Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Кафедра геометрии и компьютерных наук

Т.А. Фомина, О.Н. Казакова

СИСТЕМЫ АНАЛИТИЧЕСКИХ ВЫЧИСЛЕНИЙ

Часть 1

Методические указания

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по программам высшего образования по направлениям подготовки 01.03.01 Математика, 02.03.02 Фундаментальная информатика и информационные технологии

Оренбург
2017

УДК 512.5 (076.5)
ББК 22.14.я7
Ф 76

Рецензент – кандидат физико-математических наук, доцент О.А. Пихтилькова

Фомина, Т.А.

Ф 76 Системы аналитических вычислений: методические указания к лабораторным работам в 2 ч. Ч. 1 / Т.А. Фомина, О.Н. Казакова; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2017. – 59 с.

Методические указания содержат материал, предназначенный для выполнения лабораторных работ по курсу «Системы аналитических вычислений».

Методические указания предназначены для обучающихся по направлениям подготовки 01.03.01 Математика, 02.03.02 Фундаментальная информатика и информационные технологии, и составлены в соответствии с утвержденными рабочими программами дисциплины «Системы аналитических вычислений». Материалы могут быть полезны преподавателям высших учебных заведений, ведущим соответствующую дисциплину и изучающим эту дисциплину студентам.

УДК 512.5 (076.5)
ББК 22.14.я7

Содержание

Введение

Дисциплина «Системы аналитических вычислений» изучается обучающимися по направлениям подготовки 01.03.01 Математика, 02.03.02 Фундаментальная информатика и информационные технологии и направлена на изучение программирования с использованием математической системы Matlab.

Рабочей программой дисциплины предусмотрено выполнение студентами серии лабораторных работ, предназначенных для знакомства студентов с пакетом Matlab и получения навыков работы с ним.

Методические указания содержат материал, предназначенный для выполнения лабораторных работ и способствующий активной самостоятельной работе студентов.

Лабораторный цикл состоит из 11 работ. Каждая лабораторная работа содержит необходимый для выполнения индивидуальных заданий теоретический материал, практическую часть и список контрольных вопросов для самопроверки. Для проведения лабораторных работ необходим компьютер с установленным пакетом Matlab. От студентов требуются первичные навыки работы в среде Windows и знания основ программирования.

Первая часть методических указаний включает в себя пять лабораторных работ, позволяющих отработать основные навыки работы с данным математическим пакетом: основы работы с Matlab, операции с векторами и матрицами в Matlab и графические возможности среды Matlab.

Вторая часть включает в себя шесть лабораторных работ по двум разделам: программирование в пакете Matlab и символьные вычисления в среде Matlab.

Материалы могут быть полезны преподавателям высших учебных заведений, ведущим соответствующую дисциплину, и изучающим эту дисциплину студентам.

1 Основы работы с Matlab

1.1 Лабораторная работа №1. Вычисление арифметических выражений

1.1.1 Теоретические сведения

Арифметические выражения в Matlab состоят, как и в большинстве языков программирования, из чисел, знаков арифметических операций, знака ^ (возведение в степень), круглых скобок, переменных, и встроенных функций. Десятичная часть числа отделяется точкой. Для вычисления простейшего выражения следует набрать его в командной строке и нажать <Enter>. Ответ записывается в специальную переменную *ans* и результат выводится в командное окно:

```
>> 1.5+2.9
ans =
4.4000
```

После вычисления следующего выражения значение *ans* изменится. Для сохранения результатов промежуточных вычислений их следует записывать в переменные. При использовании переменных необходимо придерживаться правил:

- имя переменной может состоять из символов латинского алфавита, знака подчёркивания и цифр, но начинается обязательно с символа алфавита;
- прописные и строчные буквы различаются;
- пробел не входит в имя переменной.

В качестве знака присваивания используется =, например:

```
>> a=3.25*(0.7-3.3/5.1)+2.3^3
a =
12.3391
```

Обратите внимание, что результат сразу же выводится в командное окно. Для подавления вывода следует завершить строку с оператором присваивания точкой с запятой.

В Matlab имеется ряд констант приведенных в таблице 1.

Таблица 1 – Имена констант

Имя	Описание
ans	Результат последней операции
i, j	Мнимая единица
pi	Число π
eps	Машинная точность
realmax	Максимальное вещественное число
realmin	Минимальное вещественное число
inf	Бесконечность
NaN	Нечисловая переменная
end	Наибольшее значение индекса размерности массива

Комплексные числа вводятся при помощи буквы *i*.

```
>> b=5*(2.2+3.9i)+0.8
b =
11.8000 +19.5000i
```

Matlab автоматически переходит в область комплексных чисел, продолжая вычисления.

Matlab обладает большим набором встроенных математических функций. Некоторые из них приведены в таблице 2. При вызове математических функций аргумент заключается в круглые скобки. Полный список всех встроенных элементарных математических функций можно получить, набрав в командной строке *help elfun*. Команда *help* отображает в командном окне список разделов справочной системы. Для получения содержимого раздела необходимо указать через пробел его название после *help*, а для вывода детальной информации о какой-либо функции, следует ввести в строке с *help* имя функции.

Таблица 2 – Основные математические функции

Функция	Описание
Тригонометрические функции (аргумент задаётся в радианах)	
sin, cos, tan, cot	Синус, косинус, тангенс и котангенс
sec, csc	Секанс, косеканс
Обратные тригонометрические функции (результат вычисляется в радианах)	
asin, acos, atan, acot	Арксинус, арккосинус, арктангенс и арккотангенс
asec, acsc	Арксеканс, арккосеканс

Продолжение таблицы 2

Гиперболические функции	
sinh, cosh, tanh, coth	Гиперболические синус, косинус, тангенс и котангенс
sech, csch	Гиперболические секанс и косеканс
asinh, acosh, atanh, acoth	Гиперболические арксинус, арккосинус, арктангенс и арккотангенс;
Экспоненциальная функция, логарифмы, степенные функции	
exp	Экспоненциальная функция
log, log2, log10	Натуральный логарифм, логарифмы по основанию 2 и 10
sqrt	Квадратный корень
Модуль, знак и функции для работы с комплексными числами	
abs, sign	Модуль и знак числа
conj, imag, real	Комплексно-сопряжённое, мнимая и вещественная часть

Пусть, например, требуется найти значение выражения:

$$c = \sqrt{\frac{sin\left(\frac{4}{3}\pi x\right)+e^{0.1y}}{cos\left(\frac{4}{3}\pi x\right)+e^{0.1y}}} + \sqrt[3]{\frac{sin\left(\frac{4}{3}\pi x\right)+e^{0.1y}}{cos\left(\frac{4}{3}\pi x\right)+e^{0.1y}}}$$, при $x = 0{,}2$ и $y = 3{,}9$.

Если набирать сразу все выражение, то получается достаточно длинная строка. Для переноса на следующую строку любой команды Matlab можно использовать три идущие подряд точки, после нажатия на <Enter> среда Matlab ждет продолжения ввода:

```
>> x=0.2;
>> y=-3.9;
>> c=sqrt((sin(4/3*pi*x)+exp(0.1*y))/(cos(4/3*pi*x)+exp(0.1*y)))+...
((sin(4/3*pi*x)+exp(0.1*y))/(cos(4/3*pi*x)+exp(0.1*y)))^(1/3)
```

Проще всего решить поставленную задачу, используя промежуточные переменные:

```
>> x=0.2;
>> y=-3.9;
>> a=sin(4/3*pi*x)+exp(0.1*y);
```

>> b = cos(4/3*pi*x)+exp(0.1*y);

>> c = sqrt(a/b)+(a/b)^(1/3)

c =

2.0451

Необязательно набирать выражение для *b*, похожее на только что введенное для *a*. После ввода третьей строки нажмите клавишу <↑>. В командной строке появится предыдущее выражение, внесите в него необходимые изменения, а именно, замените sin на cos, и нажмите <Enter>. Клавиши <↑> и <↓> служат для перехода по истории команд, т.е. для занесения ранее набранных команд в командную строку, а <←>, <→>, <Home>, <End> – для перемещения в пределах командной строки. Передвижение по экрану (только для просмотра команд, а не для редактирования) осуществляется клавишами <PageUp>, <PageDown> или вертикальной полосой скроллинга.

Просмотр текущих переменных рабочей среды производится при помощи команды *whos*.

Предположим, что ранее переменным *a* и *b* были присвоены значения:

>> a=-1.34;

>> b=2.98+3.86i;

Вызовите команду *whos*, указав через пробелы имена переменных:

>> whos a b

Name	Size	Bytes	Class
a	1x1	8	double array
b	1x1	16	double array (complex)

Grand total is 2 elements using 24 bytes

В столбике Class указан тип переменной, в Bytes — число байт, выделенных под хранение значения, а Size содержит информацию о размере. После таблицы размещена строка с указанием суммарного объема памяти в байтах.

Для представления чисел на экране имеются разные форматы (таблица 3). Нужный формат может быть определен в меню *File/Preferences*, либо при помощи команды *format*, которая служит для установки формата из командной строки.

Вне зависимости от установленного формата все вычисления производятся с двойной точностью. После смены формата не требуется повторно находить значения переменных, достаточно снова вывести их значения в командном окне.

Таблица 3 – Форматы чисел

Формат	Представление
short	Число отображается с 4 цифрами после десятичной точки или в формате short e
short e	Число в экспоненциальной форме с мантиссой из 5 цифр и показателем из 3 цифр
rat	Представление в виде рационального дробного числа
long	Число с 16 десятичными цифрами
long e	Число в экспоненциальной форме с мантиссой из 16 цифр и показателем из 3 цифр
hex	Число в шестнадцатеричной форме

1.1.2 Практическая часть

Контрольные задания

1) Во всех заданиях требуется занести в некоторую переменную значения выражений при заданных $x = -1.75 \cdot 10^{-3}$ и $y = 3.1\pi$, отобразить результат в различных форматах и изучить информацию о переменных при помощи команды *whos*.

Варианты:

1. $F = \left(\dfrac{e^x \sin y + 2^x \cos y}{200x + y} \right)^{2.3} + \ln\left|\sin y\right| - \sqrt{\dfrac{e^x \sin y + 2^x \cos y}{200x + y}}$

2. $Z = \operatorname{arctg} \dfrac{\sqrt[3]{x - \sin y}}{\sqrt{1 - x^2}} - \dfrac{|x|\sqrt{1 - x^2}}{\sqrt[3]{x - \sin y}}$

3. $T = \dfrac{(\sin y + \sin 2y + \sin 3y)^4}{1 + \dfrac{\sin y + \sin 2y + \sin 3y}{e^x}} + \sqrt{1 + \dfrac{\sin y + \sin 2y + \sin 3y}{e^x}}$

4. $W = \left(1 + \dfrac{\ln y}{x + \operatorname{tg} y} \right)^{1 + \frac{x + \operatorname{tg} y}{\ln y}}$

5. $R = sh\dfrac{(x + ln\,y)^3}{\sqrt{|x - ln\,y|}} \cdot ch\left[(x + ln\,y)\sqrt{|x - ln\,y|}\right]$

6. $H = \dfrac{\sqrt{cos\,2y + sin\,4y + \sqrt{e^x + e^{-x}}}}{\left(e^{-x} + e^x\right)^3 (sin\,4y + cos\,2y - 2)^2}$

7. $Q = \sqrt{e^x\,sin\,y + e^{-x}\,cos\,y + \sqrt{1 + \dfrac{e^x\,sin\,y + e^{-x}\,cos\,y}{tg\,y}}}$

8. $A = \sqrt[5]{x(1 + x)^2(1 + 2x)^3 + \sqrt[3]{\dfrac{x(1 + x)^2(1 + 2x)^3}{ln|ctg\,y|}}}$

9. $S = arctg\sqrt{\left|\dfrac{x - sin\,y}{x + sin\,y} + \dfrac{x + sin\,y}{x - sin\,y}\right|} + e^{(x - sin\,y)(x + sin\,y)}$

10. $B = \dfrac{1 + arcsin(cos\,2y\,)}{2^x + 3^{-x}} + \left(\dfrac{2^x + 3^{-x} - 1}{x + arcsin(cos\,2y\,)}\right)^2$

11. $M = \dfrac{\sqrt[3]{ln\,x + y^2}}{0.47x^2} - \left|0.47x^2 - \dfrac{10^4}{7}cos^2\,y\right| - \dfrac{1.52}{x}$

12. $L = \dfrac{\left|x^2 - y^2\right|}{sin\,2x} + \sqrt[5]{|sin\,2x - 2.42y|} - \dfrac{x^2 + tg\,3y}{e^{2x}}.$

13. $C = sin\dfrac{-1.3 - x}{0.75y} + \sqrt[3]{\dfrac{y - 8x^2}{2x}} + \dfrac{cos\,8x^2}{tg\,3} - \dfrac{0.91y}{-1.3x}$

14. $Z = -\dfrac{(\,x - 1.25\,)(\,x^2 + y^2\,)}{\sqrt[3]{x^2 + y^2 - 2.2xy}} + 10^{-3}tg\,2y - \dfrac{cos\,2x}{sin\,5}.$

15. $N = \dfrac{\sqrt{|x - y| + (\,x + y\,)^2}}{sin\,2x} + 10^{-3}e^{-1.25x} - \dfrac{|y - x| + y^2}{\sqrt[3]{(\,0.31 + x\,)^2}}.$

2) Даны длины сторон треугольника. Найдите значение площади треугольника и выведите его в виде рациональной дроби, и приближенное значение, выведенное с 14 знаками после запятой.

Варианты:

1) 52, 45, 37
2) 34,62, 45
3) 76, 32, 56
4) 57, 42,79
5) 39, 67, 52
6) 56, 44, 81
7) 96, 69, 88
8) 54, 39, 76
9) 56, 90, 71
10) 45, 67, 74
11) 93, 66, 81
12) 59, 48, 73
13) 64, 59, 81
14) 55, 88, 66
15) 59, 41, 50

3) Решить текстовые задачи.

Варианты:

1. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.

2. Известны координаты двух точек A, B на плоскости. Найти координаты и длину вектора $\overrightarrow{AB}$.

3. Дан радиус окружности. Найти длину окружности и площадь круга.

4. Даны два целых числа. Найти их среднее геометрическое и среднее арифметическое.

5. Даны катеты прямоугольного треугольника. Найти его медиану и площадь.

6. В трапецию можно вписать окружность. Дан периметр трапеции. Найти среднюю линию трапеции.

7. Даны длины сторон прямоугольника. Найти его периметр и длину диагонали.

8. Даны катеты прямоугольного треугольника. Найти его гипотенузу и высоту, опущенную на гипотенузу.

9. Даны длины сторон прямоугольного параллелепипеда. Найти площадь его боковой поверхности и объем.

10. Известны координаты двух точек на плоскости. Найти расстояние между этими точками.

11. В равнобедренной трапеции угол при основании равен 30^0. Даны боковая сторона и верхнее основание трапеции. Найти ее площадь.

12. Даны две концентрические окружности. Известны их радиусы. Найти площадь кольца, образованного этими окружностями.

13. Дан параллелограмм с тупым углом в 150^0. Известны длины большей стороны параллелограмма и его высота, проведенная к большей стороне. Найти его периметр и площадь.

14. Известны координаты двух точек A и B в пространстве. Найти координаты точки M, делящей отрезок в отношении 2:3, считая от точки A.

15. Известны координаты вершин треугольника ABC на плоскости. Точка M - середина медианы BH. Найти длину CM.

1.1.3 Контрольные вопросы

1) Как называется окно, в котором реализуются прямые вычисления в пакете MatLab?

2) В какой переменной MatLab всегда сохраняет результат, если он не был присвоен никакой другой переменной?

3) Приведите примеры основных математических функций системы MatLab.

4) Какой формат представления результатов вычислений используется в MatLab по умолчанию? Как изменить на экране формат вывода числа?

5) Как можно просмотреть в MatLab список всех элементарных математических функций?

6) Какой символ используется для переноса длинных формул на другую строку?

7) Для чего служат клавиши <↓ > и <↑> в MatLab?

8) Что произойдет если завершить строку с оператором присваивания точкой с запятой?

9) Что может содержать имя переменной?

10) Как называется окно системы MatLab, предназначенное для ввода и просмотра результатов вычислений?

2 Операции с векторами и матрицами в Matlab

2.1 Лабораторная работа №2. Вектор-строки и вектор-столбцы

2.1.1 Теоретические сведения

Массивы являются одним из самых распространенных способов хранения данных и используются во всех языках программирования и вычислительных пакетах. По умолчанию все числовые переменные в Matlab считаются матрицами, так что скалярная величина есть матрица первого порядка, а векторы являются матрицами, состоящими из одного столбца или одной строки.

Для ввода вектора используются квадратные скобки, элементы вектора отделяются друг от друга:

- точкой с запятой, если требуется получить вектор–столбец;
- пробелом или запятой, если необходимо разместить элементы в вектор–строке.

```
>> a = [0.2;  -3.9;  4.6]
a =
0.2
-3.9
4.6
>> b = [7.6;  0.1;  2.5];
>> u = [0.1  0.5  -3.7  8.1]
u =
0.1  0.5  -3.7  8.1
>> v = [5.2  9.7  3.4  -0.2];
```

Точка с запятой в конце строки для подавления вывода на экран никак не связана с точкой с запятой, которая является разделителем элементов в вектор-столбцах.

Все числа в Matlab хранятся в двумерных массивах, каждый из размеров которых равен единице. Векторы также представляются двумерными массивами, один из размеров которых равен единице.

Векторы могут быть сформированы как диапазоны – при помощи двоеточий, разделяющих стартовое значение, шаг и конечное значение. Если величина шага отсутствует, то по умолчанию его значение равно единице.

```
>> c = 1:5
c =
    1    2    3    4    5
>> d = 1:2:5
d =
    1    3    5
```

Шаг может быть отрицательным, в этом случае начальное значение должно быть больше, либо равно конечному для получения непустого вектора.

Для получения длины вектора предназначена функция *length*, вектор указывается в качестве ее входного аргумента:

```
>> L = length (a)
L =
3
```

Вектор-столбцы с одинаковым числом элементов можно складывать и вычитать друг из друга при помощи знаков "+" и "–". Аналогичное верно и для вектор-строк:

```
>> c = a + b;
>> w = u - v;
```

Сложение и вычитание вектор-строки и вектор-столбца или векторов разных размеров приводит к ошибке. Операция * предназначена для умножения векторов по правилу матричного умножения.

Скалярное произведение двух векторов возвращает функция *dot*, а векторное – *cross*:

```
>> s = dot (a, b);
>> c = cross (a, b);
```

Разумеется, векторное произведение определено только для векторов из трех элементов.

Для операции транспонирования зарезервирован апостроф < ' >.

Matlab поддерживает поэлементные операции с векторами. Наряду с умножением по правилу матричного умножения, существует операция поэлементного умножения .* (точка со звездочкой). Данная операция применяется к векторам одинаковой длины и приводит к вектору той же длины, что исходные, элементы которого равны произведениям соответствующих элементов исходных векторов. Например, для векторов *a* и *b*, введенных выше, поэлементное умножение дает следующий результат:

```
>> e = a.*b
e =
1.5200
-0.3900
11.5000
```

Аналогичным образом работает поэлементное деление ./ (точка с косой чертой). Кроме того, операция .\ (точка с обратной косой чертой) осуществляет обратное поэлементное деление. Возведение элементов вектора a в степени, равные соответствующим элементам *b*, производится с использованием .^ (точка с крышечкой).

Векторы могут быть аргументами встроенных математических функций, таких, как *sin*, *cos* и т. д. В результате получается вектор с элементами, равными значению вызываемой функции от соответствующих элементов исходного вектора.

```
>> q=sin ([0  pi/2  pi])
q =
0   1.0000   0.0000
```

Однако для вычисления более сложной функции от вектора значений, требует использования поэлементных операций:

```
>> x = -1.2:0.5:1.8
>> x
x =
-1.2000 -0.7000 -0.2000 0.3000 0.8000 1.3000 1.8000
>> f = (x.*sin (x)+x.^2)./(x+1)
```

f =

-12.7922 3.1365 0.0997 0.1374 0.6744 1.2794 1.7832

Matlab обладает большим набором встроенных функций для обработки векторных данных, часть из них приведена в таблице 4. Полный список имеющихся функций выводится в командное окно при помощи *help datafun*, а для получения подробной информацию о каждой функции требуется указать ее имя в качестве аргумента команды *help*.

Таблица 4 – Функции обработки данных

Функции	Назначение
s=sum(a)	Сумма всех элементов вектора a
p=prod(a)	Произведение всех элементов вектора a
m=max(a)	Нахождение максимального значения среди элементов вектора a
[m,k]=max(a)	Второй выходной аргумент k содержит номер максимального элемента в векторе a
m=min(a)	Нахождение минимального значения среди элементов вектора a
[m,k]=min(a)	Второй выходной аргумент k содержит номер минимального элемента в векторе a
m=mean(a)	Вычисление среднего арифметического элементов вектора a
a1=sort(a)	Упорядочение элементов вектора по возрастанию
[a1,ind]=sort(a)	Второй выходной аргумент ind является вектором из целых чисел от 1 до length(a), который соответствует проделанным перестановкам

Очень часто требуется обработать только часть вектора, или обратиться к некоторым его элементам. Для доступа к элементу вектора необходимо указать его номер в круглых скобках сразу после имени переменной, в которой содержится вектор. По умолчанию нумерация элементов в векторе начинается с единицы.

>> d (2)

d =

3

>> s=v(1)+v(3)

s =

8.6

Обращение к последнему элементу вектора можно произвести с использованием *end*, т.е. *v(end)* и *v(length(v))* приводят к одинаковым результатам.

Указание номеров элементов вектора можно использовать и при вводе векторов, последовательно добавляя новые элементы (не обязательно в порядке возрастания их номеров).

```
>> h=10;
>> h(2)=20;
>> h(4)=40;
>> h
h =
10   20   0   40
```

Заметьте, что для ввода первого элемента *h* не обязательно указывать его индекс, т.к. при выполнении оператора *h=1* создается вектор (массив размера один на один). Следующие операторы присваивания приводят к автоматическому увеличению длины вектора *h*, а пропущенные элементы (в нашем случае *h*(3)) получают значение ноль.

Индексация двоеточием позволяет выделить идущие подряд элементы в новый вектор. Начальный и конечный номера указываются в круглых скобках через двоеточие, например:

```
>> z = [0.2  -3.8  7.9  4.5  7.2  -8.1  3.4];
>> znew = z(3:6)
znew =
7.9000  4.5000  7.2000  -8.1000
```

Индексный вектор должен содержать номера требуемых элементов, например:

```
>> ind = [3 5 7];
>> znew = z (ind)
znew =
7.9000 7.2000 3.4000
```

Конструирование новых векторов из элементов имеющихся векторов производится при помощи квадратных скобок. Следующий оператор приводит к образованию вектора, в котором пропущен пятый элемент вектора *z:*

```
>> znew = [z (1:4) z (6:end)]
znew =
0.2000 -3.8000 7.9000 4.5000 -8.1000 3.4000
```

Для удаления элемента вектора достаточно присвоить ему пустой массив – пару квадратных скобок []:

```
>> c (3) = []
c =
   1   2   4   5
```

2.1.2 Практическая часть

Контрольные задания

1) Для заданных векторов *a* и *b* длины *n*:

a) вычислить их сумму, разность и скалярное произведение;

b) образовать вектор $c = [a_1, a_2, ..., a_n, b_1, b_2, ..., b_n]$, определить его максимальный и минимальный элементы и поменять их местами;

c) упорядочить вектор *c* по возрастанию и убыванию;

d) переставить элементы вектора *c* в обратном порядке и записать результат в новый вектор;

e) найти векторное произведение $u = [a_1, a_3, a_4]$ и $v = [b_2, b_3, b_4]$.

Варианты:

1. a = [0.5 3.7 6.0 -4.3 1.2 -2.7 2.4 2.2]; b = [3.6 7.0 7.0 5.4 2.6 -2.7 -6.4 0.3].
2. a = [-4.8 -1.3 -1.0 0.7 4.0 5.8 4.3 -8.0]; b = [-1.1 -1.9 7.1 -2.1 6.8 2.8 0.3 1.6].
3. a = [1.0 -3.9 -2.3 -3.3 -1.7 2.2 -0.6 1.8]; b = [2.7 -2.7 -2.2 4.4 0.4 -6.0 -3.4 -5.2].
4. a = [-2.4 3.3 -0.1 3.6 7.4 -2.8 0.3 2.2]; b = [6.3 0.6 4.3 -3.7 -7.0 3.7 3.7 8.0].
5. a = [8.4 -5.9 -6.5 -0.9 6.9 -1.7 1.7 0.8]; b = [0.0 2.0 -1.5 7.5 -4.0 -3.0 -6.2 0.0].
6. a = [5.3 6.8 -7.1 6.8 -4.0 -2.3 -4.4 -0.2]; b = [7.5 -1.5 -4.9 -4.6 -2.3 -5.3 5.5 2.3].

7. a = [1.2 -4.1 -0.8 -0.7 -2.2 1.7 3.3 -6.1]; b = [-1.5 2.2 1.0 -4.3 -0.0 -1.8 -1.5 2.4].

8. a = [6.6 -5.0 -2.7 8.3 3.8 1.9 1.1 2.7]; b = [-1.0 3.2 4.2 -6.4 1.9 -6.5 -6.2 -8.1].

9. a = [-1.9 0.4 1.8 4.2 -3.8 -4.7 4.0 -2.1]; b = [-8.7 -4.2 -1.4 2.8 -2.2 7.8 0.0 -0.1].

10. a = [0.9 1.7 -3.2 -3.8 7.3 6.0 -0.2 8.6]; b = [0.6 -0.4 -6.9 -2.2 1.6 3.8 -3.2 0.4].

11. a = [1.9 2.7 -3.8 -1.8 10.3 -6.0 0.4 8.6]; b = [-3.9 -2.3 -3.3 2.2 2.0 -1.5 7.5 -4.0].

12. a = [0.9 -3.2 7.3 -0.2 -1.3 0.7 5.8 -8]; b = [4.0 5.8 4.3 -8.0 -1.1 -1.9 7.1 -2.1].

13. a = [7.5 -1.5 -4.9 -3.8 7.3 1.3 -3.3 8.6]; b = [-2.4 3.3 -0.1 3.6 -0.0 -1.8 -1.5 2.4].

14. a = [1.1 -2.5 0.5 -0.1 -3.5 4.6 -9.2 1.2]; b = [2.6 -2.7 6.4 0.3 0.6 -0.4 -6.9 -2.2].

15. a = [-6.5 0.9 6.9 -1.7 -2.2 4.4 0.4 -6.0]; b = [-2.3 -4.4 -0.2 -6.4 1.9 -6.5 -6.2 -8.1].

2) Вычислить значения функции на отрезке в заданном числе N равномерно отстоящих друг от друга точек.

Варианты:

1. $y(x)=\dfrac{\sin x\cos x}{x^2+1}$ $\quad [0,2\pi]$, $N = 10$

2. $y(x)=\ln(x+1)\sqrt{e^x+e^{-x}}$ $\quad [-0.2,4]$, $N = 8$

3. $y(x)=x^2 tg\sqrt{\arcsin x}$ $\quad \left[0,\dfrac{1}{3}\right]$, $N = 9$

4. $y(x)=x\sin x+x^3\dfrac{e^x}{x+1}$ $\quad [0,1]$, $N = 7$

5. $y(x)=\dfrac{1}{1+\dfrac{x}{\sqrt{1+x}}}$ $\quad [0,3]$, $N = 9$

6. $y(x)=\dfrac{e^{\sin x}+e^{\cos x}}{x^2}$ $\quad [\pi,3\pi]$, $N = 11$

7. $y(x)=ctg\left(x^2+1\right)\cdot\left(\sin 2x+\cos 2x\right)$ $\quad [-1,1]$, $N = 7$

8. $y(x)=\log_2\left(x^2+1\right)\cdot\sin\dfrac{1}{x^2+1}$ $\quad [-1,1]$, $N = 10$

9. $y(x)=\left|x^3+2x^2-3\right|\sin\pi x$ $\quad [-2,2]$, $N = 7$

10. $y(x)=\dfrac{\sqrt[3]{x+1}}{\sqrt{|x|+\dfrac{1}{2}}}\cdot\dfrac{\sin x+1}{\cos x+2}$ $[-2,2]$, $N = 9$

11. $y(x)=-\dfrac{(x-1)(x^2+0{,}75^2)}{\sqrt[3]{x^3+x^2-2}}$ $[-0.4,2]$, $N = 8$

12. $y(x)=\dfrac{2x}{0.5^2}-\left|\dfrac{x-5}{\sin x}\right|$ $[-1,3]$, $N = 10$

13. $y(x)=\cos\sqrt{(x^2+0.04)}+\dfrac{\sqrt{x^2+e^x}}{0.4x}$ $[-2,4]$, $N = 11$

14. $y(x)=10^{-3}tg\pi x-\dfrac{\cos 2x}{\sin 5}.$ $[0,2\pi]$, $N = 9$

15. $y(x)=\dfrac{\ln 3}{\sqrt[3]{3.5x^2+1}}-e^{-2x}$ $\left[0,\dfrac{4}{3}\right]$, $N = 7$

2.1.3 Контрольные вопросы

1) Как осуществляется ввод вектор–строки и вектор–столбца?

2) Как обратиться к элементу вектора? Как вывести последний элемент вектора?

3) Как найти длину вектора?

4) Как вывести номер максимального элемента?

5) Чем отличаются операции: a*b и a.*b?

6) Какая функция вычисляет векторное произведение двух векторов? Для каких векторов можно найти векторное произведение?

7) Что произойдет с вектором после выполнения операции: a(2) = []?

8) Что делает команда *sort*?

9) С помощью какой функции можно найти произведение всех элементов вектора?

10) С помощью какой команды можно переставить элементы вектора в обратном порядке?

2.2 Лабораторная работа №3. Матрицы

2.2.1 Теоретические сведения

Матрицы небольших размеров удобно вводить из командной строки. Существует три способа ввода матриц:

1) набрать в командной строке [(открывающая квадратная скобка), далее элементы каждой строки матрицы набираются через пробел, ввод каждой строки завершается нажатием на <Enter>. При вводе последней строки в конце ставится] (закрывающая квадратная скобка).

```
>> A = [0.7 -2.5 9.1
8.4 0.3 1.7
-3.5 6.2 4.7]
A =
0.7   -2.5   9.1
8.4    0.3   1.7
-3.5  6.2    4.7
```

2) Матрицу можно рассматривать как вектор-столбец, каждый элемент которого является строкой матрицы.

```
>> B = [6.1 0.3; -7.9 4.4; 2.5 -8.1]
B =
6.1   0.3
-7.9  4.4
2.5   -8.1
```

3) Матрицу можно считать вектор-строкой, каждый элемент которой является столбцом матрицы.

```
>> C = [[0.4; 0.1] [-7.2; -2.1] [5.3; -9.5]]
C =
0.4   -7.2   5.3
0.1   -2.1   -9.5
```

Обратите внимание, что внутренние квадратные скобки действительно нужны. Оператор C = [0.4; 0.1 -7.2; -2.1 5.3; -9.5] является недопустимым и приводит к сообщению об ошибке, поскольку оказывается, что в первой строке матрицы содержится только один элемент, во второй и третьей – по два, а в четвертой – снова один.

Функция *size* позволяет установить размеры массивов, она возвращает результат в виде вектора, первый элемент которого равен числу строк, а второй – столбцов:

```
>> s = size(B)
s =
3 2
```

Сложение и вычитание матриц одинаковых размеров производится с использованием знаков +, -. Звездочка * служит для вычисления матричного произведения, причем соответствующие размеры матриц должны совпадать, например:

```
>> P = A*B
P =
46.7700 -84.5000
53.1200 -9.9300
-58.5800 -11.8400
```

Допустимо умножение матрицы на число и числа на матрицу, при этом происходит умножение каждого элемента матрицы на число и результатом является матрица тех же размеров, что и исходная. Апостроф < ' > предназначен для транспонирования вещественной матрицы или нахождения сопряженной к комплексной матрице. Для возведения квадратной матрицы в степень применяется знак < ^ > (крышечка).

Вычислите для тренировки матричное выражение $R = (A - BC)^3 + ABC$, в котором *A*, *B* и *C* – определенные выше матрицы. Ниже приведена запись в Matlab этого выражения:

```
>> R = (A-B*C)^3+A*B*C
R =
1.0e+006 *
-0.0454 0.1661 -0.6579
0.0812 -0.2770 1.2906
-0.0426 0.1274 -0.7871
```

Matlab обладает многообразием различных функций и способов для работы с матричными данными. Для обращения к элементу двумерного массива следует указать его строчный и столбцевой индексы в круглых скобках после имени массива, например:

```
>> C(1,2)
ans =
-7.2000
```

Индексация двоеточием позволяет получить часть матрицы – строку, столбец или блок, например:

```
>> c1 = A(2:3,2)
c1 =
0.3000
6.2000
>> r1 = A(1,1:3)
r1 =
0.7000 -2.5000 9.1000
```

Для обращения ко всей строке или всему столбцу не обязательно указывать через двоеточие начальный (первый) и конечный индексы, то есть операторы r1=A(1,1:3) и r1=A(1,:) эквивалентны. Для доступа к элементам строки или столбца от заданного до последнего можно использовать *end*, так же как и для векторов: A(1,2:end). Выделение блока, состоящего из нескольких строк и столбцов, требует индексации двоеточием, как по первому измерению, так и по второму. Пусть в массиве T хранится матрица:

$$T = \begin{bmatrix} 1 & 7 & -3 & 2 & 4 & 9 \\ 0 & -5 & -6 & 3 & -8 & 7 \\ 2 & 4 & 5 & -1 & 0 & 3 \\ -6 & -4 & 7 & 2 & 6 & 1 \end{bmatrix}.$$

Для выделения ее элементов на пересечении второй и третьей строк, и второго по четвертый столбца, достаточно использовать оператор:

```
>> T1 = T(2:3,2:4)
T1 =
-5  -6   3
4    5  -1
```

Индексация двоеточием так же очень полезна при различных перестановках в массивах. В частности, для перестановки первой и последней строк в произвольной матрице, хранящейся в массиве *A*, подойдет последовательность команд:

```
>> s = A(1,:);
>> A(1,:) = A(end,:);
>> A(end,:) = s;
```

Matlab поддерживает такую операцию, как вычеркивание строк или столбцов из матрицы. Достаточно удаляемому блоку присвоить пустой массив, задаваемый квадратными скобками. Например, вычеркивание второй и третьей строки из массива *T*, введенного выше, производится следующей командой:

```
>> T(2:3,:) = []
T =
1    7  -3  2  4  9
-6  -4   7  2  6  1
```

Индексация двоеточием упрощает заполнение матриц, имеющих определенную структуру. Предположим, что требуется создать матрицу:

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Первый шаг состоит в определении нулевой матрицы размера пять на пять, затем заполняются первая и последняя строки и первый и последний столбцы:

```
>> W(1:5,1:5) = 0;
>> W(1,:) = 1;
>> W(end,:) = 1;
>> W(:,1) = 1;
>> W(:,end) = 1;
```

Ряд встроенных функций, приведенных в таблице 5, позволяет ввести стандартные матрицы заданных размеров. Обратите внимание, что во всех функциях, кроме *diag*, допустимо указывать размеры матрицы следующими способами:

- числами через запятую (в двух входных аргументах);
- одним числом, результат – квадратная матрица.

Если, к примеру, *A* была определена ранее, то команда I = eye(size (A)) приводит к появлению единичной матрицы, размеры которой совпадают с размерами *A*, так как функция *size* возвращает размеры матрицы в векторе.

Разберем, как получить трехдиагональную матрицу *T* размера семь на семь, приведенную ниже, с использованием функций Matlab:

$$T = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 5 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 3 & -3 & 0 & 0 & 0 \\ 0 & 0 & 5 & 4 & -4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 5 & -5 & 0 \\ 0 & 0 & 0 & 0 & 5 & 6 & -6 \\ 0 & 0 & 0 & 0 & 0 & 5 & 7 \end{bmatrix}.$$

Введите вектор *v* с целыми числами от одного до семи и используйте его для создания диагональной матрицы и матрицы со смещенной на единицу вверх диагональю. Вектор длины шесть, содержащий пятерки, заполняется, например, так: 5*ones(1,6). Этот вектор укажите в первом аргументе функции *diag*, а минус единицу – во втором и получите третью вспомогательную матрицу. Теперь достаточно вычесть из первой матрицы вторую и сложить с третьей:

```
>> T = diag(v)-diag(v(1:6),1)+diag(5*ones(1,6),-1)
```

Таблица 5 – Функции для создания стандартных матриц

Функция	Описание
zeros(n) zeros(m, n)	Нулевая матрица
eye (n) eye (m, n)	Единичная прямоугольная матрица (единицы расположены на главной диагонали)
ones(n) ones (m, n)	Матрица, целиком состоящая из единиц
rand(n)	Матрица, элементы которой — случайные числа, равномерно распределенные на интервале (0,1)
randn(n)	Матрица, элементы которой — случайные числа, распределенные по нормальному закону с нулевым средним и дисперсией равной единице
diag(v)	Диагональная матрица, элементы которой задаются во входном аргументе — векторе
diag(v,k)	Диагональная матрица со смещенной на k позиций диагональю (положительные k — смещение вверх, отрицательные — вниз)
d=diag(A)	Выделение главной диагонали из матрицы в вектор
d=diag(A,k)	Выделение k-ой диагонали из матрицы в вектор

В предыдущем разделе было описано применение поэлементных операций к векторам. Поэлементные вычисления с матрицами производятся практически аналогично, разумеется, необходимо следить за совпадением размеров матриц:

– *A.*B, A./B* – поэлементные умножение и деление;

– *A.^p* – поэлементное возведение в степень, *p* – число;

– *A.^B* – возведение элементов матрицы A в степени, равные соответствующим элементам матрицы B;

– *A.'* – транспонирование матрицы (для вещественных матриц A' и A.' приводят к одинаковым результатам);

Иногда требуется не просто транспонировать матрицу, но и "развернуть" ее. Разворот матрицы на 90^o против часовой стрелки осуществляет функция *rot90*:

```
>> Q = [1 2; 3 4]
Q =
1 2
3 4
>> R = rot90(Q)
R =
2 4
1 3
```

Допустимо записывать сумму и разность матрицы и числа, при этом сложение или вычитание применяется, соответственно, ко всем элементам матрицы. Вызов математической функции от матрицы приводит к матрице того же размера, на соответствующих позициях которой стоят значения функции от элементов исходной матрицы.

В Matlab определены и матричные функции, например, *sqrtm* предназначена для вычисления квадратного корня. Например, найдем квадратный корень из матрицы $K = \begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}$ и проверим полученный результат, возведя его в квадрат (по правилу матричного умножения, а не поэлементно!):

```
>> K = [3 2; 1 4];
>> S = sqrtm(K)
S =
1.6882 0.5479
0.2740 1.9621
>> S*S
ans =
3.0000 2.0000
```

1.0000 4.0000

Матричная экспонента вычисляется с использованием *exp*.

Все функции обработки данных, приведенные в таблице 4, могут быть применены и к двумерным массивам. Основное отличие от обработки векторных данных состоит в том, что эти функции работают с двумерными массивами по столбцам, например, функция *sum* суммирует элементы каждого из столбцов и возвращает вектор-строку, длина которой равна числу столбцов исходной матрицы:

```
>> M = [1 2 3; 4 5 6; 7 8 9]
M =
1 2 3
4 5 6
7 8 9
>> s = sum(M)
s =
12 15 18
```

Если в качестве второго входного аргумента *sum* указать 2, то суммирование произойдет по строкам. Для вычисления суммы всех элементов матрицы требуется дважды применить *sum*:

```
>> s = sum (sum (M))
s =
45
```

Очень удобной возможностью Matlab является конструирование матрицы из матриц меньших размеров. Пусть заданы матрицы:

$$M1=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}, M2=\begin{bmatrix}-2 & -3 & -5\\ -1 & -5 & -6\end{bmatrix}, M3=\begin{bmatrix}9 & 8\\ -7 & -5\\ 1 & 2\end{bmatrix}, M4=\begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}.$$

Требуется составить из *M1, M2, M3* и *M4* блочную матрицу $M=\left[\begin{array}{c|c} M1 & M2 \\ \hline M3 & M4\end{array}\right]$.

Можно считать, что *M* имеет размеры два на два, а каждый элемент является, соответственно, матрицей *M1, M2, M3* или *M4*. Следовательно, для получения матрицы *M* можно использовать оператор:

```
>> M=[M1 M2; M3 M4];
```

С помощью пустого массива можно удалять из массива строки и столбцы. Например, в матрице

```
c =
16     2     3    13   160    20    30   130
 5    11    10     8    50   110   100    80
 9     7     6    12    90    70    60   120
 4    14    15     1    40   140   150    10
```

удалим сначала 2 столбец, а затем 3 строку:

```
>> c(:,2) = []
c =
16     3    13   160    20    30   130
 5    10     8    50   110   100    80
 9     6    12    90    70    60   120
 4    15     1    40   140   150    10
>> c(3,:) = []
c =
16     3    13   160    20    30   130
 5    10     8    50   110   100    80
 4    15     1    40   140   150    10
```

2.2.2 Практическая часть

Контрольные задания

1) Введите матрицы:

$$A = \begin{bmatrix} -9.8 & 4.4 & 1.3 \\ -5.7 & 0.1 & 0.8 \\ 2.4 & 4.4 & 8.6 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 2 \\ 3 & 0 & -1 \\ 5 & 2 & 2 \\ 8 & 9 & 0 \end{bmatrix}, C = \begin{bmatrix} 0.1 & 0.2 & -1.3 & 0.7 \\ -0.2 & 0.3 & 2.2 & 0.8 \\ 1.9 & 2.3 & 6.5 & 4.9 \end{bmatrix}$$

и найдите значения следующих выражений.

Варианты:

1. $(A^3 + CB)(A^2 - 3CB)^T$
2. $A^4 + 2A^3 - ACB$
3. $BAC - 4C^T B^T$
4. $3BA^3C - BAC + 2BC$
5. $-3C^T AC - BB^T$
6. $(BCB - 4C^T)A^4$
7. $(AB^T - C)(C + AB^T)^T - 3A$
8. $(AB^T B)^4 - 2A^3 + CC^T$
9. $C(BB^T + C^T C)C^T - 8A$
10. $2AA^T - (CB)^2 + 4A$
11. $A^2 - 3A^5 + ACB$
12. $2BA^4C + 3BAC - 1.2BC$
13. $2(6BCB + C^T)A^3$
14. $(A^3 - 3CB)(A^4 + 2CB)^T$
15. $(3AB^T + 1.1C)(5C - AB^T)^T + 3A$

2) При помощи встроенных функций для заполнения стандартных матриц, индексации двоеточием и, возможно, поворота, транспонирования или вычеркивания получите следующие матрицы.

Варианты:

1. $$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 & 0 & 7 & 1 \\ 1 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

5. $$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 1 \\ 2 & 0 & 0 & 0 & 7 & 0 & 1 \\ 3 & 0 & 0 & 7 & 0 & 7 & 1 \\ 4 & 0 & 7 & 0 & 7 & 0 & 1 \\ 5 & 7 & 0 & 7 & 0 & 0 & 1 \\ 6 & 0 & 7 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 1 \end{bmatrix}$$

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

6. $$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

7. $$\begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

8. $$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4. $$\begin{bmatrix} 4 & 3 & 3 & 3 & 3 & 3 & 9 \\ 3 & 4 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 4 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 4 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 4 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 & 4 & 3 \\ 9 & 3 & 3 & 3 & 3 & 3 & 4 \end{bmatrix}$$

9. $$\begin{bmatrix} 1 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 0 & 0 & 5 & 0 & 2 \\ 1 & 0 & 0 & 5 & 0 & 5 & 3 \\ 1 & 0 & 5 & 0 & 5 & 0 & 4 \\ 1 & 5 & 0 & 5 & 0 & 0 & 5 \\ 1 & 0 & 5 & 0 & 0 & 0 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 1 \end{bmatrix}$$

10. $$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 3 & 0 & 5 & 0 & 1 \\ 1 & 0 & 0 & 3 & 0 & 0 & 1 \\ 1 & 0 & 5 & 0 & 3 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 3 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

11. $$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

12. $$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

13. $$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

14. $$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 2 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 2 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

15. $$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

3) Вычислить значения функции для всех элементов матрицы и записать результат в матрицу того же размера, что и исходная.

Варианты:

1. $f(x) = x^3 - 2x^2 + \sin x - 4,\ A = \begin{bmatrix} 9.33 & -4.01 & 8.19 & 2.64 \\ 0.55 & 3.81 & 3.32 & 5.07 \end{bmatrix}$

2. $f(x) = \dfrac{e^x - x}{e^x + x},\ A = \begin{bmatrix} 9.32 & 0.21 & -9.89 & 3.11 \\ 0.54 & 4.99 & 5.01 & -0.03 \end{bmatrix}$

3. $f(x) = \sqrt{1 + \sqrt{|x|^3 + 1}},\ A = \begin{bmatrix} -1.54 & 0.49 & 3.11 & 2.99 \\ 4.05 & -5.85 & 3.72 & 0.11 \end{bmatrix}$

4. $f(x) = e^x \cos x - e^{-x} \sin x,\ A = \begin{bmatrix} -9.04 & 3.36 & 3.09 & -2.49 \\ -4.33 & -5.09 & 9.74 & 1.65 \end{bmatrix}$

5. $f(x) = \ln(|x|) \sin \pi x,\ A = \begin{bmatrix} 0.33 & 0.95 & 7.12 & -9.22 \\ -0.64 & 3.76 & 1.74 & -0.03 \end{bmatrix}$

6. $f(x) = e^{x^2 + x + 1},\ A = \begin{bmatrix} -4.53 & -2.12 & -6.54 & -3.22 \\ 3.4 & 7.46 & -1.74 & -0.03 \end{bmatrix}$

7. $f(x) = \dfrac{\sqrt[3]{x^2 - 1}}{|x| + 3},\ A = \begin{bmatrix} 0.23 & 3.89 & 7.12 & -7.22 \\ 5.84 & 5.13 & -0.89 & 3.55 \end{bmatrix}$

8. $f(x) = \dfrac{1}{1 + \dfrac{1 + x}{1 - x^2}},\ A = \begin{bmatrix} -5.84 & 9.84 & 0.23 & 1.59 \\ -9.25 & -0.25 & 1.54 & 0.43 \end{bmatrix}$

9. $f(x) = \dfrac{x^3 + \sin x}{x^3 - \cos x} \sqrt{e^x + 1},\ A = \begin{bmatrix} 0.64 & 634 & 0.32 & -4.23 \\ 1.19 & 3.13 & 1.54 & 0.43 \end{bmatrix}$

10. $f(x) = \arcsin(\cos x^2),\ A = \begin{bmatrix} \pi & 2.2\pi & -2\pi & 0.3\pi \\ 3\pi & -\pi & 0.1\pi & 5\pi \end{bmatrix}$

11. $f(x) = 2e^x \ln(|x|) \cos \pi x,\ A = \begin{bmatrix} 0.63 & -0.95 & 7.12 & -9.27 \\ -3.64 & -5.76 & 1.74 & 2.03 \end{bmatrix}$

12. $f(x) = \dfrac{2x^3 + \cos x}{x^3 - 2\sin x} \sqrt{5e^x - 1},\ A = \begin{bmatrix} -0.64 & 634 & 8.36 & -4.23 \\ 1.17 & -6.13 & 1.54 & 0.23 \end{bmatrix}$

13. $f(x) = 5x^4 + 2x^3 + \cos x - 7,\ A = \begin{bmatrix} -9.33 & 4.01 & 8.19 & 2.64 \\ 0.55 & 3.81 & -3.32 & 5.07 \end{bmatrix}$

14. $f(x)=\dfrac{\sqrt[4]{x^2-3}}{|x|-7}$, $A=\begin{bmatrix}-9.12 & 3.89 & -8.12 & -7.22\\ 3.45 & -5.13 & 0.89 & 5.51\end{bmatrix}$

15. $f(x)=e^x \sin x - e^{-x}\cos x$, $A=\begin{bmatrix}9.04 & -3.36 & 4.09 & -2.47\\ 4.33 & -5.09 & -8.74 & 1.68\end{bmatrix}$

4) Сконструировать блочные матрицы (используя функции для заполнения стандартных матриц) и применить функции обработки данных и поэлементные операции для нахождения заданных величин.

Варианты:

1. $A=\begin{bmatrix}1&0&0&4&0&0\\0&1&0&0&4&0\\0&0&1&0&0&4\\2&2&2&3&3&3\\2&2&2&3&3&3\\2&2&2&3&3&3\end{bmatrix}$, $m=\max\limits_{j=1,\dots,6}\left\{\sum\limits_{i=1}^{6} a_{ij}^2\right\}$

2. $A=\begin{bmatrix}2&2&-1&-1&2&2\\2&2&-1&-1&2&2\\2&2&-1&-1&2&2\\2&2&-1&-1&2&2\\2&2&-1&-1&2&2\\2&2&-1&-1&2&2\end{bmatrix}$, $s=\sum\limits_{i=1}^{6}\sum\limits_{j=1}^{6}|a_{ij}|$

3. $A=\begin{bmatrix}1&2&3&4&5&6\\-1&-2&-3&-4&-5&-6\\4&4&4&4&4&4\\4&4&4&4&4&4\\5&5&5&5&5&5\\5&5&5&5&5&5\end{bmatrix}$, $m=\min\limits_{i,j=1,\dots,6} a_{ij}^3$

4. $A=\begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 \end{bmatrix}$, $s=\sum_{k=1}^{6} a_{kk}^{3}$

5. $A=\begin{bmatrix} 1 & 1 & -3 & -3 & -3 & -3 \\ 1 & 1 & -3 & -3 & -3 & -3 \\ -3 & -3 & 2 & 0 & 0 & 0 \\ -3 & -3 & 0 & 2 & 0 & 0 \\ -3 & -3 & 0 & 0 & 2 & 0 \\ -3 & -3 & 0 & 0 & 0 & 2 \end{bmatrix}$, $s=\sum_{i=1}^{5} a_{ii+1}$

6. $A=\begin{bmatrix} -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 4 & 4 \end{bmatrix}$, $s=\sum_{i=1}^{6} a_{ii} + \sum_{i=1}^{5} a_{ii+1}$

7. $A=\begin{bmatrix} 0 & 0 & 0 & 0 & 3 & 4 \\ 0 & 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 3 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 & 0 & 4 \\ 3 & 0 & 0 & 0 & 0 & 4 \\ 1 & 1 & 1 & 1 & 1 & -2 \end{bmatrix}$, $s=\sum_{i=1}^{6}\sum_{j=1}^{6} sin\left(\frac{\pi}{6} a_{ij}^{2}\right)$

8. $A=\begin{bmatrix} 1 & 2 & 3 & -1 & 0 & 0 \\ 1 & 2 & 3 & 0 & -1 & 0 \\ 1 & 2 & 3 & 0 & 0 & -1 \\ 0 & 0 & 7 & 2 & 2 & 2 \\ 0 & 7 & 0 & 2 & 2 & 2 \\ 7 & 0 & 0 & 2 & 2 & 2 \end{bmatrix}$, $m=\max_{i=1,\dots,6}\ \min_{j=1,\dots,6} a_{ij}$

9. $$A=\begin{bmatrix} 1 & 0 & 0 & 1 & -3 & -3 \\ 0 & 1 & 1 & 0 & -3 & -3 \\ 0 & 1 & 1 & 0 & -3 & -3 \\ 1 & 0 & 0 & 1 & -3 & -3 \\ -2 & -2 & -2 & -2 & 4 & 4 \\ -2 & -2 & -2 & -2 & 4 & 4 \end{bmatrix},\ s=\sum_{i=1}^{6}\max_{j=1,\dots,6}\left(a_{ij}+a_{ji}\right)$$

10. $$A=\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 \\ -1 & -1 & -1 & 4 & 0 & 0 \\ -1 & -1 & -1 & 0 & 4 & 0 \\ -1 & -1 & -1 & 0 & 0 & 4 \end{bmatrix},\ p=\prod_{i=1}^{6}\sum_{j=1}^{6}\left(a_{ij}\right)^{a_{ij}}$$

11. $$A=\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 4 & 4 \\ 0 & 0 & 0 & 0 & 4 & 4 \end{bmatrix},\ s=2\sum_{i=1}^{6}a_{ii}-\sum_{i=2}^{6}a_{ii-1}$$

12. $$A=\begin{bmatrix} 1 & 1 & -3 & -3 & -3 & -3 \\ 1 & 1 & -3 & -3 & -3 & -3 \\ -3 & -3 & 6 & 0 & 0 & 0 \\ -3 & -3 & 0 & 6 & 0 & 0 \\ -3 & -3 & 0 & 0 & 6 & 0 \\ -3 & -3 & 0 & 0 & 0 & 6 \end{bmatrix},\ s=\sum_{i=2}^{6}a_{ii-1}$$

13. $$A=\begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 2 & 0 & 4 \\ 0 & 0 & 2 & 0 & 0 & 4 \\ 0 & 2 & 0 & 0 & 0 & 4 \\ 2 & 0 & 0 & 0 & 0 & 4 \\ 1 & 1 & 1 & 1 & 1 & -3 \end{bmatrix},\ s=\sum_{i=1}^{6}\sum_{j=1}^{6}\cos\left(\frac{\pi}{3}a_{ij}^{3}\right)$$

14. $A=\begin{bmatrix} 1 & 0 & 0 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 & 4 \\ 2 & 2 & 2 & 6 & 6 & 6 \\ 2 & 2 & 2 & 6 & 6 & 6 \\ 2 & 2 & 2 & 6 & 6 & 6 \end{bmatrix}, m=\min\limits_{j=1,\dots,6}\left\{\sum\limits_{i=1}^{6} a_{ij}^3\right\}$

15. $A=\begin{bmatrix} 1 & 0 & 0 & 0 & -5 & -5 \\ 0 & 1 & 0 & 0 & -5 & -5 \\ 0 & 0 & 1 & 0 & -5 & -5 \\ 0 & 0 & 0 & 1 & -5 & -5 \\ -3 & -3 & -3 & -3 & 4 & 4 \\ -3 & -3 & -3 & -3 & 4 & 4 \end{bmatrix}, s=\sum\limits_{i=1}^{6}\min\limits_{j=1,\dots,6}\left(a_{ij}+a_{ji}\right)$

2.2.3 Контрольные вопросы

1) Для формирования многомерной единичной матрицы в Matlab используется функция…

2) Как выделить элементы главной диагонали из матрицы в вектор?

3) Для чего служат команды: *zeros, ones, rand, eye*?

4) В каком случае можно выполнить вертикальную конкатенацию матриц?

5) Как определяется число строк и столбцов матрицы?

6) Для чего предназначена функция *sqrtm*?

7) Какие операции служат для определения минимального и максимального элемента матрицы?

8) Что происходит при выполнении команды A'?

9) Как найти сумму элементов по строкам матрицы?

10) Что произойдет с матрицей при выполнении команды rot90(A)?

2.3 Лабораторная работа №4. Индексирование элементов матрицы

2.3.1 Теоретические сведения

Рассмотрим индексирование двумерной матрицы.

Элемент массива *a*, расположенный на пересечении строки *i* и столбца *j*, обозначается как *a*(*i*, *j*).

Рассмотрим в качестве примера матрицу *a*, заполненную с помощью функции *magic*, то есть образующую магический квадрат:

```
>> a = magic(4)
a =
16    2    3   13
5   11   10    8
9    7    6   12
4   14   15    1
```

Тогда a(4, 3) – это элемент, расположенный на пересечении 4 строки и 3 столбца, то есть

```
>> a(4,3)
ans =
15
```

Можно также обратиться к строке или столбцу целиком, или к части столбца или строки. Например, вывести элементы 2 столбца, расположенные на 1, 2 и 3 строках:

```
>> a(1:3,2)
ans =
2
11
7
>> q = a(1:3,2)
q =
2
```

11

7

или вывести элементы 3 строки:

```
>> a(3,:)
ans =
 9    7    6    12
```

В системе Matlab принято хранить каждый массив, независимо от его размерности как вектор-столбец, который образован объединением столбцов исходного массива. Например, массив

```
>> z = [2 6 9; 4 2 8; 3 0 1]
z =
2    6    9
4    2    8
3    0    1
```

хранится в виде столбца:

2

4

3

6

2

0

9

8

1

То есть к элементу z(2,3) данной матрицы можно обратиться z(8):

```
>> z(2,3)
ans =
8
>> z(8)
ans =
```

8

При удалении из матрицы одного элемента, нужно обращаться к нему, указывая один индекс, соответствующий данному элементу в соответствующем векторе-столбце. Например, удалим из матрицы *c*

c =

16	3	13	160	20	30	130
5	10	8	50	110	100	80
4	15	1	40	140	150	10

все элементы со 2 по 20 с шагом 3, при этом оставшиеся элементы преобразуются в вектор-строку:

```
>> c  (2:3:20) = []
c =
Columns 1 through 13
16    4    3   15   13    1  160   40   20  140   30  150  130
Column 14
10
```

2.3.2 Практическая часть

Контрольные задания

1) Создайте двумерную матрицу m, заполненную элементами магического квадрата. Создайте матрицу $m1$, объединяющую матрицу m и матрицу удвоенных элементов матрицы m. Удалите из полученной матрицы $m1$ вторую строку, третий столбец и все элементы с четными номерами с 5 по 25.

2) Решить текстовую задачу.

Варианты:

1. В данной действительной матрице размера n*m поменять местами строку, содержащую элемент с наибольшим значением, со строкой, содержащей элемент с наименьшим значением. Предполагается, что эти элементы единственны.

2. В данной действительной квадратной матрице порядка n найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.

3. Дана действительная матрица размера n*m, в которой не все элементы равны нулю. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.

4. Характеристикой столбца матрицы назовем произведение элементов с четными номерами строк. Определить номер столбца с наибольшей характеристикой.

5. Характеристикой строки целочисленной матрицы назовем сумму ее элементов с четными номерами столбцов. Определить номер строки с наименьшей характеристикой.

6. В данной действительной матрице размера n*m поменять местами столбец, содержащий элемент с наибольшим значением, со столбцом, содержащим элемент с наименьшим значением. Предполагается, что эти элементы единственны.

7. В данной действительной квадратной матрице порядка n найти произведение элементов столбца, в котором расположен элемент с наибольшим значением. Предполагается, что такой элемент единственный.

8. Дана действительная матрица размера n*m. Получить новую матрицу путем вычитания из каждого элемента данной матрицы ее наименьшего элемента.

9. Дана действительная матрица размера n*m. Переставьте все строки матрицы в обратном порядке.

10. Дана действительная матрица размера n*m. Переставьте все столбцы матрицы в обратном порядке.

11. В данной действительной квадратной матрице порядка n найти сумму элементов столбца, в котором расположен элемент с наибольшим значением. Предполагается, что такой элемент единственный.

12. Дана действительная матрица размера n*m. Получить новую матрицу путем умножения каждого элемента данной матрицы на ее наибольший элемент.

13. Дана действительная матрица размера n*m. Переставьте все строки матрицы в обратном порядке.

14. Характеристикой строки целочисленной матрицы назовем сумму ее элементов с нечетными номерами столбцов. Определить номер строки с наименьшей характеристикой.

15. Характеристикой строки целочисленной матрицы назовем сумму ее элементов с нечетными номерами столбцов. Определить номер строки с наименьшей характеристикой.

2.3.3 Контрольные вопросы

1) Как обратиться к элементу матрицы?
2) Как вывести все элементы 2 столбца матрицы?
3) С чего начинается нумерация элементов матрицы?
4) Что получится в результате выполнения команды: A(:)?
5) Что делает функция magic(n)?
6) Как записать команду, которая удалит из матрицы все элементы с нечетными номерами?
7) Что произойдет в результате выполнения команды c(:,3) = []?
8) Как вывести все элементы 3 строки матрицы?
9) Что получится в результате выполнения команды A(:,2:3)?
10) Будут ли отличаться результаты выполнения команд: A(2, 2) и A(5) для квадратной матрицы третьего порядка?

3 Графические возможности Matlab

3.1 Лабораторная работа №5. Графика и визуализация данных

3.1.1 Теоретические сведения

Matlab обладает широким набором средств для построения графиков функций одной и двух переменных и отображения различных типов данных. Все графики выводятся в графические окна со своими меню и панелями инструментов. Вид графиков определяется аргументами графических команд и затем может быть изменен при помощи инструментов графического окна. Важно понимать, что для построения графиков функций на некоторой области изменения аргументов следует вычислить значения функции в точках области, часто для получения хороших графиков следует использовать достаточно много точек.

Построение графиков в декартовой системе координат

Команда *plot (x, y)* служит для построения графиков в прямоугольной декартовой системе координат.

Тип линии, цвет и маркеры определяются значением третьего дополнительного аргумента функции *plot*. Этот параметр указывается в апострофах, например, вызов plot (x, f, 'ro:') приводит к построению графика красной пунктирной линией, размеченной круглыми маркерами. Всего в дополнительном аргументе может быть заполнено три позиции, соответствующие цвету, типу маркеров и стилю линии. Обозначения для них приведены в таблице 7. Порядок позиций может быть произвольный, допустимо указывать только один или два параметра, например, цвет и тип маркеров.

Разберем сначала, как построить график функции одной переменной, к примеру, $f(x) = \sin x + i \cos 3x$ на отрезке [-2π, 2π]:

1) задаем область (диапазон) построения для переменной *x:*

```
>> x = -2*pi: 0.02: 2*pi;
```

2) поэлементно вычисляем значения функции для каждого элемента вектора *x* и записываем результат в вектор *f:*

```
>> f = sin(x) + i.*cos(3.*x);
```

3) строим график функции используя функцию *plot:*

```
>> plot (x, y, 'b: *')
```

Последовательность команд, записанная выше, приводит к появлению графического окна Figure №1 с графиком функции (рисунок 1).

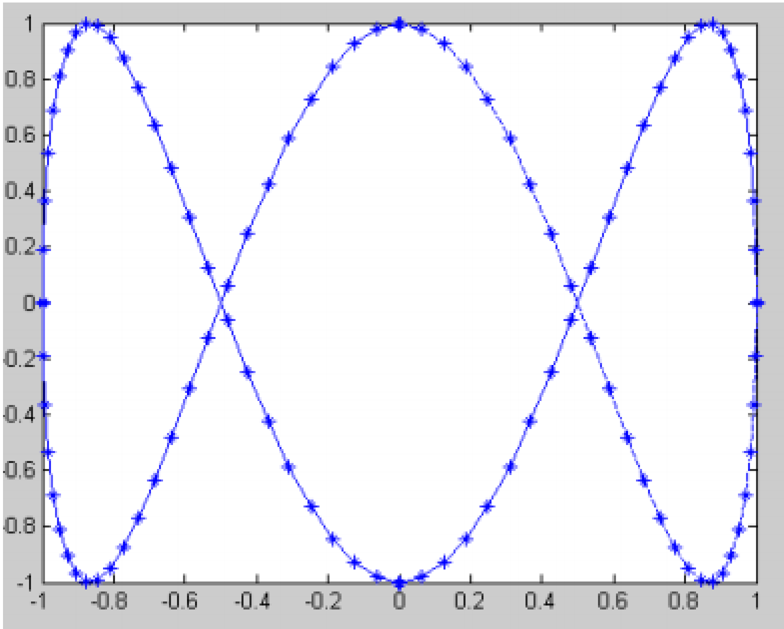

Рисунок 1 – График функции $f(x) = sin\,x + i\,cos\,3x$

Таблица 7 – Сокращения для цвета, типа маркеров и стиля линий

Цвет		Тип линии		Тип маркера	
y	Желтый	-	Сплошная	.	Точка
m	Розовый	:	Пунктирная	o	Кружок
c	Голубой	-.	Штрих-пунктирная	x	Крестик
r	Красный	--	Штриховая	+	Знак плюс
g	Зеленый			*	Звездочка
b	Синий			s	Квадрат
w	Белый			d	Ромб
k	Черный			v	Треугольник вершиной вниз
				^	Треугольник вершиной вверх
				<	Треугольник вершиной влево
				>	Треугольник вершиной вправо
				p	Пятиконечная звезда
				h	Шестиконечная звезда

Построение нескольких графиков в одной системе координат

Функция *plot* имеет достаточно универсальный интерфейс, она, в частности, позволяет отображать графики нескольких функций на одних осях и при этом свойства всех линий могут быть различными.

Рассмотрим пример построения трех функций с различным стилем представления каждой (рисунок 2):

```
>> x = -2*pi:2*pi;
>> y1 = sin(x);
>> y2 = sin(x).^2;
>> y3 = sin(x).^3;
>> plot(x, y1, '– b', x, y2, '-.+ r', x, y3, '– ok')
```

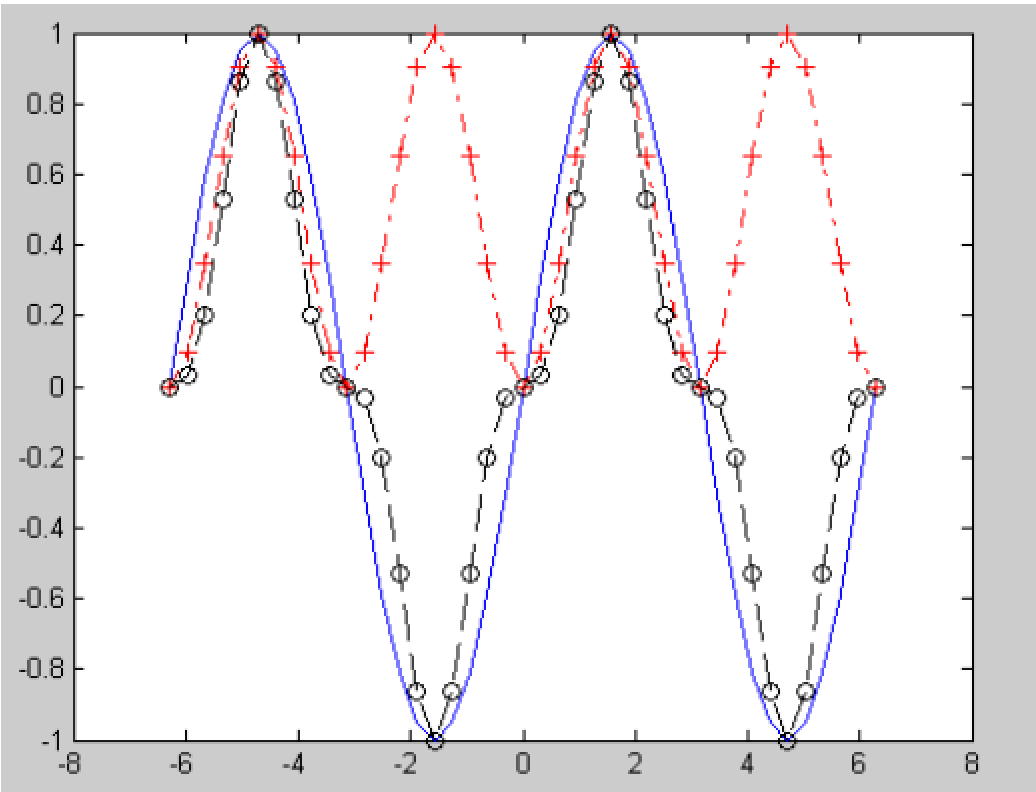

Рисунок 2 – Графики функций y1, y2, y3

Для получения графика кусочно-заданной функции:

$$y(x)=\begin{cases} sin\,x & -4\pi \le x \le \pi \\ 3\left(\dfrac{x}{\pi}+1\right)^2 & -\pi < x \le 0 \\ 3e^{-x} & 0 < x \le 5 \end{cases}$$

достаточно выполнить последовательность команд:

```
>> x1 = [-4*pi: pi/10: -pi];
>> y1 = sin(x1);
>> x2 = [-pi: pi/30: 0];
>> y2 = 3*(x2/pi+1).^2;
>> x3 = [0:0.02:5];
>> y3 = 3*exp(-x3)
>> plot (x1,y1,x2,y2,x3,y3)
```

Заметьте, что графики ветвей функции будут отображаться различными цветами. Можно было поступить и по-другому, а именно: после заполнения *x1, y1, x2, y2, x3* и *y3* собрать вектор *x* для значений аргумента и вектор *y* для значений *y*(*x*) и построить зависимость *y* от *x*:

```
>> x = [x1 x2 x3];
>> y = [y1 y2 y3];
>> plot (x, y)
```

Построим график *параметрически заданной функции*, например график астроиды: $x(t)=cos^3(t),\ y(t)=sin^3(t),\ t\in[0,2\pi]$. Для этого следует задать вектор *t*, затем в векторы *x, y* занести значения *x*(*t*), *y*(*t*) и воспользоваться *plot* для отображения зависимости *y* от *x*:

```
>> t = [0:pi/20:2*pi];
>> x = cos(t).^3;
>> y = sin(t).^3;
>> plot (x, y)
```

Функция *comet* позволяет проследить за движением точки по траектории параметрически заданной линии. Вызов comet(x, y) приводит к появлению графического окна, на осях которого рисуется перемещение точки в виде движения кометы с хвостом. Управление скоростью движения осуществляется изменением шага при определении вектора значений параметра.

Построение графиков в полярной системе координат

Для построения графиков функций в полярной системе координат используются команда *polar(φ, ρ)*.Например, построим график функции *r = sin(3φ)*, где $\varphi \in [0,\ 2\pi]$ в полярной системе координат (рисунок 3)

```
>> phi = 0:0.1:2*pi;
>> r = sin(3*phi);
>> polar (phi, r)
```

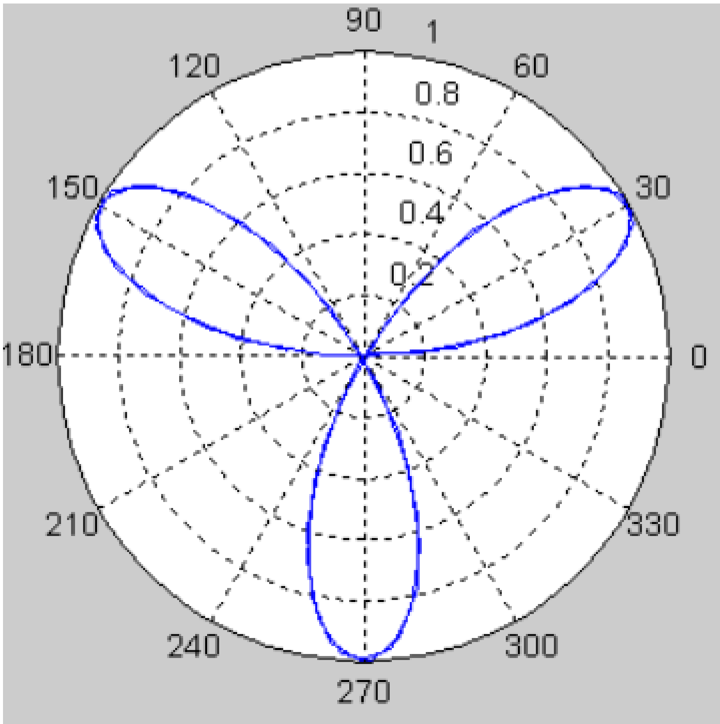

Рисунок 3 – График функции *r = sin(3φ)* в полярной системе координат

Построение нескольких графиков. Вывод графиков в отдельные окна

Первый вызов любой графической функции приводит к появлению на экране графического окна Figure №1, содержащего оси с графиком. Однако при дальнейших обращениях к графическим функциям прежний график пропадает, а новый выводится в тоже самое окно. Команда *figure* предназначена для создания пустого графического окна. Если требуется получить несколько графиков в разных окнах, то перед вызовом графических функций следует прибегать к *figure*. Графические окна при этом нумеруются так: Figure №2, Figure №3 и т. д.

Каждое окно имеет свои оси, при наличии нескольких пар осей (в одном окне или в разных) вывод графиков производится в текущие оси. Последняя созданная пара осей является текущей. Для того чтобы выбрать текущие оси из нескольких имеющихся, достаточно щелкнуть по ним левой кнопкой мыши перед вызовом графической функции. Возможна и обратная ситуация, когда в процессе работы требуется добавлять графики к уже имеющимся на некоторых осях. В этой ситуации перед добавлением графика следует выполнить команду *hold on*. Для завершения такого режима достаточно воспользоваться *hold off*.

Построение нескольких графиков в одном графическом окне на разные оси

В одном графическом окне можно расположить несколько осей со своими графиками. Команда *subplot* предназначена для разбиения графического окна на части и определения текущей из них. Данная функция используется с тремя параметрами: subplot (i, j, n). Здесь *i* и *j* – число подграфиков по вертикали и горизонтали, а n – номер подграфика, который надо сделать текущим. Номер отсчитывается от левого верхнего угла построчно. Например,

```
>> x = -pi: 0.01: pi;
>> y = sin(x);
>> z = cos(x);
>> subplot(2,1,1);
>> plot(x, y, 'c')
>> subplot(2,1,2);
>> plot(x, z, 'g')
```

В результате появится графическое окно с двумя подграфиками (рисунок 4).

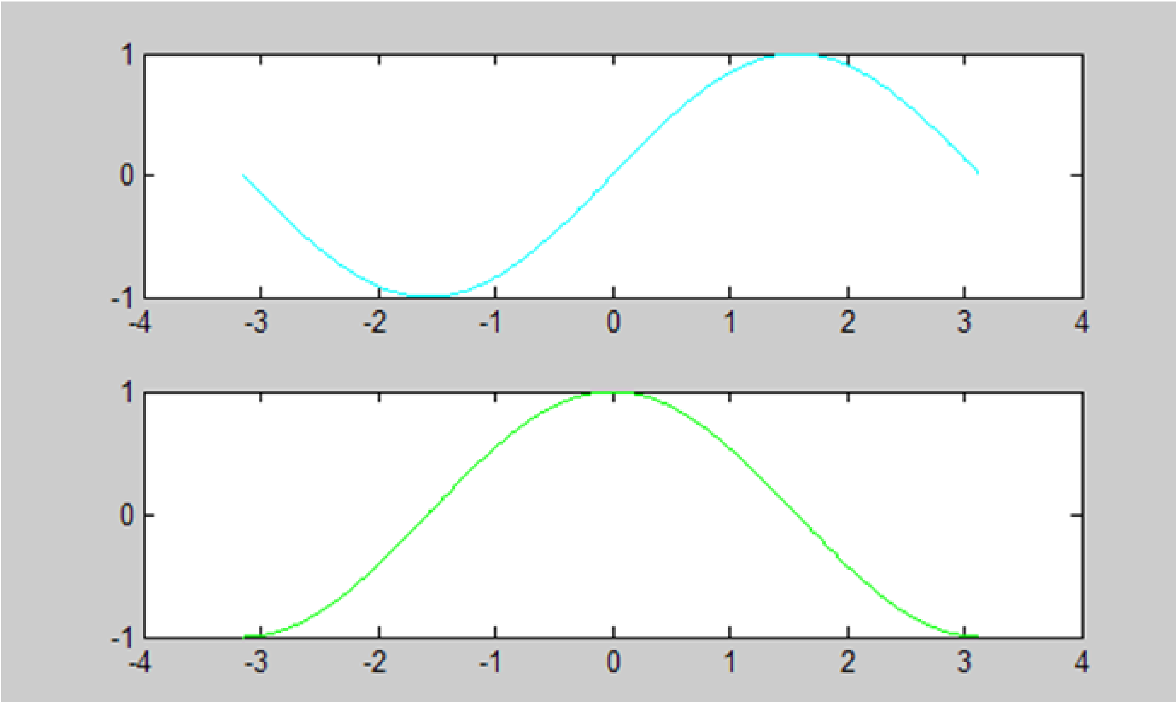

Рисунок 4 – Графики функций $y = sin(x), z = cos(x)$

Оформление графиков

Графики оформляются в Matlab специальными командами и функциями. Сетка наносится на оси командой *grid on*, а убирается при помощи *grid off*. Заголовок размещается в графическом окне посредством функции *title*, входным аргументом которой является строка, заключенная в апострофы:

>> title ('Результаты эксперимента')

При наличии нескольких графиков требуется расположить легенду, обратившись к *legend*. Надписи легенды, заключенные в апострофы, указываются во входных аргументах функции *legend*, их число должно совпадать с числом линий графиков. Кроме того, последний дополнительный входной аргумент определяет положение легенды:

–1 – вне графика в правом верхнем углу графического окна;

0 – выбирается лучшее положение в пределах графика так, чтобы как можно меньше перекрывать сами графики;

1 – в верхнем правом углу графика (это положение используется по умолчанию);

2 – в верхнем левом углу графика;

3 – в нижнем левом углу графика;

4 – в нижнем правом углу графика.

Функции *xlabel* и *ylabel* предназначены для подписей к осям, их входные аргументы так же заключаются в апострофы.

Построение трехмерных графиков

Рассмотрим здесь только построение графиков функций двух переменных на прямоугольной области определения. Трехмерные поверхности обычно описываются функцией двух переменных *z(x, y)*. Специфика построения трехмерных графиков требует не просто задания векторов *x, y* – она требует определения двумерных массивов *X, Y* – матриц. Для создания таких массивов служит команда *meshgrid.*

Предположим, что требуется получить поверхность функции $z(x,y) = e^{-x}\sin\pi y$ на прямоугольнике $x \in [-1,1]$, $y \in [0,2]$:

```
>> [X,Y] = meshgrid (-1:0.1:1,0:0.1:2);
>> Z = exp(-X).*sin(pi*Y);
>> mesh(X,Y,Z)
```

В результате на экране появляется графическое окно, содержащее каркасную поверхность исследуемой функции (рисунок 5). Обратите внимание, что цвет поверхности соответствует значению функции.

Команда *colorbar* приводит к отображению в графическом окне столбика, показывающего соотношение между цветом и значением *z*(*x, y*). Цветовые палитры графика можно изменять, пользуясь функцией *colormap*, например colormap(gray) отображает график в оттенках серого цвета. Некоторые цветовые палитры приведены ниже:

- bone – похожа на палитру gray, но с легким оттенком синего цвета;
- colorcube – каждый цвет изменяется от темного к яркому;
- cool – оттенки голубого и пурпурного цветов;
- copper – оттенки медного цвета;
- hot – плавное изменение: черный-красный-оранжевый-желтый-белый;

– hsv – плавное изменение (как цвета радуги);
– jet – плавное изменение: синий-голубой-зеленый-желтый-красный;
– spring – оттенки пурпурного и желтого;
– summer – оттенки зеленого и желтого;
– winter – оттенки синего и зеленого;

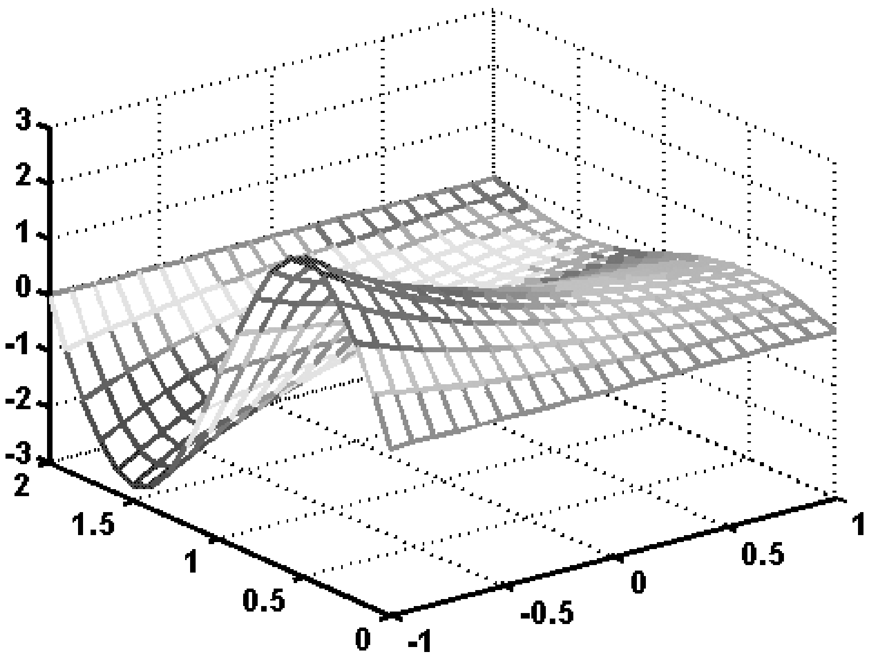

Рисунок 5 – График поверхности $z(x,y) = e^{-x} \sin \pi y$

Matlab предоставляет целый набор графических функций для визуализации функций двух переменных, среди них:

– *surf* – залитая цветом каркасная поверхность;
– *meshc*, *surfc* – поверхности с линиями уровня на плоскости *xy*;
– *contour* – плоский график с линиями уровня;
– *contourf* – залитый цветом плоский график с линиями уровня;
– *contour3* – поверхность, составленная из линий уровня;
– *surfl* – освещенная поверхность

Все перечисленные функции допускают то же самое обращение, что и *mesh*, например:

>> surf (X,Y,Z)

>> contour (X,Y,Z)

Остановимся подробнее на нескольких вопросах. Первый из них: как изменять установки, определённые по умолчанию, при отображении функций линиями уровня при помощи *contour*, *contourf* и *contour3*. Число линий уровня задается в четвертом дополнительном аргументе, например:

>> contourf (X,Y,Z,10)

Вместо числа линий уровня можно указать в векторе те значения $z(x, y)$, для которых требуется построить линии уровня:

>> contour (X,Y,Z,[-0.51 -0.25 -0.01 0.89])

Несколько сложнее нанести подписи с соответствующим значением $z(x,y)$ к каждой линии уровня. Для этого придется вызвать *contour* с двумя выходными аргументами, первый из них — матрица с информацией о положении линий уровня, а второй — вектор с указателями на линии. Полученные переменные следует использовать в качестве входных аргументов функции *clabel*:

>> [CMatr, h] = contour(X, Y, Z,[-0.51 -0.25 -0.01 0.89]);

>> clabel (CMatr, h)

Залитые цветом каркасные поверхности, построенные при помощи *sufr* и *surfc*, имеют постоянный цвет в пределах каждой ячейки. Команда *shading interp*, вызываемая после *surf* и *surfc*, служит для плавного изменения цвета в пределах ячеек и скрытия линий сетки на поверхности. Если желательно убрать сетку и сохранить постоянный цвет ячеек, то достаточно использовать *shading flat*, а *shading faceted* придает графику прежний вид.

Графические функции по умолчанию располагают поверхность так, что наблюдатель видит ее часть под некоторым углом, а другая – скрыта от взора. Положение наблюдателя определяется двумя углами: азимутом (AZ) и углом возвышения (EL). Азимут отсчитывается от оси, противоположной y, а угол возвышения – от плоскости xy (рисунок 6).

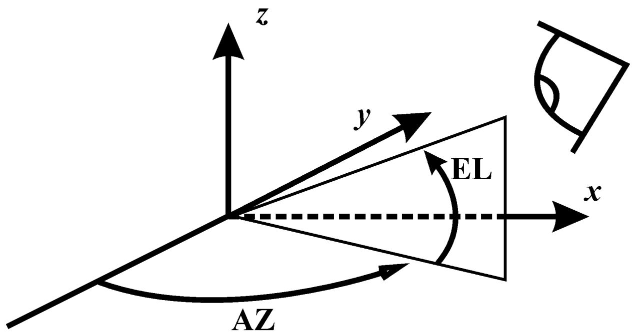

Рисунок 6 – Положительные направления отсчета углов азимута и возвышения

Осмотреть поверхность со всех сторон позволяет функция *view*. Вызов функции *view* с двумя выходными аргументами и без входных дает возможность определить текущее положение наблюдателя (углы выводятся в градусах):

>> [AZ, EL] = view

AZ =

-37.5000

EL =

30

Эти значения Matlab использует по умолчанию при построении трехмерных графиков. Для задания положения наблюдателя следует указать азимут и угол возвышения (в градусах) в качестве входных аргументов view, например: view(0,90) показывает вид на график сверху. Перед поворотом графика целесообразно расставить обозначения к осям, используя, как и для двумерных графиков *xlabel* и *ylabel*, и *zlabel* для подписи к вертикальной оси. Функция *view* допускает еще несколько вариантов вызова:

- view(3) – возврат к стандартным установкам;
- view([x,y,z]) – помещение наблюдателя в точку с координатами *x, y* и *z*.

Освещенная поверхность строится при помощи функции *surfl*, которая позволяет получить наглядное представление о поведении исследуемой функции. Следует учесть, что лучше сочетать вызов *surfl* с командой *shading interp* и цветовой палитрой, содержащей большое количество оттенков (*gray, copper, bone, winter* и т.д.), поскольку поверхность обладает свойствами рассеивания, отражения и поглощения света, исходящего от некоторого источника. Положение источника можно задавать в четвертом дополнительном аргументе *surfl*, причем либо вектором из двух элементов (азимут и угол возвышения источника), либо вектором из трех элементов (положение источника света в системе координат осей), например: surfl(X,Y,Z,[20 80]) или surfl(X,Y,Z,[6 8 11]).

3.1.2 Практическая часть

Контрольные задания

1) Построить графики функций одной переменной на указанных интервалах. Вывести графики различными способами:

a) в отдельные графические окна;

b) в одно окно на одни оси;

c) в одно окно на отдельные оси.

Дать заголовки, разместить подписи к осям, легенду, использовать различные цвета, стили линий и типы маркеров, нанести сетку.

Варианты:

1. $f(x) = sin\,x,\ g(x) = sin^2\,x,\ x \in [-2\pi; 3\pi]$

2. $u(x) = \dfrac{x}{20},\ v(x) = e^x,\ x \in [-2; 2]$

3. $f(x) = x^3 + 2x^2 + 1,\ g(x) = (x-1)^4,\ x \in [-1; 1]$

4. $f(x) = ln\,x,\ g(x) = x\,ln\,x,\ x \in [0.2; 10]$

5. $f(x) = |2x|^3,\ g(x) = |2x|^5,\ x \in [-0.5; 0.5]$

6. $u(x) = x^2,\ v(x) = x^3,\ x \in [-1; 1]$

7. $f(x) = arcsin\,x,\ g(x) = arccos\,x,\ x \in [-1; 1]$

8. $u(x) = e^x,\ v(x) = e^{-x},\ x \in [-0.6; 0.6]$

9. $f(x) = \dfrac{sin\,x}{x},\ g(x) = e^{-x}\cos x,\ x \in [0.01; 2\pi]$

10. $u(x)=\dfrac{1}{1+x}$, $v(x)=\dfrac{1}{1+\dfrac{1}{1+x}}$, $x\in[0;1]$

11. $u(x)=0.01x^2$, $v(x)=e^{-|x|}$, $x\in[-0.2;9.4]$

12. $f(x)=sin\,x^2$, $g(x)=cos\,x^2$, $x\in[-\pi;\pi]$

13. $u(x)=\sqrt{x}$, $v(x)=e^{-x^2}$, $x\in[0;1]$

14. $f(x)=sin(ln(x+1))$, $g(x)=cos(ln(x+1))$, $x\in[0;2\pi]$

15. $u(x)=arctgx$, $v(x)=arctg3x$, $x\in[-1;1]$

2) Построить график кусочно-заданной функции, отобразить ветви разными цветами и маркерами.

Варианты:

1. $f(x)=\begin{cases}-1 & -3\le x\le -1\\ x & -1<x\le 1\\ e^{1-x} & 1<x\le 3\end{cases}$

2. $f(x)=\begin{cases}\dfrac{3+sin^2 x}{1+x^2} & x\le 0\\ 2x^2\,cos^2 x & x>0\end{cases}$

3. $f(x)=\begin{cases}ln\,x & 1\le x\le e\\ \dfrac{x}{e} & e<x\le 9\\ 9e^{8-x} & 9<x\le 12\end{cases}$

4. $f(x)=\begin{cases}sin\,x & -2\pi\le x\le 0\\ -x^3 & 0<x\le 1\\ cos\,\pi x & 1<x\le 3\pi\end{cases}$

5. $f(x)=\begin{cases}\sqrt{1+2x^2+sin^2 x} & x\le 0\\ \dfrac{2+x}{\sqrt[3]{2+e^{-0.5x}}} & x>0\end{cases}$

6. $f(x)=\begin{cases}|x| & -2\le x\le 1\\ sin\,\dfrac{\pi}{2}x & 1<x\le 2\\ (2-x)^3 & 2<x\le 3\end{cases}$

7. $f(x)=\begin{cases}(x-1)^2 & -2\le x\le 1\\ cos\,\dfrac{\pi}{2}x & 1<x\le 3\\ 1-e^{3-x} & 3<x\le 8\end{cases}$

8. $f(x)=\begin{cases}\dfrac{\sqrt{1+|x|}}{2+|x|} & x\le 0\\ \dfrac{1+x}{2+cos^3 x} & x>0\end{cases}$

9. $f(x)=\begin{cases}e^{x+1} & -2\le x\le -1\\ x^2 & -1<x\le 1\\ (2-x)^3 & 1<x\le 2\end{cases}$

10. $f(x)=\begin{cases} \dfrac{1+x}{\sqrt[3]{1+x+x^2}} & x\le 0 \\ \dfrac{1+cos^4 x}{3+x} & x>0 \end{cases}$

11. $f(x)=\begin{cases} \sqrt{x} & 0\le x\le 1 \\ 1 & 1<x\le 3 \\ (x-4)^2 & 3<x\le 5 \end{cases}$

12. $f(x)=\begin{cases} e^x & -2\le x\le -1 \\ \dfrac{|x|}{e} & -1<x\le 1 \\ e^{-x} & 1<x\le 2 \end{cases}$

13. $f(x)=\begin{cases} \sqrt[3]{1+x^2} & x\le 0 \\ sin^2 x+\dfrac{1+x}{1+e^x} & x>0 \end{cases}$

14. $f(x)=\begin{cases} arcsin\, x-1 & 0\le x\le 1 \\ \dfrac{\pi}{2}-x & 1<x\le \dfrac{\pi}{2} \\ cos\, x & \dfrac{\pi}{2}<x\le \pi \end{cases}$

15. $f(x)=\begin{cases} x^2 \log_2 x & 1\le x\le 2 \\ \dfrac{x^3}{2} & 2<x\le 3 \\ \dfrac{x^x}{2} & 3<x\le 3.5 \end{cases}$

3) Построить график параметрически заданной функции, используя *plot* и *comet*.

Варианты:

1. $x(t)=t-sin\,t,\ y(t)=1-cos\,t$.

2. $x(t)=2\,sin\,t-\frac{2}{3}sin\,2t,\ y(t)=2\,cos\,t-\frac{2}{3}cos\,2t$.

3. $x(t)=9\,sin\frac{t}{10}-\frac{1}{2}sin\frac{9}{10}t,\ y(t)=9\,cos\frac{1}{10}t+\frac{1}{2}cos\frac{9}{10}t$.

4. $x(t)=cos\,t,\ y(t)=sin(sin\,t)$.

5. $x(t)=e^{-t}cos\,t,\ y(t)=sin\,t$.

6. $x(t)=e^{-t}cos\,t,\ y(t)=e^{t}sin\,t$.

7. $x(t)=t(t-2\pi),\ y(t)=sin\,t$.

8. $x(t)=sin\,t(t-2\pi),\ y(t)=sin\,t$.

9. $x(t)=sin\,t(t-2\pi),\ y(t)=sin\,t\cdot cos\,t$.

10. $x(t) = \sin t + \cos^3 t,\ y(t) = \sin t \cdot \cos t$.

11. $r(t) = \cos t\ ,\ \varphi(t) = 2t - \sin(t)$

12. $r(t) = 3\sin t,\ \varphi(t) = \cos t + t$

13. $r(t) = \cos^2 t + 1\ ,\ \varphi(t) = t - 2$

14. $r(t) = 2 - \cos^2 t,\ \varphi(t) = 2t + 1$

15. $r(t) = 1 + \sin^2 t\ ,\ \varphi(t) = 2t + \sin(t)$

4) Визуализировать функцию двух переменных на прямоугольной области определения различными способами:

a) каркасной поверхностью;

b) залитой цветом каркасной поверхностью;

c) промаркированными линиями уровня (самостоятельно выбрать значения функции, отображаемые линиями уровня);

d) освещенной поверхностью.

Расположить графики в отдельных графических окнах и в одном окне с соответствующим числом пар осей. Представить вид каркасной или освещенной поверхности с нескольких точек обзора.

Варианты:

1. $z(x, y) = \sin x \cdot e^{-3y}$, $x \in [0, 2\pi]$, $y \in [0, 1]$

2. $z(x, y) = \sin^2 x \cdot \ln y$, $x \in [0, 2\pi]$, $y \in [1, 10]$

3. $z(x, y) = \sin^2(x - 2y) \cdot e^{-|y|}$, $x \in [0, \pi]$, $y \in [-1, 1]$

4. $z(x, y) = \dfrac{x^2 y^2 + 2xy - 3}{x^2 + y^2 + 1}$, $x \in [-2, 2]$, $y \in [-1, 1]$

5. $z(x, y) = \dfrac{\sin xy}{x}$, $x \in [0.1, 5]$, $y \in [-\pi, \pi]$

6. $z(x, y) = \left(\sin x^2 + \cos y^2\right)^{xy}$, $x \in [-1, 1]$, $y \in [-1, 1]$

7. $z(x, y) = \operatorname{arctg}(x + y)(\arccos x + \arcsin y)$, $x \in [-1, 1]$, $y \in [-1, 1]$

8. $z(x,y)=(1+xy)(3-x)(4-y),\ x\in[0,3],\ y\in[0,4]$

9. $z(x,y)=e^{-|x|}\left(x^5+y^4\right)\sin xy,\ x\in[-2,2],\ y\in[-3,3]$

10. $z(x,y)=\left(y^2-3\right)\sin\dfrac{x}{|y|+1},\ x\in[-2\pi,2\pi],\ y\in[-3,3]$

11. $z(x,y)=e^x y+\cos x,\ x\in[1,5],\ y\in[1,5]$

12. $z(x,y)=x^2y+\dfrac{1}{1+x},\ x\in[1,3],\ y\in[1,3]$

13. $z(x,y)=x^3y+\cos x,\ x\in[0,1],\ y\in[0,1]$

14. $z(x,y)=y\cos x+\cos^2 x,\ x\in[1,2],\ y\in[1,2]$

15. $z(x,y)=\dfrac{e^{-xy}}{x}+x,\ x\in[-3,-2],\ y\in[-3,-2]$

3.1.3 Контрольные вопросы

1) С помощью какой команды осуществляется построение графиков декартовой и полярной системах координат?

2) Как построить несколько графиков в одной системе координат?

3) Какая функция позволяет разделить графическое окно MatLab на несколько подокон и вывести в каждом из них графики различных функций?

4) Как изменить цвет и стиль линий на графиках?

5) Как добавить к графикам сетку из координатных линий, названия осей, легенду, заголовок для графика?

6) Какая функция используется для построения динамического графика?

7) Для чего используется команда *hold*?

8) Для чего служит команда *meshgrid*?

9) Как построить график функции двух переменных?

10) Какой командой можно создать новое графическое окно?

Список использованных источников

1 Ануфриев, И.Е. Самоучитель MatLab 5.3/6.x / И.Е. Ануфриев. – СПб: БХВ-Петербург, 2002. – 736 с.

2 Гультяев, А.П. Визуальное моделирование в среде MATLAB: учебный курс/ А.П. Гультяев. – СПб.: Питер, 2000.– 432 с.

3 Дьяконов, В.П. Matlab: учебный курс / В.П. Дьяконов. – СПб.: Питер, 2001. – 592 с.

4 Дьяконов, В.П. Matlab 6.5: Основы применения / В.П. Дьяконов. – М.: СОЛН-Пресс, 2005

5 Иглин, С.П. Математические расчеты на базе MATLAB/ С.П. Иглин. – СПб.: БХВ-Петербург, 2005. – 640 с.

6 Колокольникова, А.И. Спецразделы информатики: введение в MatLab: учебное пособие [Электронный ресурс] / А.И. Колокольникова, А.Г. Киренберг. – М., Берлин: Директ-Медиа, 2014. – 73 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=275268

7 Конев, В.Ю. Основные функции пакета MATLAB: учебное пособие/ В.Ю. Конев, Л.А. Мироновский. – 2-ое издание. – СПб.: ГААПСПб., 1994. – 76 с.

8 Кривилев, А.В. Основы компьютерной математики с использованием системы MATLAB: учебное пособие / А.В. Кривилев. – М.: Лекс-Книга, 2005. – 496 с.

9 Потемкин, В.Г. MATLAB 6: среда проектирования инженерных приложений/ В.Г. Потемкин. – М.: Диалог-МИФИ, 2003. – 448 с.

10 Потемкин, В. Г. Система MATLAB: справочное пособие/ В. Г. Потемкин. – М.: ДИАЛОГ-МИФИ, 1997.– 350 с.