 \end{bmatrix}; \quad b_i = \begin{bmatrix} 7 \\ 10 \\ 4 \\ 17 \end{bmatrix}$
9	$a_{ij} = \begin{bmatrix} 1 & 2 & 15 & 11 \\ -2 & 4 & 3 & 61 \\ -3 & -8 & 11 & 12 \\ 15 & 7 & 8 & -4 \end{bmatrix}; \quad b_i = \begin{bmatrix} 2 \\ 4 \\ 5 \\ 9 \end{bmatrix}$	**10**	$a_{ij} = \begin{bmatrix} 5 & 3 & -4 & -15 \\ 6 & 1 & 10 & 2 \\ 1 & 5 & 3 & 8 \\ -4 & 5 & -1 & 9 \end{bmatrix}; \quad b_i = \begin{bmatrix} 6 \\ 9 \\ 12 \\ -9 \end{bmatrix}$
11	$a_{ij} = \begin{bmatrix} 7 & -8 & 9 & 11 \\ -5 & 6 & 5 & 3 \\ -4 & -7 & 3 & 6 \\ 2 & 1 & 5 & 9 \end{bmatrix}; \quad b_i = \begin{bmatrix} 3 \\ -4 \\ -6 \\ 11 \end{bmatrix}$	**12**	$a_{ij} = \begin{bmatrix} 1 & 2 & 5 & -11 \\ 2 & -6 & -3 & 8 \\ 3 & 4 & 5 & 1 \\ 7 & 6 & 2 & 20 \end{bmatrix}; \quad b_i = \begin{bmatrix} 7 \\ 11 \\ 15 \\ -9 \end{bmatrix}$

Продолжение таблицы 2.2

13	$a_{ij}=\begin{bmatrix}5 & 3 & 3 & -4\\ -6 & -4 & 2 & 17\\ 12 & 26 & 1 & 5\\ 5 & 8 & 9 & 7\end{bmatrix};\quad b_i=\begin{bmatrix}7\\ -24\\ 34\\ -6\end{bmatrix}$	**14**	$a_{ij}=\begin{bmatrix}4 & 7 & 6 & -19\\ 5 & 9 & 12 & 3\\ -4 & -3 & 1 & -8\\ 2 & 5 & 9 & 10\end{bmatrix};\quad b_i=\begin{bmatrix}10\\ 12\\ -14\\ 22\end{bmatrix}$
15	$a_{ij}=\begin{bmatrix}12 & 8 & 24 & -5\\ 16 & 25 & 9 & 9\\ 2 & 7 & 3 & 4\\ 21 & -9 & -7 & 6\end{bmatrix};\quad b_i=\begin{bmatrix}5\\ -17\\ 22\\ 3\end{bmatrix}$	**16**	$a_{ij}=\begin{bmatrix}-3 & -7 & 16 & 8\\ -4 & 6 & 12 & 13\\ 2 & 3 & 7 & 5\\ 18 & -14 & 1 & 2\end{bmatrix};\quad b_i=\begin{bmatrix}6\\ -12\\ 31\\ 41\end{bmatrix}$
17	$a_{ij}=\begin{bmatrix}2 & 5 & 9 & 16\\ 11 & 16 & 3 & 1\\ 25 & 14 & -38 & 7\\ -8 & 9 & -16 & -42\end{bmatrix};\quad b_i=\begin{bmatrix}1\\ 1\\ -3\\ 6\end{bmatrix}$	**18**	$a_{ij}=\begin{bmatrix}2 & -1 & 3 & 2\\ 3 & 3 & 3 & 2\\ 3 & -1 & -1 & -2\\ 3 & -1 & 3 & -1\end{bmatrix};\quad b_i=\begin{bmatrix}4\\ 6\\ 6\\ 6\end{bmatrix}$
19	$a_{ij}=\begin{bmatrix}2 & 2 & -1 & 1\\ 4 & 3 & -1 & 2\\ 8 & 5 & -3 & 4\\ 3 & 3 & -2 & 2\end{bmatrix};\quad b_i=\begin{bmatrix}4\\ 6\\ 12\\ 6\end{bmatrix}$	**20**	$a_{ij}=\begin{bmatrix}2 & 3 & 11 & 5\\ 1 & 1 & 5 & 2\\ 2 & 1 & 3 & 2\\ 1 & 1 & 3 & 4\end{bmatrix};\quad b_i=\begin{bmatrix}2\\ 1\\ -3\\ -3\end{bmatrix}$
21	$a_{ij}=\begin{bmatrix}2 & 5 & 4 & 1\\ 1 & 3 & 2 & 1\\ 2 & 10 & 9 & 9\\ 3 & 8 & 2 & 2\end{bmatrix};\quad b_i=\begin{bmatrix}20\\ 11\\ 40\\ 37\end{bmatrix}$	**22**	$a_{ij}=\begin{bmatrix}3 & 4 & 1 & 2\\ 3 & 5 & 3 & 5\\ 6 & 8 & 1 & 5\\ 3 & 5 & 3 & 7\end{bmatrix};\quad b_i=\begin{bmatrix}-3\\ -6\\ -8\\ -8\end{bmatrix}$
23	$a_{ij}=\begin{bmatrix}3.2 & 5.4 & 4.2 & 2.2\\ 2.1 & 3.2 & 3.1 & 1.1\\ 1.2 & 0.4 & -0.8 & -0.8\\ 4.7 & 10.4 & 9.7 & 9.7\end{bmatrix};\quad b_i=\begin{bmatrix}2.6\\ 4.8\\ 3.6\\ -8.4\end{bmatrix}$	**24**	$a_{ij}=\begin{bmatrix}3 & -2 & -5 & 1\\ 2 & -3 & 1 & 5\\ 1 & 2 & 0 & -4\\ 1 & -1 & -4 & 9\end{bmatrix};\quad b_i=\begin{bmatrix}3\\ -3\\ -3\\ 22\end{bmatrix}$
25	$a_{ij}=\begin{bmatrix}1 & 1 & -6 & -4\\ 3 & -1 & -6 & -4\\ 2 & 3 & 9 & 2\\ 3 & 2 & 3 & 8\end{bmatrix};\quad b_i=\begin{bmatrix}6\\ 2\\ 6\\ -7\end{bmatrix}$	**26**	$a_{ij}=\begin{bmatrix}2 & -1 & 1 & 2\\ 6 & -4 & 2 & 4\\ 6 & -3 & 4 & 8\\ 4 & -2 & 1 & 1\end{bmatrix};\quad b_i=\begin{bmatrix}2\\ 3\\ 9\\ 1\end{bmatrix}$
27	$a_{ij}=\begin{bmatrix}12 & 14 & -15 & 24\\ 16 & 18 & -22 & 29\\ 18 & 20 & -21 & 32\\ 10 & 12 & -16 & 20\end{bmatrix};\quad b_i=\begin{bmatrix}5\\ 8\\ 9\\ 4\end{bmatrix}$	**28**	$a_{ij}=\begin{bmatrix}24 & 14 & 30 & 40\\ 36 & 25 & 45 & 61\\ 48 & 28 & 60 & 82\\ 60 & 35 & 75 & 99\end{bmatrix};\quad b_i=\begin{bmatrix}28\\ 43\\ 58\\ 69\end{bmatrix}$
29	$a_{ij}=\begin{bmatrix}5 & -3 & 2 & 4\\ 4 & -2 & 3 & 7\\ 8 & -6 & -1 & -5\\ 7 & -3 & 7 & 17\end{bmatrix};\quad b_i=\begin{bmatrix}3\\ 1\\ 9\\ 15\end{bmatrix}$	**30**	$a_{ij}=\begin{bmatrix}2 & -1 & 3 & 4\\ 4 & -8 & 5 & 6\\ 6 & -5 & 7 & 8\\ 12 & -4 & -9 & 10\end{bmatrix};\quad b_i=\begin{bmatrix}5\\ 7\\ 9\\ 13\end{bmatrix}$

3.Построение графиков функций

3.1 Задание графиков функций

Построить не менее трех видов диаграмм, задав заголовок, сетки, оси согласно выбранному варианту.

Таблица 2.1

№ п/п	Функция $f(x)$	Отрезок [a; b]
1	$f(x) = x/\ln x$	[1.2; 4]
2	$f(x) = x - 2\sin x$	[0; π/2]
3	$f(x) = (x^2-1)/(x^2+1)$	[-2; 2]
4	$f(x) = 3x^4 - 4x^3 + 12x^2 + 1$	[-2; 2]
5	$f(x) = x - 2\ln x$	[1; 3]
6	$f(x) = e^x \cos x$	[π; 3π/2]
7	$f(x) = (1 - x + x^2)/(1 + x - x^2)$	[0; 1]
8	$f(x) = -\sqrt{2x - x^2}$	[0; 2]
9	$f(x) = (x-2)^5(2x+1)^4$	[-0.5; 1.5]
10	$f(x) = x^x$	[0,1; 1.0]
11	$f(x) = e^{-1/x^2}$	[-0.5; 0,5]
12	$f(x) = x\sqrt{1 - x^2}$	[-1.0; 0]
13	$f(x) = ((e^x + e^{-x})/2)^3 + 1$	[-0.5; 0.5]
14	$f(x) = -x/(x^3+2)$	[0.5; 1.5]
15	$f(x) = x^2/\sqrt[3]{x^3-4}$	[1.6; 2.2]
16	$f(x) = \sqrt{\lvert x^2-3\rvert}/x$	[1; 2]
17	$f(x) = x^2\sqrt{\lvert x^2-1\rvert}$	[1.1; 1.6]
18	$f(x) = 1/(\sin x + \cos x)$	[0; π/3]
19	f(x)=sin x^2+cos x^2	[0;π/2]
20	f(x)=1/(sin x^2+cos x^2)	[0;π/2]
21	f(x)=sin x^3+cos x^3	[0;π/2]
22	f(x)=sin^2 x+cos ^2x	[0;π/2]
23	f(x)=sin^3 x+cos ^3x	[0;π/2]
24	f(x)=sin^4 x+cos ^4x	[0;π/2]
25	f(x)=sin^5 x+cos ^5x	[0;π/2]
26	f(x)=sin^6 x+cos ^6x	[0;π/2]
27	f(x)=ln x^2+cos x^2	[0;π/2]
28	f(x)=1/(ln x^2+cos x^2)	[0;π/2]
29	f(x)=sin x^3+ln x^3	[0;π/2]
30	f(x)=sin^2 x+ln ^2x	[0;π/2]

3.2 Построение графиков функции одной переменной

Построение графика одной переменной $y(x)=e^{-x} * \sin 10x$, определённой на отрезке [0,1].

Вывод отображения функции в виде графика состоит из следующих этапов:

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

Команды для задания вектора x и вычисления функции лучше завершить точкой с запятой для подавления вывода в командное окно их значений.

После команды *plot* точку с запятой ставить не требуется, так как она ничего не выводит в командное окно. Для поэлементного умножения необходимо использовать точку со звёздочкой (.*).

```
>>x=0:0.05:1;
>>y=exp(-x).*sin(10*x);
>>plot(x,y)
```

После выполнения команд на экране появится окно Figure1 с графиком функции.

Сравнение нескольких функций удобно производить, отобразив их графики на одних осях. Например, на отрезке [-1, -0,3] построить графики функций

$f(x)=\sin(1/x^2)$, $g(x)=\sin(1.2/ x^2)$.

```
>> x=-1: 0.005: -0.3;
>>f=sin(x.^-2);
>> g=sin(1.2*x.^-2);
>>plot(x,f,x,g)
```

Графики можно вывести разным цветом. Команда *plot* позволяет задать стиль и цвет линий, например,

```
>>plot(x,f,'k-'x,g,'k:')
```

Первый график выводится сплошной чёрной линией, а второй - чёрной пунктирной. Параметр k обозначает цвет линии чёрный, дефис- сплошную линию, а двоеточие – пунктирную линию (таблица 1.1).

Таблица 1.1- Параметры графика

Параметр цвета	Цвет линий графика	Тип маркера	Назначение маркера	Параметр линии	Тип линии
y	жёлтый	.	точка	-	сплошная
m	розовый	∘	кружок	:	пунктирная
c	голубой	×	крестик	-.	штрих-пунктирная
r	красный	+	знак «плюс»	--	штриховая
g	зелёный	*	звёздочка		
b	синий	s	квадрат		
w	белый	d	ромб		
k	чёрный	v	треугольник		

Например, для построения первого графика красными точечными маркерами без линии, а второго графика пунктирной черной линией следует использовать команду

```
>>plot (x, f, 'r.', x, g, 'k:').
```

4. Аппроксимация функций

Одним из распространенных и практически важных случаев связи между
аргументом и функцией является задание этой связи в виде некоторой таблицы $\{x_i ; y_i\}$, например, экспериментальные данные. На практике часто приходится использовать табличные данные для приближенного вычисления y при любом значении аргумента x (из некоторой области). Этой цели служит задача о приближении (аппроксимации) функций: данную функцию $f(x)$ требуется приближенно заменить некоторой функцией $g(x)$ так, чтобы отклонение $g(x)$ от $f(x)$ в заданной области было наименьшим. Функция $g(x)$ при этом называется аппроксимирующей. **Если приближение строится на заданном дискретном множестве точек $\{x_i\}$, то аппроксимация называется точечной.** К ней относятся интерполирование, среднеквадратичное приближение и др. **При построении приближения на непрерывном множестве точек (например, на отрезке [a, b]) аппроксимация называется непрерывной или интегральной.**

MATLAB имеет мощные средства точечной и непрерывной аппроксимации с визуализацией результата. Рассмотрим наиболее важную точечную аппроксимацию (обработка
экспериментальных данных).

Пример 4. Используя линейную и полиномиальную аппроксимации, получить эмпирические формулы для функции $y=f(x)$, заданной в табличном виде:

x_i 0.75 1.50 2.25 3.00 3.75

y_i 2.50 1.20 1.12 2.25 4.28

Оценить погрешность эмпирических формул.

Протокол программы. В окне команд набираются значения x_i и y_i. Далее выполняется команда построения графика только узловых точек.

>> *x* = [0.75, 1.50, 2.25, 3.00, 3.75];
>> *y* = [2.50, 1.20, 1.12, 2.25, 4.28];

>> *plot* (*x*, *y*, ′ 0 ′);

Появляется окно с символами ′ 0 ′ на месте узловых точек (рис. 2.1).

Появляется окно с символами ′ 0 ′ на месте узловых точек (рис. 2.1).

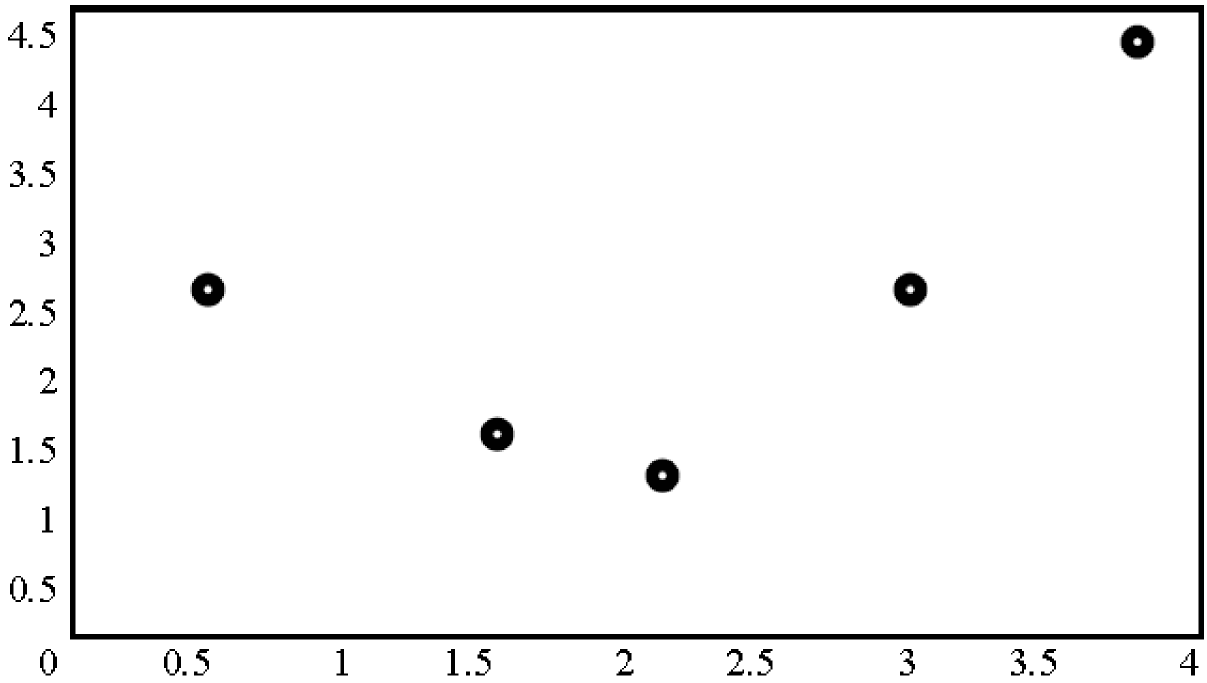

Рис. 2.1

Внимание. Следует помнить, что при полиномиальной аппроксимации максимальная степень полиноме на 1 меньше числа экспериментальных точек!

На панели инструментов окна графика узловых точек в меню **Tools** исполняем команду **Basic Fitting**. Появляется окно **Основной Монтаж**. В этом окне птичкой отмечаются необходимые данные аппроксимации. В частности, можно задать следующие операции:

- *показать уравнение* аппроксимирующей функции $y = g(x)$;
- *выбрать метод подбора*: сплайн интерполяции

эрмитовская интерполяция

линейный

квадратные

кубический

В нашей задаче выбираем линейную и полиномиальную аппроксимации.

В окне графика появляются соответствующие графики разноцветом и формулы аппроксимирующих функций (рис. 2.2).

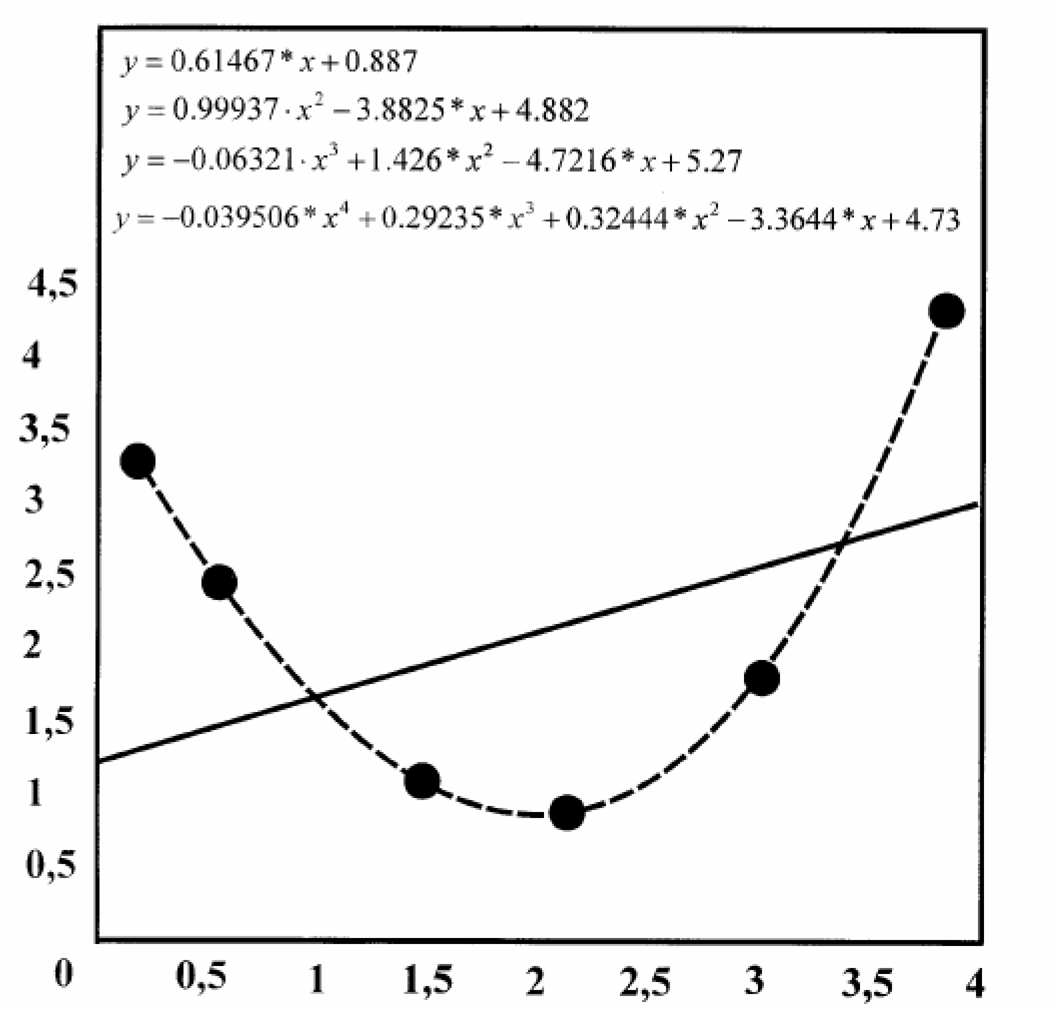

Рис. 2.2

Чтобы узнать погрешность аппроксимации, надо *отметить птичкой* параметр ***График остатка*** в окне *Основной Монтаж,* и *Показать норму остатков.* График погрешностей с нормами можно вынести в *отдельное окно*, или вместе с графиком и аппроксимирующих функций – суб-график. Норма погрешностей указывает на статистическую оценку среднеквадратической погрешности. Чем она меньше, тем точнее полученная аппроксимирующая функция $y = g(x)$. В нашем примере:

Linear : norm of residuals **(**норма погрешности) **= 2.1061**
Quadratic : norm of residuals **= 0.10736**
Cubic : norm of residuals **= 0.035857**
4 *th degree : norm of residuals* = 9.6305*e*-015.

График погрешностей можно выводить в виде диаграмм (зоны), линий

(линии) или отдельных точек (фрагменты). Сам график погрешностей представляет собой зависимость разности $g(x) - f(x)$ в условных точках, соединенных прямыми линиями.
Кроме линейной и полиномиальной аппроксимации можно выбрать *сплайн- аппроксимацию* – когда на каждом интервале приближения используется *кубический полином* с новыми коэффициентами. В этом случае нельзя получить выражение для аппроксимирующей функции, т. е. такая аппроксимация является неполной. Аналогичными свойствами обладает и *Эрмитовая аппроксимация*. Она имеет только графическую интерпретацию.

Варианты заданий. Получить эмпирические формулы и оценить их погрешность для функции $y = f(x)$, заданной таблично. Данные, согласно вашему варианту, взять из таблицы 2.3.

Таблица 2.3

1.	x_i	-3	-2	-1	0	1	2	3
	y_i	-0.71	-0.01	0.51	0.82	0.88	0.51	0.49
2.	x_i	-6.6	-5.38	-3.25	-1.76	2.21	3.6	4.5
	y_i	2.89	1.41	0.29	-0.41	-0.69	-0.7	1.2
3.	x_i	0	1	2	3	4	5	6
	y_i	-0.31	0.9	2.11	3.3	4.51	5.73	6.93
4.	x_i	-2	-1	0	1	2	3	4
	y_i	7.1	3.9	1.1	0.8	3.1	4.5	5.3
5.	x_i	-2	-1	-0.5	0	1.5	2	3.5
	y_i	5.9	2.8	2.1	3.2	6.1	7.6	4.3
6.	x_i	-3	-2	-1	0	1	2	3
	y_i	3.1	0.9	0.9	2.8	7.1	6.5	4.1
7.	x_i	0	1	2	3	4	5	6
	y_i	10.0	7.5	5.5	4.0	3.0	2.0	2.24
8.	x_i	-2	-1	0	1.5	2.3	2.6	2.9
	y_i	4.2	5.6	6.8	7.2	9.4	10.5	11.8
9.	x_i	10.0	12.0	13.0	15.0	18.0	20.0	21.0
	y_i	0.66	0.89	1.24	1.36	1.56	1.76	1.92
10.	x_i	22.0	24.0	27.0	30.0	31.0	35.0	40.0
	y_i	1.24	1.37	1.46	1.26	1.66	1.84	1.99
11.	x_i	-7.0	-6.0	-5.0	-4.0	-3.0	-2.0	-1.0
	y_i	22.6	24.7	25.6	24.6	23.5	21.8	19.3

Продолжение таблицы 2.3

12.	x_i	-25.0	-23.0	-21.0	-18.0	-17.2	-15.4	-14.0
	y_i	0.76	0.74	0.61	0.58	0.84	0.92	1.22
13.	x_i	-4.0	-3.0	-2.0	-1.0	0.0	1.0	2.0
	y_i	1.71	1.56	1.24	1.36	1.78	2.21	4.31
14.	x_i	-22.0	-20.0	-18.0	-16.0	-14.0	-12.0	-10.0
	y_i	-2.26	-1.84	-1.92	-1.76	-1.56	-1.64	-1.34
15.	x_i	23.0	24.0	25.0	26.0	27.0	28.0	29.0
	y_i	1.26	1.37	1.44	1.56	1.15	1.28	1.06
16.	x_i	30.0	33.0	35.0	37.0	39.0	41.0	43.0
	y_i	-2.6	-3.7	-2.5	-4.3	-2.3	-5.6	-1.9
17.	x_i	44.0	45.0	46.0	47.0	48.0	49.0	50.0
	y_i	2.24	3.46	5.36	1.89	1.76	1.54	2.12
18.	x_i	52.0	54.0	56.0	58.0	60.0	62.0	64.0
	y_i	-1.28	-1.33	-1.44	-1.67	-1.77	-2.81	-2.16
19.	x_i	2.2	2.6	3.0	3.4	3.8	4.2	4.6
	y_i	1.88	1.65	1.61	1.73	1.56	1.24	1.99
20.	x_i	5.1	5.3	5.5	5.7	5.9	6.1	6.3
	y_i	-2.8	-3.6	-5.7	-3.4	-1.9	-1.7	-1.5
21.	x_i	7.15	7.35	7.55	7.75	7.95	8.15	8.35
	y_i	-2.2	-3.6	-1.7	-2.8	-1.6	-4.5	-2.2
22.	x_i	9.1	9.2	9.3	9.4	9.5	9.6	9.7
	y_i	1.48	1.16	2.08	1.96	1.81	2.31	5.61
23.	x_i	-10.2	-10.1	-10.0	-9.9	-9.8	-9.7	-9.6
	y_i	-6.5	-7.8	-10.2	-5.4	-4.6	-9.5	-10.3
24.	x_i	11.0	14.0	17.0	20.0	23.0	26.0	29.0
	y_i	1.2	1.6	1.9	1.1	1.16	1.24	1.36
25.	x_i	-50.0	-48.0	-46.0	-44.0	-42.0	-40.0	-38.0
	y_i	1.23	1.32	1.57	1.19	1.16	1.10	2.28
26.	x_i	-36.0	-34.0	-32.0	-30.0	-28.0	-26.0	-24.0
	y_i	1.1	1.3	2.1	1.9	1.7	1.5	1.8
27.	x_i	21.0	23.0	24.0	28.0	31.0	32.0	36.0
	y_i	1.24	1.37	1.56	1.64	1.84	1.26	1.14
28.	x_i	10.0	13.0	17.0	22.0	28.0	35.0	43.0
	y_i	1.21	1.36	1.51	1.84	1.06	1.21	1.36
29.	x_i	-1.0	0.0	3.0	5.0	8.0	12.0	15.0
	y_i	-2.1	-3.6	1.2	-4.3	1.8	2.6	-0.2
30.	x_i	-8.0	-7.0	-5.0	-3.0	-1.0	2.0	5.0
	y_i	1.36	1.88	2.45	-2.1	-10.2	-4.4	1.16

5. Численное решение нелинейных уравнений

Задача нахождения корней нелинейных уравнений встречается в разных областях научно-технических исследований. Проблема формулируется следующим образом. Пусть задана непрерывная функция $f(x)$ и требуется найти корень уравнения $f(x) = 0$.

Будем предполагать, что имеется интервал изменения x $[a; b]$, на котором необходимо исследовать функцию $f(x)$ и найти значение x_0, при котором $f(x_0)$ равно или весьма мало отличается от нуля.

Данная задача в системе MATLAB может быть решена следующим образом. Вначале необходимо построить график функции $f(x)$ на заданном интервале и убедиться в существовании корня или нескольких корней. Затем применить программы поиска корней. Если существует *один* корень и график $f(x)$ *пересекает* ось *ox*, то можно применить программу ***fzero***. Если $f(x)$ имеет больше одного корня, и может касаться, и пересекать ось *ox,* то следует применить более мощную программу ***fsolve*** из пакета ***Optimization Toolbox***, которая решает задачу методом наименьших квадратов. Программа ***fzero*** использует известные численные методы: деление отрезка пополам, секущей и обратной квадратичной интерполяции.

Пример 7. Найти корень нелинейного уравнения $10^x + 2x - 100 = 0$ на интервале [1.0; 2.0].

Протокол программы

>> % Строим график заданной функции

>> *x* = 1.0 : 0.001 : 2.0; *y* = 10.0.^*x* + 2.0x* – 100.0;**

>> *plot* (*x*, *y*) ; *grid on*

Появляется окно с графиком функции $10^x + 2x - 100$ (см. рис. 2.4), из которого следует, что корень функции на заданном интервале существует. Для точного определения корня применяем ***fzero*** и ***fsolve***.

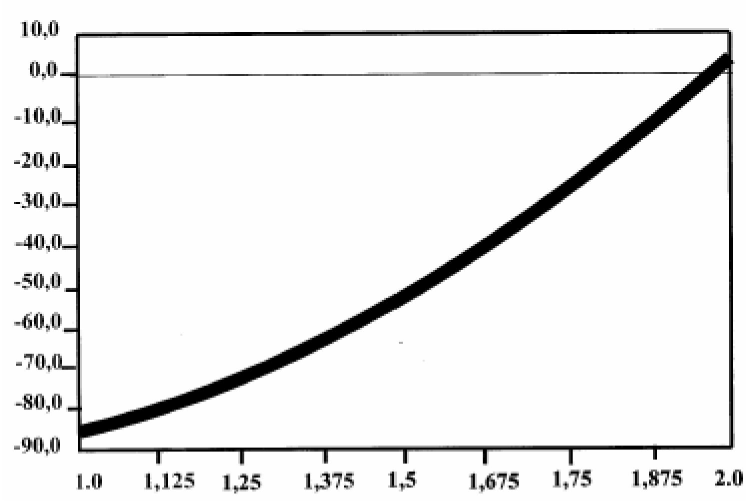

Рис. 2.4

>> X1 =*fzero* (' (10.0.^*x* + 2.0x* – 100.0) ', [1.0 2.0])**

<u>Результат решения</u>

<u>X1 =</u>
       1.9824

>> X2 =*fsolve* (' (10.0.^*x* + 2.0x* – 100.0) ', 1.0 : 2.0)**

<u>Результат решения</u>

<u>*X*2 =</u>
       1.9824 1.9824

<u>Варианты заданий</u>. Построить график и найти корень нелинейного уравнения. Данные взять из таблицы 2.7.

Таблица 2.7

№ п/п	Уравнение $f(x) = 0$	Отрезок [a; b]
1	$x arctg - 1 = 0$	$[1.0; \sqrt{3}]$
2	$e^{x-2} - \ln(x+2) = 0$	[2.0; 3.0]
3	$x^3 - 9x^2 + 5x - 6 = 0$	[8.0; 9.0]
4	$e^{x} - \frac{1}{x} - 1 = 0$	[0.5; 1.0]

Продолжение таблицы 2.7

5	$arctg2x-\frac{1}{1+x}=0$	[0.0; 1.0]
6	$e^{x}-\ln x-20=0$	[3.0; 3.2]
7	$\sqrt{x}-tgx(1-x)=0$	[0.0; 1.0]
8	$\sin\sqrt{x}-\cos\sqrt{x}+2\sqrt{x}=0$	[0.0; 0.2]
9	$x^{4}+2x^{3}-x-1=0$	[0.8; 1.0]
10	$x^{3}-e^{4x}-5.5=0$	[2.6; 3.0]
11	$x^{6}-3x^{2}+x-1=0$	[1.0; 1.5]
12	$\sqrt[3]{5-x}-x=0$	[1.0; 2.0]
13	$x^{2}-\ln x=0$	[0.0; 1.0]
14	$x^{2}-\cos x=0$	[0.0; 1.0]
15	$\ln x-arctgx=0$	[3.0; 4.0]
16	$x^{2}arctgx-1=0$	[1.0; 1.2]
17	$x^{2}+\ln x-4=0$	[1.0; 2.0]
18	$x-arctg\sqrt[3]{x}=0$	[0.0; 1.0]
19	$x^{2}-e^{x}-2=0$	[-0.2; -0.1]
20	$x^{2}-\ln(x+1)=0$	[0.1; 0.9]
21	$x^{5}-x-2=0$	[1.0; 1.4]
22	$x-2-\sqrt[4]{x}=0$	[3.0; 4.0]
23	$x-tgx=0$	[0.0; 1.5]
24	$x+\ln x-0.5=0$	[0.0; 1.0]
25	$\ln x+\sqrt{x}=0$	[0.1; 1.0]
26	$\sqrt{x}-\cos\sqrt{x}=0$	[0.4; 0.6]
27	$x^{2}-\ln(1+x^{2})-9.75=0$	[3.0; 4.0]
28	$x+\sqrt[3]{x}-6.09=0$	[4.0; 5.0]
29	$x^{3}-\sqrt{x}-9.5=0$	[2.0; 3.0]
30	$\arccos 2x-x^{2}-0.35=0$	[0.0; 0.48]

6. Численное решение оптимизационных задач

Под оптимизацией понимают процесс выбора наилучшего варианта из всех возможных. С точки зрения инженерных расчетов методы оптимизации позволяют выбрать наилучший вариант конструкции, наилучшее распределение ресурсов, минимальный урон природной среде и т. п. В процессе решения задачи оптимизации необходимо найти оптимальные значения некоторых параметров, их называют проектными параметрами. Выбор оптимального решения проводится с помощью некоторой функции, называемой целевой функцией. Целевую функцию можно записать в виде

$u\, f\left(x, x, ..., x_n\right)$, (2.10)

где x_1, x_2, … , x_n – проектные параметры.

Можно выделить 2 типа задач оптимизации – *безусловные* и *условные*. Безусловная задача оптимизации состоит в отыскании максимума или минимума функции (2.10) от *n* действительных переменных и определении соответствующих значений аргументов на некотором множестве G *n*-мерного пространства. Обычно рассматриваются задачи минимизации; к ним легко сводятся и задачи на поиск максимума путем замены знака целевой функции на противоположный. Условные задачи оптимизации – это такие, при формулировке которых задаются некоторые условия (ограничения) на множестве G. Здесь рассмотрим только безусловные задачи оптимизации.
Поиск минимума функции одной переменной.
Для решения этой задачи используются методы золотого сечения или параболической интерполяции (в зависимости от формы задания функции), реализованные в программе ***fminbnd***.

Пример 8. Найти и вывести на печать минимальное значение функции
$f(x) = 24 - 2x/3 + x^2/30$ на [5; 20].
Строим график этой функции, чтобы убедиться в наличии минимума на заданном интервале.
Протокол программы
>> *x* = 5.0 : 0.001 : 20.0 ; *y* = 24 – 2* *x*/3 + *x*.^2/30 ;
>> *plot*(*x, y*) ; *grid on*
Появляется окно с графиком этой функции (рис. 2.5), где отмечаем наличие минимума.

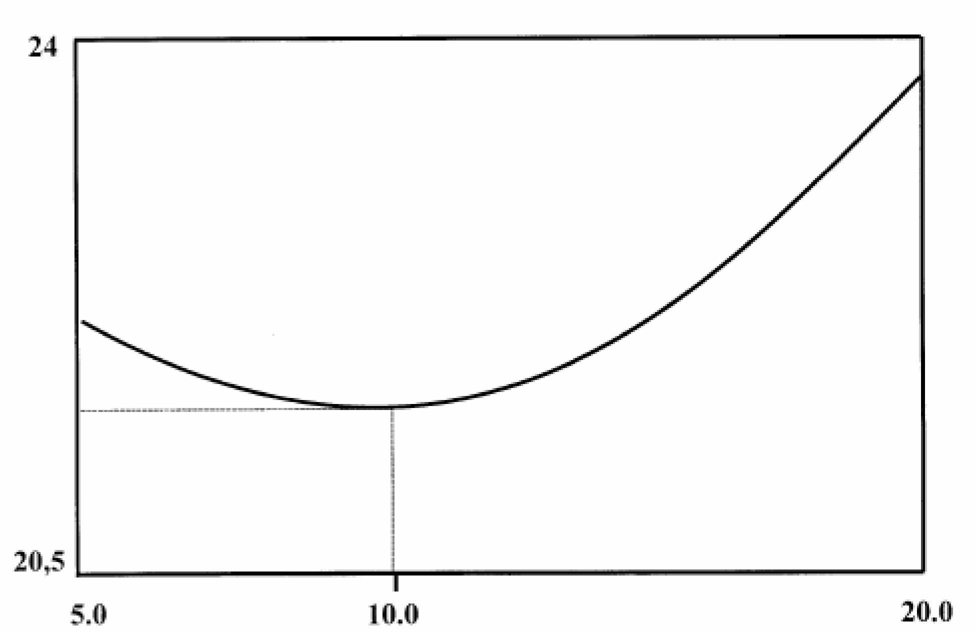

Рис. 2.5

Далее, для точного определения координаты и значения минимума привлекаем программу ***fminbnd***.

>> **[*x*, *y*] = *fminbnd* (' (24.0 – 2* *x*/3 + *x*.^2/30) ', 5.0, 20.0)**

Результат поиска

x =

10.0000

y =

20.6667

<u>Варианты заданий</u>. Найти и вывести на печать координату и минимальное значение функции *f*(*x*) на [*a; b*]. Данные взять из таблицы 2.8.

Таблица 2.8

№ п/п	Функция *f*(*x*)	Отрезок [a; b]
1	$f(x)=x/\ln x$	[1.2; 4]
2	$f(x)=x-2\sin x$	[0; π/2]
3	$f(x)=(x^2-1)/(x^2+1)$	[-2; 2]
4	$f(x)=3x^4-4x^3+12x^2+1$	[-2; 2]
5	$f(x)=x-2\ln x$	[1; 3]
6	$f(x)=e^x\cos x$	[π; 3π/2]
7	$f(x)=(1-x+x^2)/(1+x-x^2)$	[0; 1]
8	$f(x)=-\sqrt{2x-x^2}$	[0; 2]
9	$f(x)=(x-2)^5(2x+1)^4$	[-0.5; 1.5]
10	$f(x)=x^x$	[0,1; 1.0]
11	$f(x)=e^{-1/x^2}$	[-0.5; 0,5]
12	$f(x)=x\sqrt{1-x^2}$	[-1.0; 0]
13	$f(x)=((e^x+e^{-x})/2)^3+1$	[-0.5; 0.5]
14	$f(x)=-x/(x^3+2)$	[0.5; 1.5]
15	$f(x)=x^2/\sqrt[3]{x^3-4}$	[1.6; 2.2]
16	$f(x)=\sqrt{\lvert x^2-3\rvert}/x$	[1; 2]
17	$f(x)=x^2\sqrt{\lvert x^2-1\rvert}$	[1.1; 1.6]
18	$f(x)=1/(\sin x+\cos x)$	[0; π/3]
19	$f(x)=x/2+arctgx$	[0.5; 1.2]
20	$f(x)=xe^{-x^2/2}$	[-1.5; -0.5]
21	$f(x)=e^{-1/x^2}/x$	[-2.0; -1.0]
22	$f(x)=-\lvert x\rvert^3e^{-x^2/2}$	[-2.0; -1.0]
23	$f(x)=x^2\ln x$	[0.1; 1.0]
24	$f(x)=x\ln^2\lvert x\rvert$	[-0,05; -0.2]
25	$f(x)=xarctgx$	[-0.5; 0.5]
26	$f(x)=\sin x+\cos x$	[π; 3π/2]
27	$f(x)=-\ln x/x^2$	[1.0; 2.0]
28	$f(x)=x^2\ln^2 x$	[0.1; 0.5]
29	$f(x)=\sin x/x$	[π; 2π]
30	$f(x)=-x^{1/x}$	[2.0; 3.0]

7. Поиск минимума функций нескольких переменных

Данная задача значительно сложнее первой. Рассмотрим ее решение на примере функции двух переменных. Алгоритм может быть распространен на функции большего числа переменных. Для минимизации функций нескольких переменных MATLAB использует симплекс – метод Нелдера-Мида. Данный метод является одним из лучших методов поиска минимума функций многих
переменных, где не вычисляются производные или градиент функции. Он сводится к построению симплекса в n-мерном пространстве, заданного $n + 1$ вершиной. В двухмерном пространстве симплекс является треугольником, а в трехмерном – пирамидой. На каждом шаге итераций выбирается новая точка решения внутри или вблизи симплекса. Она сравнивается с одной из вершин симплекса. Ближайшая к этой точке вершина симплекса заменяется этой точкой. Таким образом, симплекс перестраивается и позволяет найти новое, более точное положение точки решения. Алгоритм поиска повторяется, пока размеры симплекса по всем переменным не станет меньше заданной погрешности решения. Программу, реализующую симплекс-методы Нелдера-Мида, удобно использовать в следующей записи

[*x*, min *f*] = *f* min search (…),
где *x* – вектор координат локального минимума;
min f – значение целевой функции в точке минимума.
Саму целевую функцию удобно представить с помощью дескриптора @ в М-файле.

Пример 9. Найти и вывести на печать координаты и значение минимума функции двух переменных $f(x, y) = (x^2 + y^2 – 3)^2 + (x^2 + y^2 – 2x – 3)^2 + 1$, если начальная точка поиска имеет координаты M_0 (1; 1). Анализ функции показывает,
что *min f* = 1 $x = 0$, $y = 3 = 1.73205$.
Строим трехмерный график этой функции, чтобы убедиться в наличии минимума. Возьмем интервал $x \in [-1; 1]$; $y \in [1; 3]$.

Протокол программы
>> [X,Y] = *mesh grid* ([-1 : 1, 1 : 3]) ;
>> Z = (X.^2 + Y.^2 – 3).^2 + (X.^2 + Y.^2 – 2*X – 3).^2 + 1 ;
>> *plot* 3 (X,Y,Z)
После построения трехмерного графика выполняем поиск минимума. В М-файле программируем целевую функцию
function f = F *xy*(*x*)
f = (*x*(1) ^ 2 + *x*(2) ^ 2 –3) ^2 – 3) ^2 + (*x*(1) ^ 2 + *x*(2) ^ 2 –2* *x*(1) – 3)^2 + 1;
Решаем поставленную задачу в окне команд

>> [*xmin, mi f*] = *fminsearch* (@ F*xy*, [1; 1])
Результаты поиска
xmin =
- 0.0000 1.7320
minf =
1.0000
Как видно, результаты решения задачи точные.
Варианты заданий. Найти и вывести на печать координаты и минимальное значение функции двух переменных. Поиск начать с точки M_0 (x_0, y_0). Данные взять из таблицы 2.9

Таблица 2.9

№ п/п	Функция $f(x, y)$	Координаты начальной точки $M_0(x_0, y_0)$.
1	2	3
1	$(2x^2 - y - 3)^2 + x^2 + 2x + 2$	(1; 1)
2	$(xy + 2)^2 + y^2 + 2y + 4$	(2; 2)
3	$(x^2y^2 - y + 2)^2 + x^2 + 1$	(2; 2)
4	$(3x^2 + 2y^2 - 1)^2 + (xy - 3)^2$	(2; 2)
5	$(2x^2 - 7y^2 - 2)^2 + (x^2 + y^2 - 20)^2 + 3$	(2; 2)
6	$(x^2 + y^2 - 2x - 3)^2 + (x^2 + y^2 - 2y - 3)^2$	(2; 2)
7	$(x^2 - 6x + y^2 + 8)^2 + x^2y^2 + 1$	(2; 2)
8	$(x^2 - y = 2)^2 + (x - y + 3)^2$	(2; 2)
9	$\ln(1 + x^2 + y^2)^2 + (x - y - 1)^2$	(2; 2)

Продолжение таблицы 2.9

1	2	3
10	$(x^2 + y^2 - 1)^2 + (x^2 - 6x + y^2 + 8)^2$	(2; 2)
11	$x^3 + y^3 - 3xy$	(0.5; 0.5)
12	$x^2 + xy + y^2 - 3x - 6y$	(0.5; 3.5)
13	$-xy^2(1 - x - y)$	(0; 0)
14	$3x^2 - x^3 + 3y^2 + 4y$	(0.1; -1.0)
15	$xy + 50/x + 20/y$	(4; 1)
16	$x^2 + y^2 - 2\ln x - 18\ln y$	(0.5; 2.5)
17	$x^3 + 3xy^2 - 15x - 12y$	(1.5; 0.5)
18	$2x^3 - xy^2 + 5x^2 + y^2$	(0.5; 0.5)
19	$(2x^2 + y^2)e^{-(x^2+y^2)}$	(0.3; 0.3)
20	$-2 + \sqrt[3]{x^2 + y^2}$	(0.25; 0.25)
21	$(x - 1)^2 + (y - 2)^2 - 1$	(0.5; 1.5)
22	$\sin(x^2 + y^2 - 0.5)$	(0.5; 0.5)
23	$x^2 - xy + y^2 + 2x - 2y + 1$	(-1.5; 0.5)
24	$xy(x + y - 4)$	(1.0; 1.0)
25	$x^3y^2(x + y - 5)$	(2.0; 1.5)
26	$x^2 + xy + y^2 + 1/x + 1/y$	(0.2; 0.3)
27	$-(\sin x + \sin y + \sin(x + y))$	(π/4; π/4)
28	$-\sin x \sin y \sin(x + y)$	(π/4; π/4)
29	$x^3y^3 - 9xy + 1$	(2.5; 2.5)
30	$x^4 + y^4 - 2x^2 + 4xy - 2y^2 + 1$	(1.0; -1.0)

8. Подготовку отчета по результатам выполнения лабораторной работы провести по каждому разделу настоящих методических указаний, подкрепив их выводами по разделу. Предварительно рекомендуется ознакомиться с прилагаемой литературой по MatLab.

Перечень задач по лаб. работе