

АЛМАТИНСКИЙ ФИЛИАЛ НЕГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ ПРОФСОЮЗОВ»

---



**Н.Ф. СТИФУТИНА**

# **ОСНОВЫ КОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ ТЕХНОЛОГИЙ**

**ПРАКТИКУМ  
для самостоятельной работы**

**Алматы  
2015**

**Автор-составитель:  
СТИФУТИНА Н.Ф.,**

старший преподаватель кафедры экономики, информатики и математики  
Алматинского филиала НОУ  
«Санкт-Петербургский Гуманитарный университет профсоюзов»

Рекомендовано к печати

Учебно-методическим советом Алматинского филиала  
Санкт-Петербургского Гуманитарного университета профсоюзов  
от « 18 » февраля 2015 г. Протокол № 4

© Стифутина Н.Ф., 2015  
© АФ НОУ ВПО «СПбГУП», 2015

## СОДЕРЖАНИЕ

<b>Введение</b> .....	4
<b>1. Методические указания по выполнению лабораторных работ</b> .....	5
<b>2. Лабораторные работы</b>	
Лабораторная работа № 1 «Векторы и операции над ними».....	6
Лабораторная работа № 2 «Матрицы и операции над ними».....	9
Лабораторная работа № 3 «Создание матриц с заданными свойствами».....	14
Лабораторная работа № 4 «Графические команды и функции».....	19
Лабораторная работа № 5 «Команды и функции трехмерной графики».....	27
Лабораторная работа № 6 «Элементы программирования в MatLab. Программы линейной и разветвляющейся структуры».....	35
Лабораторная работа № 7 «Элементы программирования в MatLab. Программы циклической структуры».....	39
Лабораторная работа № 8 «Элементы программирования в MatLab. Программы циклической структуры».....	42
Лабораторная работа № 9 «Элементы программирования в MatLab. Программы циклической структуры».....	44
Лабораторная работа № 10 «Элементы программирования в MatLab. Вычисление корней полиномов».....	45
<b>Список литературы</b> .....	47

## ВВЕДЕНИЕ

Дисциплина «Основы компьютерных вычислительных технологий» изучается студентами очной и заочной форм обучения и является логическим продолжением и развитием дисциплины «Информатика и программирование», продолжая подготовку студентов в области программирования.

Целью указанной дисциплины является рассмотрение процедурных и психологических проблем перехода от традиционных вычислений к компьютерным, введение в курс проблем компьютерных вычислительных технологий, ознакомление с методами компьютерных вычислительных технологий для экономических систем, развитие умений применять эти знания на практике.

Практическая часть курса предусматривает формирование навыков использования предметных задач для последующего компьютерного решения, необходимость уметь оценивать точность результатов компьютерных вычислений.

В результате освоения дисциплины студенты должны:

- иметь представление о принципах доведения основных математических задач до численных результатов и о подходах к оценке точности и предметной трактовке результатов;
- уметь решать типовые вычислительные задачи с помощью современных программных средств конечного пользователя;
- владеть методикой и техникой математического и компьютерного вычисления, практическими навыками выполнения математических расчетов в среде пакета Matlab.

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- способность при решении профессиональных задач анализировать социально-экономические проблемы и процессы с применением методов системного анализа и математического моделирования;
- способность применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы.

Выполнение приводимых практических заданий способствует систематизации, закреплению и углублению теоретических знаний, развитию умения творчески применять их для решения конкретных задач.

# 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Данное издание содержит десять лабораторных работ, посвященных реализации технологии численного решения средствами наиболее распространённого математического пакета Matlab. Выбор этого пакета ориентирован на осмысленное и эффективное применение имеющихся готовых методических, алгоритмических и программных разработок.

Каждая выполняемая на основе рассмотренных заданий лабораторная работа включает формулировку цели работы, описание порядка ее выполнения, требования к разрабатываемой программе, а также необходимые пояснения.

Подготовка к каждой лабораторной работе производится во внеаудиторное время. Выполнив лабораторную работу, студент оформляет отчет в текстовом редакторе Word, который состоит из следующих разделов:

1. Титульный лист.
2. Номер и тема лабораторной работы.
3. Цель работы.
4. Вариант задания.
5. Выполненные индивидуальные задания с пояснением используемых функций и операторов.
6. Листинг программы (результаты вычислений).

При защите отчета необходимо уметь пояснять работу программы и отвечать на вопросы преподавателя по данной теме.

Примеры решения задач и теоретические вопросы по необходимой теме можно найти на сайте **Exponenta.ru**.

## 2. ЛАБОРАТОРНЫЕ РАБОТЫ

### ЛАБОРАТОРНАЯ РАБОТА № 1 «ВЕКТОРЫ И ОПЕРАЦИИ НАД НИМИ»

**Цель работы:** используя всевозможные команды и функции MatLab, научиться создавать векторы и производить над ними элементарные операции.

**Методические рекомендации:**

В системе MatLab достаточно просто выполняются математические операции над матрицами и векторами. Рассмотрим сначала простые операции сложения и умножения матриц и векторов. Пусть даны два вектора

a = [1 2 3 4 5];	% вектор-строка
b = [1; 1; 1; 1; 1];	% вектор-столбец

тогда умножение этих двух векторов можно записать так

c = a*b;	% c=1+2+3+4+5=16
d = b*a;	% d - матрица 5x5 элементов

В соответствии с операциями над векторами, умножение вектор-строки на вектор-столбец дает число, а умножение вектор-столбца на вектор-строку дает двумерную матрицу, что и является результатом вычислений в приведенном примере, т.е.

$$c = \sum_{i=1}^5 a_i b_i = 1+2+3+4+5+6 = 16,$$

$$d = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}.$$

Сложение и вычитание двух векторов записывается так

<code>a1 = [1 2 3 4 5];</code>	
<code>a2 = [5 4 3 2 1];</code>	
<code>c = a1+a2;</code>	<code>% c = [1+5, 2+4, 3+3, 4+2, 5+1];</code>
<code>c = a2-a1;</code>	<code>% c = [5-1, 4-2, 3-3, 2-4, 1-5];</code>

Следует обратить внимание, что операции сложения и вычитания можно выполнять между двумя векторами-столбцами или двумя векторами-строками. Иначе MatLab выдаст сообщение об ошибке, т.к. разнотипные векторы складывать нельзя. Так обстоит дело со всеми недопустимыми арифметическими операциями: в случае невозможности их вычисления система MatLab сообщит об ошибке и выполнение программы будет завершено на соответствующей строке.

Для поиска максимального значения элемента вектора используется стандартная функция `max()`, которая возвращает найденное максимальное значение элемента и его позицию (индекс):

<code>a = [1 6 3 4];</code>	
<code>[v, i] = max(a);</code>	<code>% v = 6, i = 2;</code>

или

<code>v = max(a);</code>	<code>% v = 6;</code>
--------------------------	-----------------------

Приведенный пример показывает два разных способа вызова функции `max()`. В первом случае определяется и максимальное значение элемента и его индекс в векторе, а во втором – только максимальное значение элемента.

Полный синтаксис функции `max()` можно узнать, набрав в командном окне MatLab команду

<code>help &lt;название функции&gt;</code>
--

Аналогичным образом работает функция `min()`, которая определяет минимальное значение элемента вектора или матрицы и его индекс.

Другой полезной функцией работы с векторами является функция `sum()`, которая вычисляет сумму значений элементов вектора:

<code>a = [3 5 4 2 1];</code>	
<code>s = sum(a);</code>	<code>% s = 3+5+4+2+1=15</code>

Для сортировки значений элементов вектора по возрастанию или убыванию используется функция `sort()` следующим образом:

<code>a = [3 5 4 2 1];</code>	
<code>b1 = sort(a);</code>	<code>% b1=[1 2 3 4 5]</code>
<code>b2 = sort(a, 'descend');</code>	<code>% b2=[5 4 3 2 1]</code>
<code>b3 = sort(a, 'ascend');</code>	<code>% b3=[1 2 3 4 5]</code>

Еще одной полезной функцией работы с векторами является функция `mean()` для вычисления среднего арифметического значения, которая работает следующим образом:

<code>a = [3 5 4 2 1];</code>	
<code>m = mean(a);</code>	<code>% m = 3</code>

### ***Практические задания:***

1. Даны два вектора  $A(n)$  и  $B(n)$ . Найти их сумму и поместить в вектор  $C$ . Отсортировать вектор  $C$  по возрастанию. Изменить размерность вектора  $B(n)$  на  $B(m)$  и найти сумму векторов  $A(n)$  и  $B(m)$ .
2. Даны два вектора  $A(n)$  и  $B(n)$ . Найти их разность и поместить в вектор  $C$ . Изменить размерность вектора  $B(n)$  на  $B(m)$  и найти разность векторов  $A(n)$  и  $B(m)$ .
3. Даны два вектора  $A(n)$  и  $B(n)$ . Найти их произведение и поместить в вектор  $C$ .
4. Даны два вектора  $A(n)$  и  $B(n)$ . Найти частное от деления и поместить в вектор  $C$ . Использовать операторы `/`, `./`, `\`, `.\`.
5. Даны два вектора  $A(n)$  и  $B(n)$ . Объединить их в вектор  $C$ : построчно и по столбцу. Изменить размерность вектора  $B(n)$  на  $B(m)$  и также произвести объединение векторов  $A(n)$  и  $B(m)$  построчно и по столбцу.
6. Дан вектор  $V(n)$ . Упорядочить его элементы по возрастанию, найти минимальный элемент последовательности.



7. Дан вектор  $M(n)$ . Упорядочить его элементы по возрастанию, найти максимальный элемент последовательности.
8. Дана последовательность из  $n$  действительных чисел. Получить среднее арифметическое всех элементов исходной последовательности.

### *Варианты заданий*

<b>№ варианта</b>	<b>Значение <math>n</math></b>	<b>Значение <math>m</math></b>
1	4	6
2	5	7
3	6	2
4	7	5
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	7
12	5	9

## **ЛАБОРАТОРНАЯ РАБОТА № 2** **«МАТРИЦЫ И ОПЕРАЦИИ НАД НИМИ»**

**Цель работы:** используя всевозможные методы MatLab, научиться создавать матрицы с заданными свойствами и производить над ними элементарные операции.

### **Методические рекомендации:**

Операции умножения и сложения между матрицами выполняются следующим образом:

<code>A = [1 2 3; 4 5 6; 7 8 9];</code>	
<code>B = ones(3);</code>	
<code>C = A+B;</code>	% сложение двух матриц одинакового размера
<code>D = A+5;</code>	% сложение матрицы и числа
<code>E = A*B;</code>	% умножение матрицы A на B

$F = B * A;$	% умножение матрицы B на A
$G = 5 * A;$	% умножение матрицы на число

### Пример 1.

Возведём матрицу в степень

$$A = \begin{pmatrix} 2 & -1 \\ 3 & 1 \end{pmatrix} \quad \text{Для матрицы найдем } A^0, A^1, A^2.$$

$A = [2 \ -1; \ 3 \ 1];$	% Введём матрицу A
$A^0, A^1, A^2$	% Вычислим и выведем результат
<pre>&gt;&gt; ans = 1    0       0    1</pre>	
<pre>ans = 2   -1       9   -2</pre>	
<pre>ans = 1   -3       9   -2</pre>	

### Пример 2.

Вычислим матрицу  $2A - BA$ , где

$$A = \begin{pmatrix} 2 & 3 \\ 1 & 0 \\ -1 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 0 & 1 \\ 1 & -2 & 2 \\ 5 & 0 & 7 \end{pmatrix}$$

$A = [2 \ 3; \ 1 \ 0; \ -1 \ 3];$ $B = [2 \ 0 \ 1; \ 1 \ -2 \ 2; \ 5 \ 0 \ 7];$	% Введём матрицы A и B
$C = 2 * A - B * A;$	% Вычислим $2A - BA$
$disp(C);$	% Выведем результат
<pre>&gt;&gt; 1   -3       4   -9      -5  -30</pre>	

Для поиска максимального значения элемента матрицы также используется стандартная функция `max()`, которая возвращает найденные максимальные значения элемента, стоящие в столбцах, и их позиции (индексы):

<code>A = [4 3 5; 6 7 2; 3 1 8];</code>	
<code>[V, I] = max(A);</code>	<code>% V=[6 7 8], I = [2 2 3]</code>

или

<code>V = max(A);</code>	<code>% V=[6 7 8]</code>
--------------------------	--------------------------

Полный синтаксис функции `max()` можно узнать, набрав в командном окне MatLab команду

<code>help &lt;название функции&gt;</code>
--

Функция `sum()` при работе с матрицами вычисляет сумму значений столбцов матрицы:

<code>A = [4 3 5; 6 7 2; 3 1 8];</code>	
<code>S1 = sum(A);</code>	<code>% S1=[13 11 15]</code>
<code>S2 = sum(sum(A));</code>	<code>% S2=39</code>

При вычислении суммы `S2` сначала вычисляется сумма значений элементов матрицы `A` по столбцам, а затем - по строкам. В результате переменная `S2` содержит сумму значений всех элементов матрицы `A`.

Функция `sort()` для сортировки значений матриц по возрастанию или убыванию используется следующим образом:

<code>A = [4 3 5; 6 7 2; 3 1 8];</code>	
<code>B1 = sort(A);</code>	<code>% B1=[3 1 2</code>
	<code>% 4 3 5</code>
	<code>% 6 7 8]</code>
<code>B2 = sort(A, 'descend');</code>	<code>% B2=[6 7 8</code>
	<code>% 4 3 5</code>
	<code>% 3 1 2]</code>

Функция `mean()` для вычисления среднего арифметического значения с матрицами работает следующим образом:

<code>A = [4 3 5; 6 7 2; 3 1 8];</code>	
<code>M1 = mean(A);</code>	<code>% M1 = [4.333 3.667 5.000]</code>
<code>M2 = mean(mean(A));</code>	<code>% M2 = 4.333</code>

Несколько простых функций служат для перемножения элементов массивов:

`prod(A)` возвращает:

- если  $A$  - вектор, то возвращает произведение элементов массива;
- если  $A$  - матрица, то возвращает вектор-строку, содержащую произведения элементов каждого столбца;

`prod(A,dim)` — возвращает матрицу (массив размерности два) с произведением элементов массива  $A$  по столбцам ( $dim=1$ ), по строкам ( $dim=2$ ), по иным размерностям в зависимости от значения скаляра  $dim$ .

### *Пример 1.*

```
»
A=[1234;2457;6834]
A=
1     2     3     4
2     4     5     7
6     8     3     4
» B = prod(A)
```

`cumprod(A)` — возвращает произведение с накоплением:

- если  $A$  — вектор, то `cumprod(A)` возвращает вектор, содержащий произведения с накоплением элементов вектора  $A$ ;

- если  $A$  — матрица, то `cumprod(A)` возвращает матрицу того же размера, что и  $A$ , содержащую произведения с накоплением для каждого столбца матрицы  $A$ . (Первая строка остается без изменений, во второй строке - произведение первых двух элементов каждого столбца, в третьей строке - элементы второй строки матрицы-результата умножаются на элементы третьей строки матрицы входного аргумента по столбцам и т. д.);

`cumprod(A,dim)` — возвращает произведение с накоплением элементов по строкам или столбцам матрицы в зависимости от значения скаляра  $dim$ .

### *Пример 2.*

```
» A=[1 2 3; 4 5 6; 7 8 9]
A=
```

```

1     2     3
4     5     6
7     8     9
» B = cumprod(A)

```

*Конкатенацией* называют объединение массивов, которое реализует следующая функция.

$C = \text{cat}(\text{dim}, A, B)$  — объединяет массивы A и B в соответствии со спецификацией размерности dim и возвращает объединенный массив;

dim = 1 — горизонтальная конкатенация,

dim = 2 — вертикальная,

dim = 3 — многомерный массив размерности 3 и т. д.;

$C = \text{cat}(\text{dim}, A1, A2, A3, A4, \dots)$  объединяет все входные массивы (A1, A2, A3, A4 и т.д.) в соответствии со спецификацией размерности dim и возвращает объединенный массив;

$\text{cat}(2, A, B)$  — это тоже самое, что и  $[A, B]$ , а  $\text{cat}(1, A, B)$  — тоже самое, что и  $[A; B]$ . При записи  $\text{cat}(\text{dim}, C (:))$  или  $\text{cat}(\text{dim}, C.\text{field})$  эта функция применима к массивам ячеек или структур, содержащим численные матрицы.

### Пример 3.

```

» A = [2.4:3, 5]; B = [8, 7; 9.0]; C = cat(1, A, B)

C =

2     4
3     5
8     7
9     0

```

### Практические задания:

1. Дана матрица размерности  $M(n * m)$ . Возвести ее элементы в степень  $n$ .
2. Даны матрицы  $K(n * m)$  и  $M(n * m)$ . Найти их сумму, разность, произведение и частное от деления.
3. Даны матрицы  $K(n * m)$  и  $M(n * m)$ . Объединить их в матрицу C по строкам и по столбцам.

4. Ввести вещественную матрицу размерности  $m*n$ . Найти минимальный элемент по столбцам, по строкам, общий для матрицы и максимальный элемент по столбцам, по строкам, общий для матрицы.
5. Создать две матрицы A и B одной размерности. Найти их сумму и записать в массив C, а также произведение, деление и объединение.
6. Создать матрицу размерности  $m*n$ . Найти сумму с накоплением и отдельно по столбцам и строкам.
7. Создать матрицу размерности  $m*n$ . Удалить 3-й столбец и 2-ю строку.
8. Создать матрицу размерности  $m*n$ . Найти произведение с накоплением и отдельно по столбцам и строкам.
9. Создать матрицу размерности  $m*n$ . Найти среднее арифметическое всех столбцов и строк.

#### **Варианты заданий**

<b>№ варианта</b>	<b>Значение <math>n</math></b>	<b>Значение <math>m</math></b>
1	4	3
2	5	4
3	6	2
4	7	3
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	3
12	5	6

### **ЛАБОРАТОРНАЯ РАБОТА № 3 «СОЗДАНИЕ МАТРИЦ С ЗАДАНЫМИ СВОЙСТВАМИ»**

**Цель работы:** используя всевозможные методы MatLab, научиться создавать матрицы с заданными свойствами и производить над ними элементарные операции.

**Методические рекомендации:**

Для выполнения данной работы используйте:

1) функции формирования массивов специального вида:

- ZEROS - формирование массива нулей;
- ONES - формирование массива единиц;

- EYE - формирование единичной матрицы;
- RAND - формирование массива элементов, распределенных по равномерному закону;
- RANDN - формирование массива элементов, распределенных по нормальному закону;
- CROSS - векторное произведение.

2) операции над матрицами:

- DIAG - формирование или извлечение диагоналей матрицы;
- TRIL - формирование нижнетреугольной матрицы (массива);
- TRIU - формирование верхнетреугольной матрицы (массива);
- FLIPLR - поворот матрицы относительно вертикальной оси;
- FLIPUD - поворот матрицы относительно горизонтальной оси;
- ROT90 - поворот матрицы на 90 градусов;
- RESHAPE - преобразование размеров матрицы;
- MAGIC - магический квадрат.

Рассмотрим некоторые из них.

**EYE (Создание единичной матрицы)**

eye(n) — возвращает единичную матрицу размера  $n \times n$ ;

eye(m n) или eye([m n]) — возвращают матрицу размера  $m \times n$  с единицами по диагонали и нулями в остальных ячейках;

eye(size(A)) — возвращает единичную матрицу того же размера, что и A.

Единичная матрица не определена для многомерных массивов. Так, функция  $y = \text{eye}([2,3,4])$  при попытке ее вычисления приведет к ошибке.

Пример использования функции eye:

```

>>          5)
t=eye(4)

t =
1         0         0         0         0
0         1         0         0         0
0         0         1         0         0
0         0         0         1         0

```

**DIAG (Формирование или извлечение диагоналей матрицы)**

Синтаксис:

$X = \text{diag}(v)$	$v = \text{diag}(X)$
----------------------	----------------------

$X = \text{diag}(v, k)$	$v = \text{diag}(X, k)$
-------------------------	-------------------------

*Описание:*

Функция  $X = \text{diag}(v)$  формирует квадратную матрицу  $X$  с вектором  $v$  на главной диагонали.

Функция  $X = \text{diag}(v, k)$  формирует квадратную матрицу  $X$  порядка  $\text{length}(v) + \text{abs}(k)$  с вектором  $v$  на  $k$ -й диагонали.

Функция  $v = \text{diag}(X)$  извлекает из матрицы  $X$  главную диагональ.

Функция  $v = \text{diag}(X, k)$  извлекает из матрицы  $X$  диагональ с номером  $k$ ; при  $k > 0$  - это номер  $k$ -й верхней диагонали, при  $k < 0$  - это номер  $k$ -й нижней диагонали.

### ***Пример 1.***

$\text{diag}(\text{diag}(X))$  - диагональная матрица;

$\text{sum}(\text{diag}(X))$  - след матрицы  $X$ .

Оператор  $\text{diag}(-m:m) + \text{diag}(\text{ones}(2*m,1), 1) + \text{diag}(\text{ones}(2*m,1), -1)$  формирует трехдиагональную матрицу размера  $2*m + 1$ .

для  $m = 3$  результирующая матрица имеет вид:  $m = 3$ ;

$$\text{diag}(-m:m) + \text{diag}(\text{ones}(2*m,1), 1) + \text{diag}(\text{ones}(2*m,1), -1)$$

ans =

-3	1	0	0	0	0	0
1	-2	1	0	0	0	0
0	1	-1	1	0	0	0
0	0	1	0	1	0	0
0	0	0	1	1	1	0
0	0	0	0	1	2	1
0	0	0	0	0	1	3

### ***FLIPLR (Поворот массива относительно вертикальной оси)***

*Синтаксис:*

$$B = \text{fliplr}(A)$$

*Описание:*

Функция  $B = \text{fliplr}(A)$  переставляет столбцы массива  $A$  симметрично относительно вертикальной оси. Если массив  $A$  имеет нечетное число столбцов, то средний столбец остается на своем месте.



**Пример 2.**

Для массива  $A = [1\ 2\ 3\ 4; 1\ 2\ 3\ 4; 1\ 2\ 3\ 4]$

```
A =                                fliplr(A) =
1
1
1
1
```

$B = \text{flipud}(A)$  — зеркально переставляет строки матрицы  $A$  относительно горизонтальной оси.

**Пример 3.**

```
F =
3      2  12
6      3   2
» flipud(F)
ans =
6      3   2
3      2  12
```

$\text{perms}(v)$  — возвращает матрицу  $P$ , которая содержит все возможные перестановки элементов вектора  $v$ . Каждая перестановка в отдельной строке. Матрица  $P$  содержит  $n!$  строк и  $n$  столбцов.

**Пример 4.**

```
» v=[1 4 6]
v =
1 4 6
P=perms(v)
6 4 1
4 6 1
6 1 4
```

1	6	4
4	1	6
1	4	6

**Практические задания:**

1. Создать единичную матрицу размерности  $m*n$ .
2. Создать матрицу с единичными элементами размерности  $m*n$ .
3. Создать матрицу с нулевыми элементами размерности  $m*n$ .
4. Создать магическую матрицу размерности  $m$ . Найти сумму по строкам, столбцам и диагонали. Выделить верхнюю и нижнюю часть исходной матрицы.
5. Создать матрицу размерности  $n$  с заданной диагональю.
6. Создать матрицу размерности  $m*n$  и переставить строки, столбцы и произвести всевозможные перестановки.
7. Создать матрицу размерности  $m*n$  и переставить строки, столбцы и произвести всевозможные перестановки. Повернуть исходную матрицу на  $90^\circ$ , на  $180^\circ$ .
8. Создать матрицу размерности  $n$ . Вычислить след матрицы.
9. Создать магическую матрицу размерности  $m$ . Выделить верхнюю и нижнюю часть матрицы.
10. Создать две магические матрицы размерности  $m$  и объединить их. Определить, является ли новая матрица магической.

**Варианты заданий**

№ варианта	Значение $n$	Значение $m$
1	4	3
2	5	4
3	6	2
4	7	3
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	3
12	5	6

## ЛАБОРАТОРНАЯ РАБОТА № 4 «ГРАФИЧЕСКИЕ КОМАНДЫ И ФУНКЦИИ»

**Цель работы:** используя графические команды и функции MatLab, научиться создавать одномерные и двумерные графики и производить над ними операции.

### **Методические рекомендации:**

Элементарные графические функции системы MatLab позволяют построить на экране и вывести на печатающее устройство следующие типы графиков: линейный, логарифмический, полулогарифмический, полярный.

Для каждого графика можно задать заголовок, нанести обозначение осей и масштабную сетку.

### **Двумерные графики:**

- PLOT - график в линейном масштабе;
- LOGLOG - график в логарифмическом масштабе;
- SEMILOGX, SEMILOGY - график в полулогарифмическом масштабе;
- POLAR - график в полярных координатах;
- Linspace - формирование линейного массива равноотстоящих узлов;
- LOGSPACE - формирование узлов логарифмической сетки;
- MESHGRID - формирование узлов двумерной и трехмерной сеток.

### **PLOT (График в линейном масштабе)**

#### **Синтаксис:**

```
plot(y)
plot(x, y)
plot(x, y, s)
plot(x1, y1, s1, x2, y2, s2, ...)
```

#### **Описание:**

Команда plot(y) строит график элементов одномерного массива y в зависимости от номера элемента; если элементы массива y комплексные, то строится график plot(real(y), imag(y)). Если Y - двумерный действительный массив, то строятся графики для столбцов; в случае комплексных элементов их мнимые части игнорируются.

Команда plot(x, y) соответствует построению обычной функции, когда одномерный массив x соответствует значениям аргумента, а одномерный массив y - значениям функции. Когда один из массивов X или Y (либо оба) двумерные, то реализуются следующие построения:

- если массив  $Y$  двумерный, а массив  $x$  одномерный, то строятся графики для столбцов массива  $Y$  в зависимости от элементов вектора  $x$ ;
- если двумерным является массив  $X$ , а массив  $y$  одномерный, то строятся графики столбцов массива  $X$  в зависимости от элементов вектора  $y$ ;
- если оба массива  $X$  и  $Y$  двумерные, то строятся зависимости столбцов массива  $Y$  от столбцов массива  $X$ .

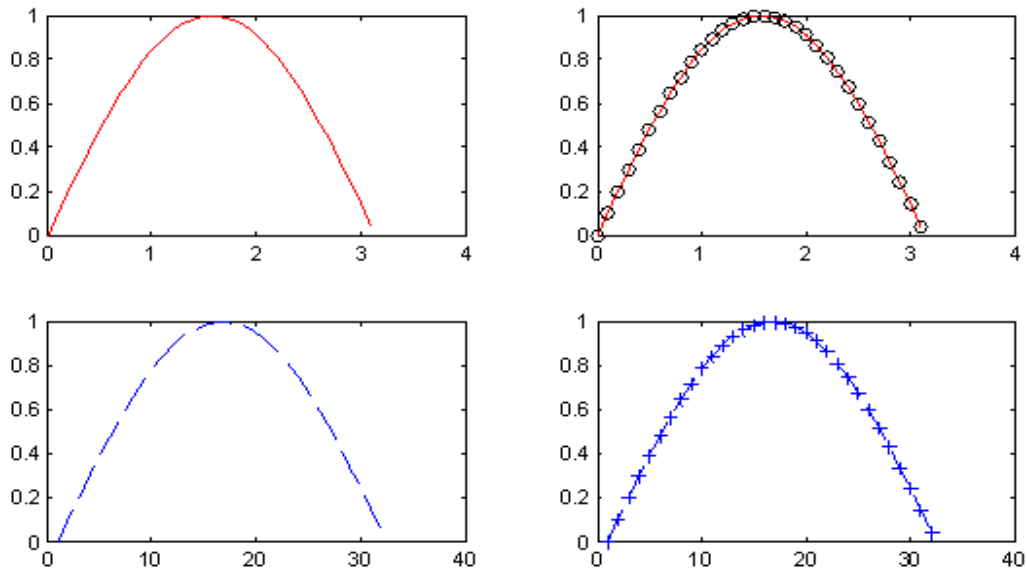
Команда `plot(x,y,s)` позволяет выделить график функции, указав способ отображения линии, способ отображения точек, цвет линий и точек с помощью строковой переменной  $s$ , которая может включать до трех символов из следующей таблицы:

цвет линии	тип линии	тип точек
c голубой	- непрерывная	. точка
m фиолетовый	-- штриховая	+ плюс
y желтый	: пунктирная	* звездочка
r красный	-. штрих-пунктирная	o кружок
g зеленый		x крестик
b синий		
w белый		
k черный		

Если цвет линии не указан, то он выбирается по умолчанию из шести первых цветов, с желтого до синего, повторяясь циклически.

Команда `plot(x1, y1, s1, x2, y2, s2, ...)` позволяет объединить на одном графике несколько функций  $y_1(x_1)$ ,  $y_2(x_2)$ , ..., определив для каждой из них свой способ отображения.

Обращение к командам `plot` вида `plot(x, y, s1, x, y, s2)` позволяет для графика  $y(x)$  определить дополнительные свойства, для указания которых применения одной строковой переменной  $s1$  недостаточно, например при задании разных цветов для линии и для точек на ней.



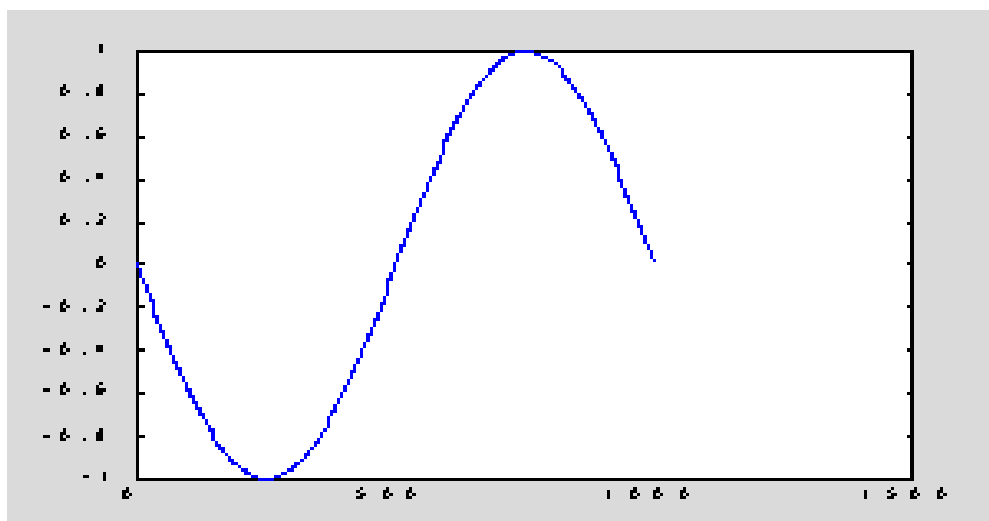
*Примеры отображения графиков с разными типами маркеров*

**Примеры:**

Построим график функции  $y = \sin(x)$  на отрезке  $[-\pi \pi]$  с шагом  $\pi/500$ :

```
x = -pi:pi/500:pi;
y = sin(x);
plot(y)      % рис. а
plot(x, y)   % рис. б
```

- график (рис. А) отображает значения одномерного массива  $y$ , состоящего из 1001 элемента как функцию от номера элемента;



*Рисунок А. Массив как функция от номера элемента*

- график (рис. Б) отображает значения того же массива как функцию элементов массива  $x$ .

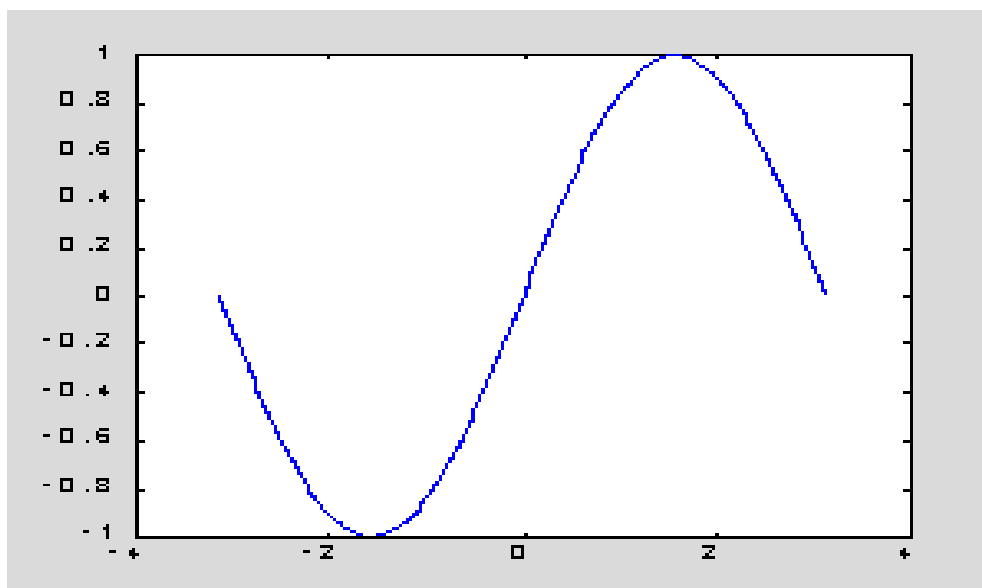


Рисунок Б. Массив как функция элементов массива

Рассмотрим различные способы применения функции  $\text{plot}(x, y)$  на примере графиков двух функций  $y_1 = \sin(x)$  и  $y_2 = x\sin(x)$ :

```
x1 = -pi:pi/500:pi;  
y1 = sin(x1);  
y2 = x1.*sin(x1);  
plot(x1', [y1' y2'])      % рис. в  
plot( [y1' y2'], x1')    % рис. г
```

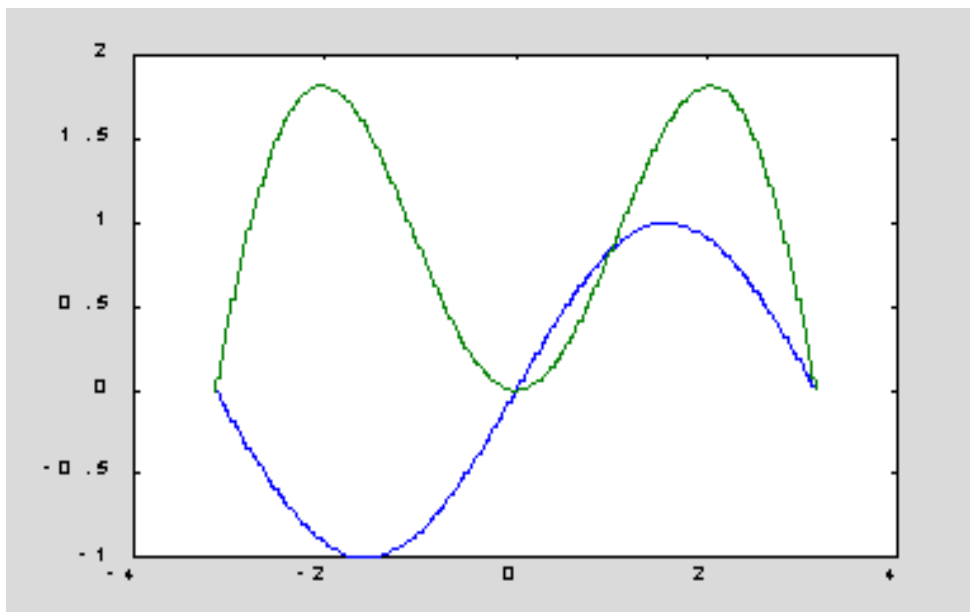
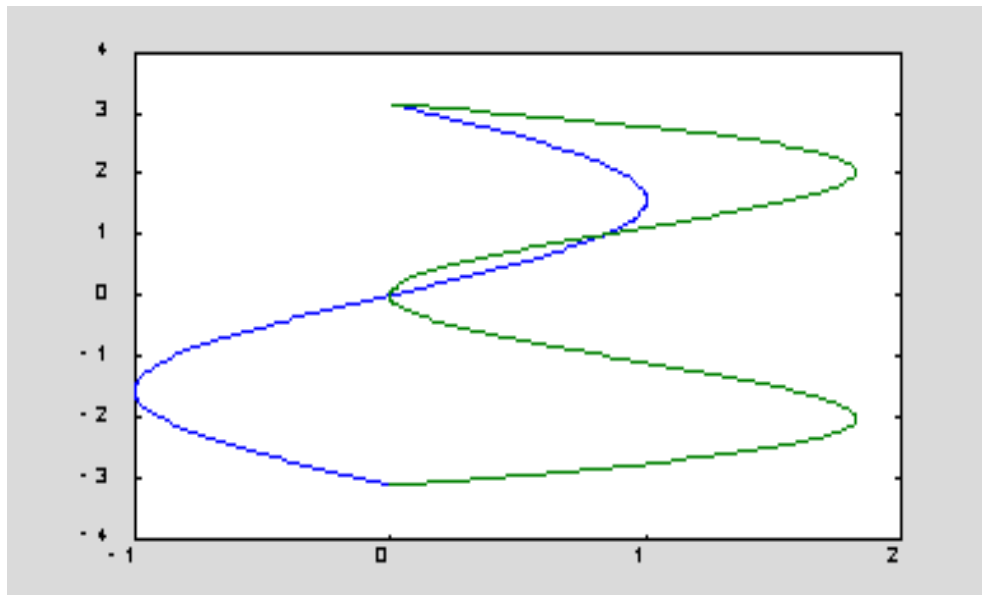
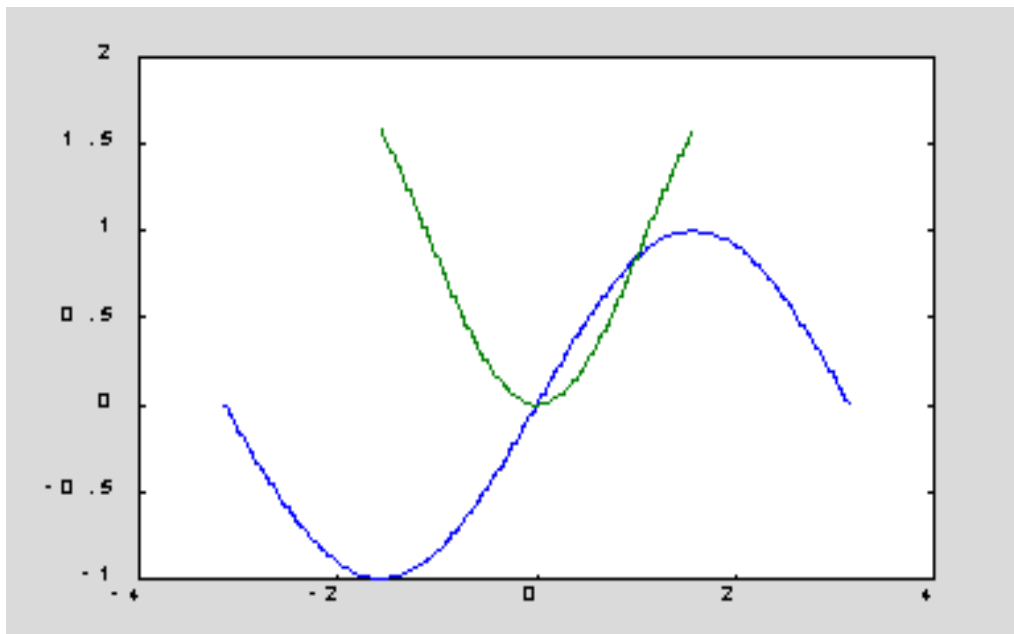


Рисунок В. Пример отображения графиков двух функций



*Рисунок Г. Пример отображения графиков двух функций*

```
x2 = x1/2;
y2 = x2.*sin(x2);
plot([x1' x2'], [y1'
y2']) % рис. д
```



*Рисунок Д. Пример отображения графиков двух функций*

***POLAR (График в полярных координатах)***

*Синтаксис:*

```
polar(phi, rho)
polar(phi, rho, s)
```

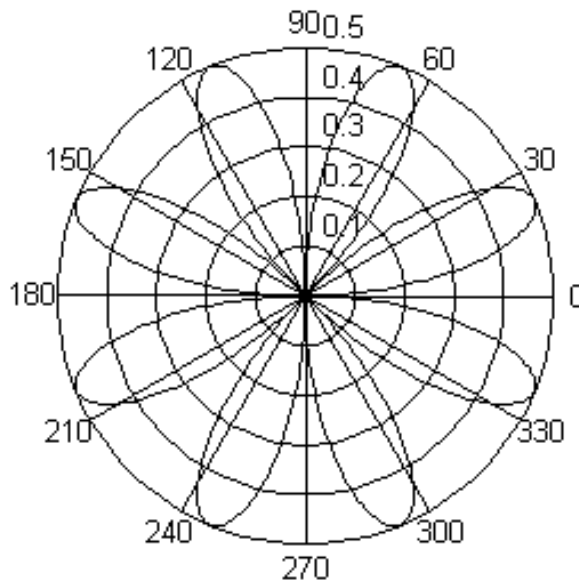
*Описание:*

Команды `polar(...)` реализуют построение графиков в полярных координатах, задаваемых углом `phi` и радиусом `rho`.

*Примеры:*

Построим график функции  $\rho = \sin(2 * \text{phi}) * \cos(2 * \text{phi})$  в полярных координатах

```
phi = 0:0.01:2 * pi;
polar(phi, sin(2 * phi) * cos(2 * phi))
```



### **Столбцовые диаграммы (гистограммы)**

Столбцовые диаграммы широко используются в литературе, посвященной финансам и экономике, а также в математической литературе. Ниже представлены команды для построения таких диаграмм.

– `bar(x,Y)` — строит столбцовый график элементов вектора `Y` (или группы столбцов для матрицы `Y`) со спецификацией положения столбцов, заданной значениями элементов вектора `x`, которые должны идти в монотонно возрастающем порядке;

– `bar(Y)` — строит график значений элементов матрицы `Y` так же, как указано выше, но фактически для построения графика используется вектор `x=1:m`;

Возможно применение этих команд и в следующем виде:

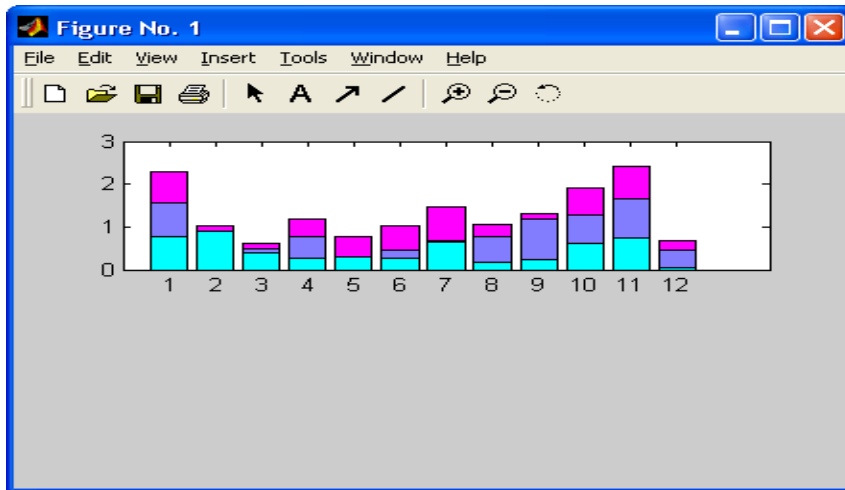


- `bar(... 'Спецификация')` для задания спецификации графиков, например типа линий, цвета и т. д., по аналогии с командой `plot`.

- спецификация `'stacked'` задает рисование всех  $n$  столбцов в позиции  $m$  друг на друге.

Пример построения столбцовой диаграммы матрицы размером  $12 \times 3$  приводится ниже:

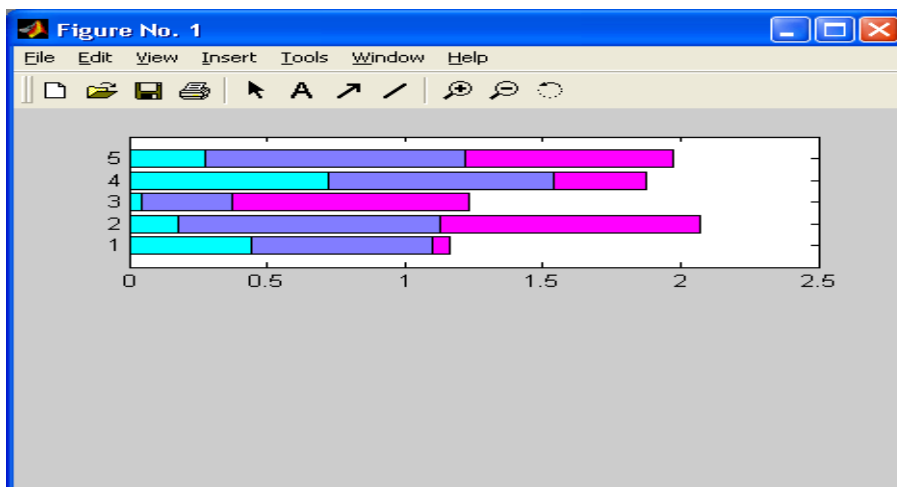
```
» % Столбцовая диаграмма с вертикальными столбцами  
» subplot(2.1.1). bar(rand(12.3). 'stacked'), colormap(cool)
```



#### *Пример построения диаграммы с вертикальными столбцами*

Помимо команды `bar(...)` существует аналогичная ей по синтаксису команда `barh(...)`, которая строит столбцовые диаграммы с горизонтальным расположением столбцов.

```
» subplot(2.1.1). barh(rand(5.3). 'stacked'), colormap(cool)
```



#### *Пример построения столбцовой диаграммы с горизонтальными столбцами*

Выбор расположения столбцов зависит от пользователя, использующего эти команды для представления своих данных.

**Практические задания:**

1. Построить график функции  $\cos 3x$  на интервале от 0 до 15 с шагом 0.2. Изменить цвет полученного графика на красный.
2. Построить график функций  $\sin 2x$  и  $\cos 3x$  на интервале от 0 до 10 с шагом 0.1. Изменить цвет полученного графика  $\sin 2x$  на красный и тип линии – пунктирный, а цвет полученного графика  $\cos 3x$  изменить на черный и тип линии – двойной пунктир с крестиком.
3. Построить график функций  $y = \sqrt{x^2 + 2} \sin x$  и  $y = \cos(x\varepsilon^{\frac{1}{5}}) \sin x$  на интервале от -15 до 15 с шагом 0.1. Изменить цвет полученных графиков и тип линий, подписать каждый график.
4. В полярных координатах построить график функции  $r = \sin(3\varphi)$ , где  $\varphi$  изменяется от 0 до  $2\pi$  с шагом 0.01.
5. В полярных координатах построить графики функций  $y = x^2 \sin x$  и  $y = x \cos 2x$ . Изменить цвет полученных графиков и подписать каждый график.
6. Построить график функции  $y = \frac{\varepsilon^x}{x}$  в логарифмическом масштабе на интервале от -1 до 3.
7. Построить график функции  $y = \varepsilon^x$  в полулогарифмическом масштабе на интервале от 0 до 10 с шагом 0.5.
8. Создать вектор  $V$  и построить гистограмму.
9. Построить лестничный график функции  $y = x^2$  на интервале от 0 до 10 с шагом 0.25.
10. Построить график дискретных отсчетов функции  $y = \sin(x^2)\varepsilon^{-x}$  на интервале от 0 до 4 с шагом 0.1.

**Варианты заданий**

№ варианта	Значение $n$	Значение $m$
1	4	6
2	5	7
3	6	2
4	7	5
5	7	2
6	8	2
7	4	3
8	5	3

9	3	6
10	4	5
11	9	7
12	5	9

## ЛАБОРАТОРНАЯ РАБОТА № 5 «КОМАНДЫ И ФУНКЦИИ ТРЕХМЕРНОЙ ГРАФИКИ»

**Цель работы:** используя графические команды и функции MatLab, научиться создавать трехмерные графики и производить над ними операции.

### **Методические рекомендации:**

#### **Трехмерные графики**

В системе MatLab предусмотрено несколько команд и функций для построения трехмерных графиков. Значения элементов числового массива рассматриваются как z-координаты точек над плоскостью, определяемой координатами x и y. Возможно несколько способов соединения этих точек. Первый из них - это соединение точек в сечении (функция plot3), второй - построение сетчатых поверхностей (функции mesh и surf). Поверхность, построенная с помощью функции mesh, - это сетчатая поверхность, ячейки которой имеют цвет фона, а их границы могут иметь цвет, который определяется свойством EdgeColor графического объекта surface.

Поверхность, построенная с помощью функции surf, - это сетчатая поверхность, у которой может быть задан цвет не только границы, но и ячейки; последнее управляется свойством FaceColor графического объекта surface.

#### **Возможные операции:**

- PLOT3 - построение линий и точек в трехмерном пространстве;
- MESHGRID - формирование двумерных массивов X и Y;
- MESH, MESHС, MESHZ - трехмерная сетчатая поверхность;
- SURF, SURFC - затененная сетчатая поверхность;
- SURFL - затененная поверхность с подсветкой;
- AXIS - масштабирование осей и вывод на экран;
- GRID - нанесение сетки;
- HOLD - управление режимом сохранения текущего графического окна;
- SUBPLOT - разбиение графического окна;
- ZOOM - управление масштабом графика;
- COLORMAP - палитра цветов;

- CAXIS - установление соответствия между палитрой цветов и масштабированием осей;
- SHADING - затенение поверхностей;
- CONTOURC - формирование массива описания линий уровня;
- CONTOUR - изображение линий уровня для трехмерной поверхности;
- CONTOUR3 - изображение трехмерных линий уровня.

*Надписи и пояснения к графикам:*

- TITLE - заголовки для двух- и трехмерных графиков;
- XLABEL, YLABEL, ZLABEL - обозначение осей;
- CLABEL - маркировка линий уровня;
- TEXT - добавление к текущему графику текста;
- GTEXT - размещает заданный текст на графике с использованием мыши;
- LEGEND - пояснение к графику;
- COLORBAR - шкала палитры.

### ***PLOT3 (Построение линий и точек в трехмерном пространстве)***

*Синтаксис:*

```
plot3(x, y, z)
plot3(X, Y, Z)
plot3(x, y, z, s)
plot3(x1, y1, z1, s1,
x2, y2, z2, s2, ...)
```

*Описание:*

Команды plot3(...) являются трехмерными аналогами функции plot(...).

Команда plot3(x,y,z), где x, y, z - одномерные массивы одинакового размера, строит точки с координатами x(i), y(i), z(i) и соединяет их прямыми линиями.

Команда plot3(X,Y,Z), где X, Y, Z - двумерные массивы одинакового размера, строит точки с координатами x(i, :), y(i, :), z(i, :) для каждого столбца и соединяет их прямыми линиями.

Команда plot3(x,y,z,s) позволяет выделить график функции z(x,y), указав способ отображения линии, способ отображения точек, цвет линий и точек с помощью строковой переменной s, которая может включать до трех символов из следующей таблицы.

цвет линии	тип линии	тип точек
c голубой	- непрерывная	. точка

m	фиолетовый	--	штриховая	+	плюс
y	желтый	:	пунктирная	*	звездочка
r	красный	-.	штрих- пунктирная	o	кружок
g	зеленый			x	крестик
b	синий				
w	белый				
k	черный				

Если цвет линии не указан, то он выбирается по умолчанию из шести первых цветов, с желтого до синего, повторяясь циклически.

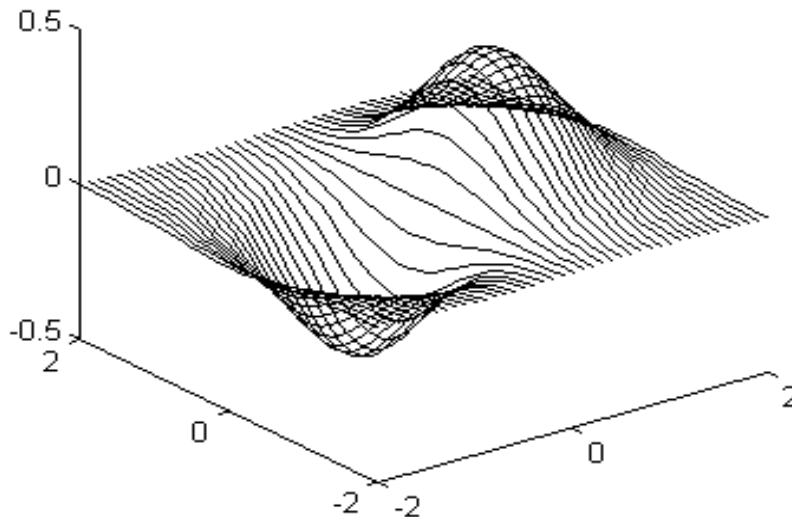
Команда `plot3(x1,y1,z1,s1,x2,y2,z2,s2, ...)` позволяет объединить на одном графике несколько функций  $z_1(x_1,y_1)$ ,  $z_2(x_2,y_2)$ , ..., определив для каждой из них свой способ отображения.

Обращение к команде `plot3` вида `plot3(x,y,z,s1,x,y,z,s2)` позволяет для графика  $z(x,y)$  определить дополнительные свойства, для указания которых применения одной строковой переменной `s1` недостаточно, например при задании разных цветов для линии и для точек на ней.

***Примеры:***

Построим график функции  $z = x \cdot \exp(-x^2 - y^2)$  в трехмерном пространстве.

```
[ X, Y ] = meshgrid([ -2 : 0.1 : 2 ]);
Z = X .* exp(- X .^ 2 - Y .^ 2);
plot3(X, Y, Z)
```



### **MESHGRID (Формирование узлов двумерной и трехмерной сеток)**

*Синтаксис:*

```
[X, Y] = meshgrid(x, y)
[X, Y] = meshgrid(x)
[X, Y, Z] = meshgrid(x, y, z)
```

*Описание:*

Функция  $[X, Y] = \text{meshgrid}(x, y)$  формирует массивы  $X$  и  $Y$ , которые определяют координаты узлов прямоугольника, задаваемого векторами  $x$  и  $y$ . Этот прямоугольник задает область определения функции от двух переменных, которую можно построить в виде 3D-поверхности.

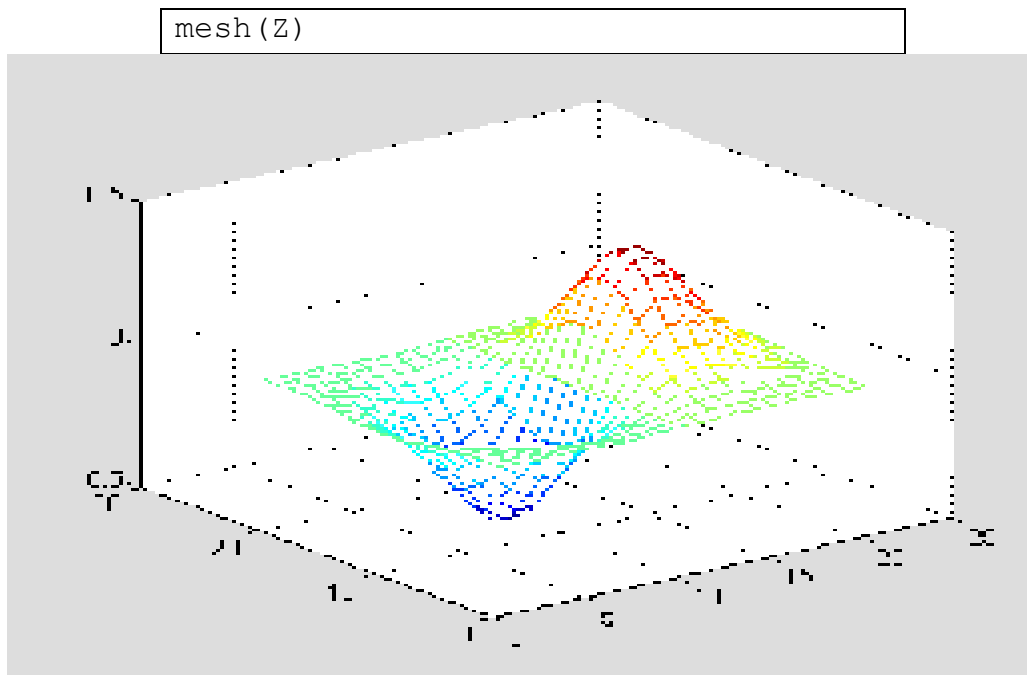
Функция  $[X, Y] = \text{meshgrid}(x)$  является сокращенной формой записи функции  $[X, Y] = \text{meshgrid}(x, x)$ .

Функция  $[X, Y, Z] = \text{meshgrid}(x, y, z)$  формирует массивы  $X$ ,  $Y$  и  $Z$ , которые определяют координаты узлов параллелепипеда, задаваемого векторами  $x$ ,  $y$  и  $z$ . Этот параллелепипед задает область определения для вычисления функции от трех переменных и построения 3D-параметрических поверхностей.

**Пример:**

Для вычисления функции в области  $-2 < x < 2$ ,  $-2 < y < 2$  необходимо выполнить следующую последовательность операций:

```
[X, Y] = meshgrid(-2:.2:2, -2:.2:2);
Z = X.*exp(-X.^2 - Y.^2);
```



*График поверхности, объединенный с контурным графиком ее проекции*

Иногда график поверхности полезно объединить с контурным графиком ее проекции на плоскость, расположенным под поверхностью.

Для этого используется команда `meshc`:

`meshc(...)` — аналогична `mesh(...)`, но помимо графика поверхности дает изображение ее проекции в виде линий равного уровня (графика типа `contour`).

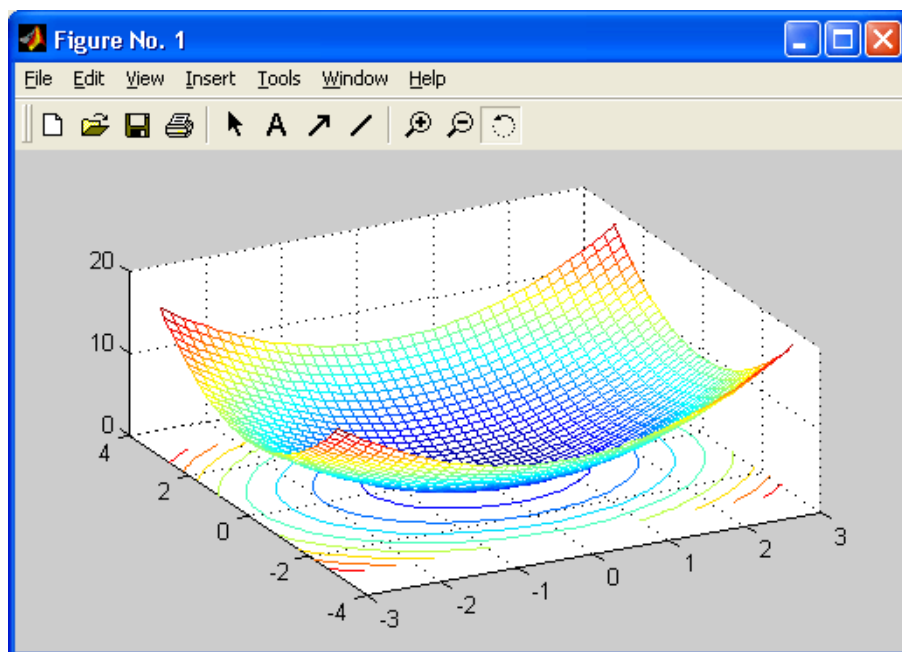
Ниже даны два примера применения этой команды:

***Пример 1.***

```

» [X,Y]=meshgrid([-3:0.15:3]);
» Z=X.^2+Y.^2;
» meshc(X,Y,Z)

```



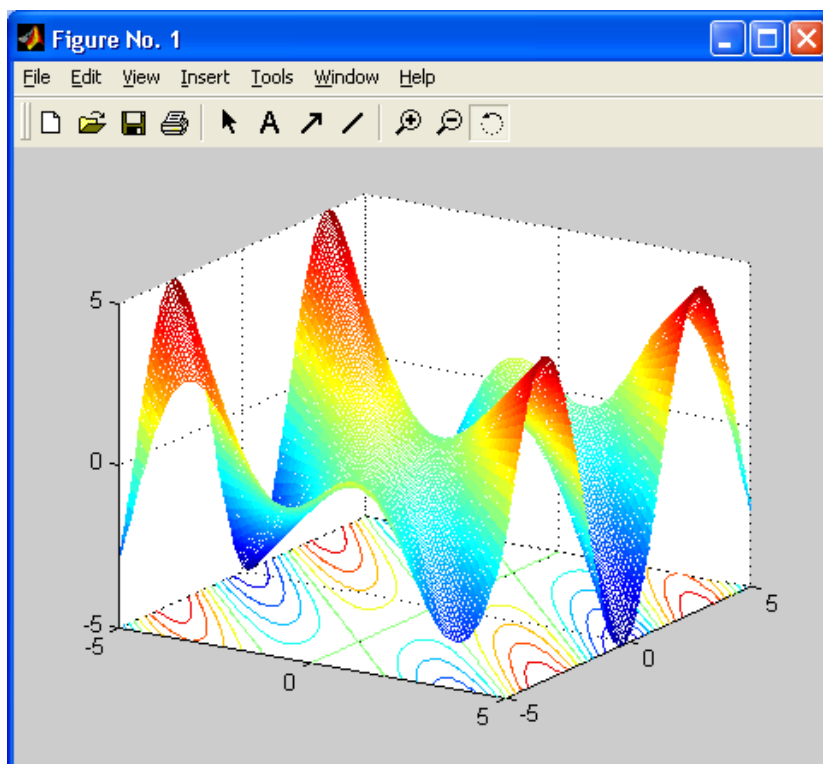
*График поверхности и ее проекции на расположенную ниже плоскость*

***Пример 2.***

```

» [X,Y]=meshgrid(-5:0.1:5);
» Z=X.*sin(X+Y);
» meshc(X,Y,Z)

```



*Окно с графиками поверхности и ее проекции на плоскость под фигурой*



## ***SUBPLOT (Разбиение графического окна)***

*Синтаксис:*

```
subplot(m,n,p)
subplot(h)
subplot(mnp)
```

*Описание:*

Данная команда выполняется перед обращением к функциям построения графиков для одновременной выдачи нескольких графиков в различных частях графического окна.

Команды `subplot(mnp)` или `subplot(m,n,p)`, где `m,n,p` - 3 цифры, производит разбику графического окна на несколько подокон, создавая при этом новые объекты `axes`; значение `m` указывает, на сколько частей разбивается окно по горизонтали, `n` - по вертикали, а `p` - номер подокна, куда будет выводиться очередной график. Эти же команды могут использоваться для перехода от одного подокна к другому.

Команда `subplot(h)`, где `h` - дескриптор для объекта `axes` соответствующего подокна, - другой способ выбора подокна для размещения графика.

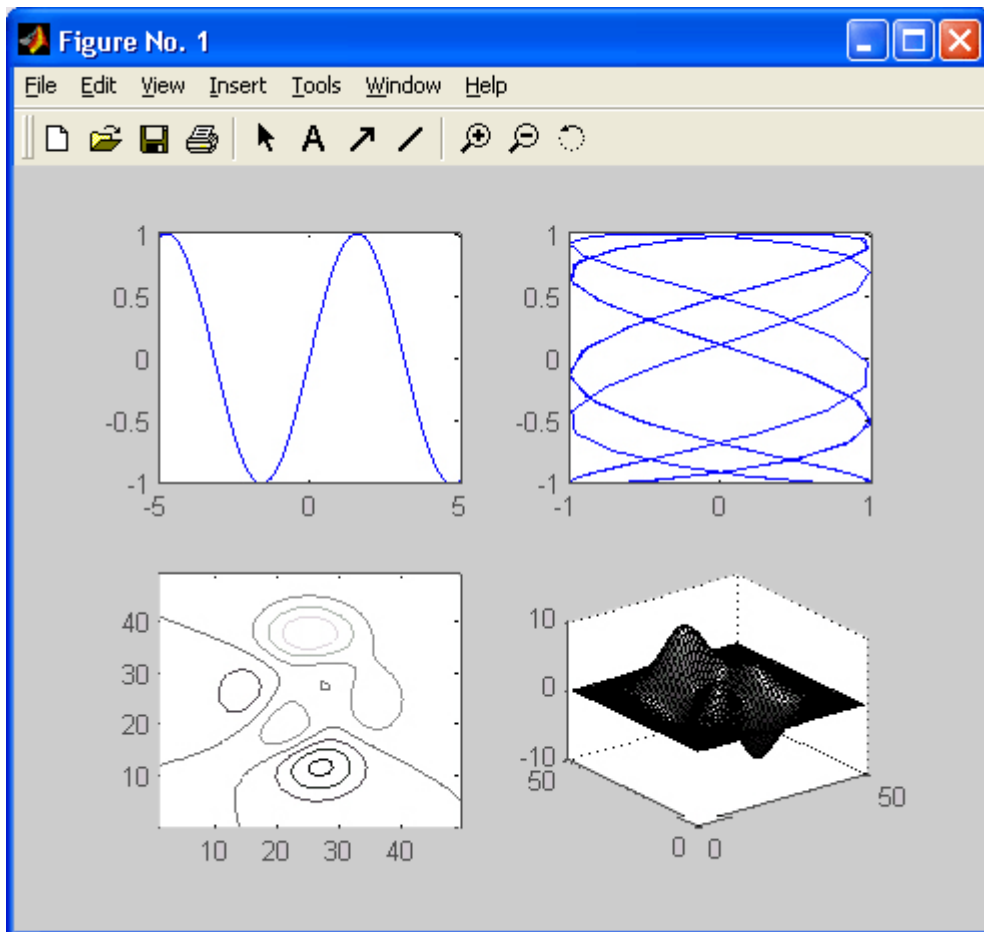
Команды `clf`, `subplot(111)`, `subplot(1,1,1)` выполняют одну и ту же функцию - удаляют все подокна и возвращают графическое окно в штатное состояние.

### ***Пример 1.***

Следующий пример иллюстрирует применение команды `subplot`:

```
» x=-5:0.1:5;
subplot(2,2,1).plot(x.sin(x))
subplot(2,2,2).plot(sin(5*x).cos(2*x+0.2))
subplot(2,2,3).contour(peaks)
subplot(2,2,4).surf(peaks)
```

В этом примере последовательно строятся четыре графика различного типа, размещаемых в разных подокнах.



*Пример размещения нескольких графиков одновременно*

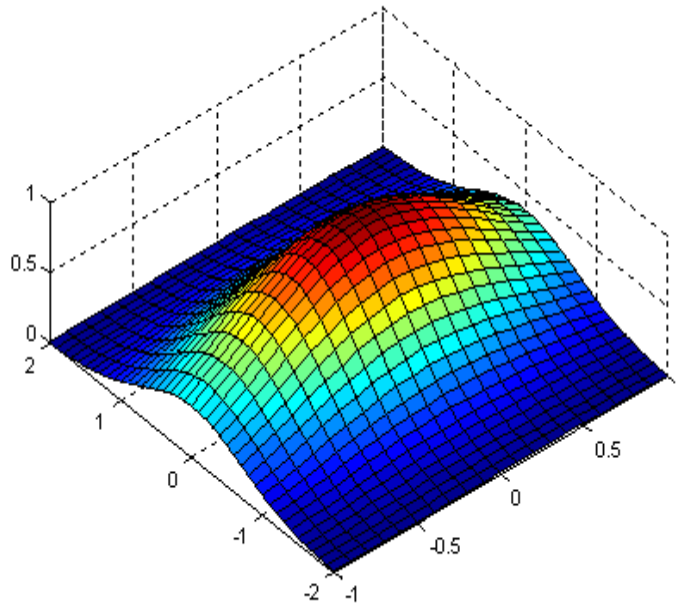
**Пример 2.**

В верхней части экрана строится функция  $y_1 = \sin(x)$ , в нижней -  $y_2 = \log(\text{abs}(y))$ .

```
x = -1:.1:1;
y1 = sin(x);
subplot(2, 1, 1), plot(x, y1)
y2 = log(abs(y1));
subplot(2, 1, 2), plot(x, y2)
```

В системе MatLab предусмотрена функция визуализации непрерывной поверхности в трехмерных осях.

```
surf(X,Y,Z);           % отображение непрерывной поверхности
```



*Результат работы функции surf()*

***Практические задания:***

1. Построить график функции  $z = xe^{-x^2-y^2}$  в трехмерном пространстве. X и Y изменяются на интервале от -2 до 2 с шагом 0.1.
2. Определить двумерные массивы и вычислите функцию  $z = xe^{-x^2-y^2}$ , заданную на квадрате  $-2 \leq x \leq 2, -2 \leq y \leq 2$ .
3. Построить график функции  $z=x^2 + y^2$  на интервале от -3 до 3 с шагом 0.15. Изменить показатель степени на 3,4,5. Проанализировать, как изменяется график. Изменить тип линий на нолики и затем на крестики.
4. Построить график функций  $z=\sin x^2 + \cos y^3$  на интервале от -2 до 2 с шагом 0.1. Использовать для построения графика, поочередно, функции meshc, meshz, surf.
5. Построить график функций  $z=2\cos(x+y)+y\cos(x-y)$  на интервале: x от 0 до 8 с шагом 0.2, y от 0 до 4 с шагом 0.2. Использовать для построения графика, поочередно, функции meshc, meshz, surf в разных подокнах, подписать графики.

***Варианты заданий***

<b>№ варианта</b>	<b>Значение <math>n</math></b>	<b>Значение <math>m</math></b>
1	4	6
2	5	7
3	6	2

4	7	5
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	7
12	5	9

**ЛАБОРАТОРНАЯ РАБОТА № 6**  
**«ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ В MATLAB.**  
**ПРОГРАММЫ ЛИНЕЙНОЙ И РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ»**

**Цель работы:** используя команды и функции MatLab, научиться создавать элементарные программы линейной и разветвляющейся структуры.

**Методические рекомендации:**

**Сервисные функции:**

- *disp* – осуществляет вывод текста или значения переменной в командное окно;
- *input* – осуществляет запрос на ввод с клавиатуры. Используется при создании приложений с интерфейсом из командной строки;
- *eval* – выполняет содержимое строки или строковой переменной, как команды MatLab;
- *clear* – удаляет переменные рабочей среды;
- *clc* – производит очистку командного окна.

Более подробную информацию об этих и других функциях можно узнать, выполнив в командной строке

```
help имя_функции
```

**Операторы ветвления**

**Условный оператор *if***

**Синтаксис:**

```
if условие
команды MATLAB
end
```

**Описание:**

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

В таблице 1 представлены варианты простых логических выражений оператора if.

*Таблица 1. Простые логические выражения*

if a < b	Истинно, если переменная a меньше переменной b и ложно в противном случае.
if a > b	Истинно, если переменная a больше переменной b и ложно в противном случае.
if a == b	Истинно, если переменная a равна переменной b и ложно в противном случае.
if a <= b	Истинно, если переменная a меньше либо равна переменной b и ложно в противном случае.
if a >= b	Истинно, если переменная a больше либо равна переменной b и ложно в противном случае.
if a ~= b	Истинно, если переменная a не равна переменной b и ложно в противном случае.

Ниже представлен пример реализации функции sign(), которая возвращает +1, если число больше нуля, -1 – если число меньше нуля и 0, если число равно нулю:

```
function
my_sign
x = 5;
if x > 0
disp(1);
end
if x < 0
disp(-1);
end
if x == 0
disp(0);
end
```

## ***Условный оператор elseif***

***Синтаксис:***

```
if условие1
команды1
else if условие2
команды2
.....
else if условие
n
команды n
else
команды
end
```

***Описание:***

В зависимости от выполнения того или иного условия работает соответствующая ветвь программы, если все условия неверны, то выполняются команды, размещенные после else.

***Пример:***

Тогда приведенный выше пример можно записать следующим образом:

```
function
my_sign
x = 5;
if x > 0
disp(1);
else
if x < 0
disp(-1);
else
disp(0);
end
end
```

## ***Оператор switch***

***Синтаксис:***

```
switch
переменная
case значение1
команды1
case значение2
команды2
.....
```

```
case значение n
команды n
otherwise
команды
end
```

Каждая ветвь определяется оператором case, переход в нее выполняется тогда, когда переменная оператора switch принимает значение, указанное после case, или одно из значение из списка case. После выполнения какой-либо из ветвей происходит выход из switch, при этом значения, заданные в других case, уже не проверяются. Если подходящих значений для переменной не нашлось, то выполняется ветвь программы, соответствующая otherwise.

Приведем *пример* работы данного оператора для преобразования малых букв латинского алфавита в заглавные.

```
function upper_symbol
ch='c';
switch ch
    case 'a', ch='A';
    case 'b', ch='B';
    case 'c', ch='C';
    case 'd', ch='D';
    case 'e', ch='E';
    ...
    case 'z', ch='Z';
end
disp(ch);
```

В данной программе задается символьная переменная ch со значением c. Затем, с помощью оператора switch проверяется ее значение со всеми возможными малыми буквами латинского алфавита от a до z. Как только одно из условий сработало, оператор switch завершает свою работу и выполнение программы переходит на функцию disp(), которая отображает значение переменной ch на экран.

### ***Практические задания:***

1. Вычислить поверхность шара по формуле  $S=4\pi R^2$ . Величина радиуса вводится с клавиатуры.
2. Вычислить площадь трапеции по формуле  $S=1/2(a+b)h$ . Значения сторон и высоту вводить с клавиатуры.

3. Определить объем шарового сектора  $V = \frac{2}{3}\pi R^2 h$ . Радиус не должен превышать значения 20, а высота – 15.
4. Вычислить значение функции

$$V = \begin{cases} c \sin^2 x & \text{при } x > 0.7 \\ c \cos x + c \sin x & \text{при } x = 0.7 \\ C \sin x & \text{при } x < 0.7 \end{cases}$$

значения  $x$  вводить с клавиатуры.

5. Вычислить значение функции

$$Z = \begin{cases} a/t + bt^2 + c & \text{при } t < 6 \\ t & \text{при } 4 < t < 6 \\ at + bt^3 & \text{при } t > 6 \end{cases}$$

значения  $t$  вводить с клавиатуры.

Все полученные результаты вывести графически и подписать.

### Варианты заданий

№ варианта	Значение $n$	Значение $m$
1	4	6
2	5	7
3	6	2
4	7	5
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	7
12	5	9

## ЛАБОРАТОРНАЯ РАБОТА № 7 «ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ В МАТЛАВ. ПРОГРАММЫ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ»

**Цель работы:** используя команды и функции MatLab, научиться создавать элементарные программы циклической структуры.

**Методические рекомендации:**

**Операторы цикла**



## **Цикл for**

*Синтаксис:*

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

*Описание:*

count – переменная цикла;

start – ее начальное значение;

final – ее конечное значение;

step – шаг, на который увеличивается count при каждом следующем заходе в цикл.

Цикл заканчивается, как только значение count становится больше final.

### **Пример 1.**

Пусть требуется вывести семейство кривых для  $x \in [0, 2\pi]$ , которое задано функцией, зависящей от параметра

$y(x, a) = e^{-ax} \sin x$ , для значений параметра  $a$  от -0.1 до 0.1.

Ниже приведен листинг файл-программы для вывода семейства кривых.

Листинг программы

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

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

### **Пример 2.**

Рассмотрим работу данного цикла на примере реализации алгоритма поиска максимального значения элемента в векторе:

<pre>function search_max a = [3 6 5 3 6 9 5 3 1 0]; m = a(1); for i=1:length(a)  if m &lt; a(i)</pre>	<pre>% текущее максимальное значение % цикл от 1 до конца вектора с % шагом 1 (по умолчанию) % если a(i) &gt; m,</pre>
---	--

<pre>m = a(i);     end end disp(m);</pre>	<pre>% то m = a(i) % конец цикла for</pre>
---	--

В данном примере цикл for задает счетчик  $i$  и меняет его значение от 1 до 10 с шагом 1. Обратите внимание, что если величина шага не указывается явно, то он берется по умолчанию равным 1.

**Практические задания:**

1. Вычислить значение функции

$$W = \begin{cases} ax^2 + bx + c & \text{при } x < 1.2 \\ a/x + \sqrt{(x^2 + 1)} & \text{при } x = 1.2 \\ (a + bx)/\sqrt{(x^2 + 1)} & \text{при } x > 1.2 \end{cases}$$

$x$  изменяется от 1 до 2 с шагом 0.05

2. Вычислить значение функции

$$Y = \begin{cases} at^2 \ln t & \text{при } 1 < t < 2 \\ 1 & \text{при } t < 1 \\ e^{at} \cos bt & \text{при } t > 2 \end{cases}$$

$t$  изменяется от 0 до 3 с шагом 0.15

3. Вывести семейство кривых для  $x \in [0, 2\pi]$ , которое задано функцией, зависящей от параметра  $y(x, a) = e^{-ax} \sin x$ , значения параметра  $a$  изменяются от -0.1 до 0.1.

Все полученные результаты вывести графически и подписать.

**Варианты заданий**

№ варианта	Значение $n$	Значение $m$
1	4	6
2	5	7
3	6	2
4	7	5
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	7
12	5	9

## ЛАБОРАТОРНАЯ РАБОТА № 8 «ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ В МАТЛАВ. ПРОГРАММЫ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ»

**Цель работы:** используя команды и функции MatLab, научиться создавать элементарные программы циклической структуры.

**Методические рекомендации:**

**Цикл while**

**Синтаксис:**

```
while условие цикла
команды MATLAB
end
```

**Описание:**

Цикл работает, пока выполняется (истинно) условие цикла. Для задания условия выполнения цикла допустимы следующие операции отношения:

Обозначение	Операция отношения
==	Равенство
<	Меньше
<=	Меньше или равно
>=	Больше или равно
>	Больше
~=	Не равно

Задание более сложных условий производится с применением логических операторов. Логические операторы приведены в следующей таблице:

Оператор	Условие	Эквивалентная запись
Логическое «и»	$x < 3$ и $k = 4$	$(x < 3) \& (k == 4)$
Логическое «или»	$x = 1$ или $x = 2$	$(x == 1)   (x == 2)$
Отрицание «не»	$a \neq 1.9$	$\sim(a == 1.9)$

Приведем *пример* работы цикла while для подсчета суммы ряда:

$$S = \sum_{i=1}^{20} i$$

```
function sum_i
```

<code>S = 0;</code>	<code>% начальное значение суммы</code>
<code>i=1;</code>	<code>% счетчик суммы</code>
<code>while i &lt;= 20</code>	<code>% цикл (работает пока i &lt;= 20)</code>
<code>S=S+i;</code>	<code>% подсчитывается сумма</code>
<code>i=i+1;</code>	<code>% увеличивается счетчик на 1</code>
<code>end</code>	<code>% конец цикла</code>
<code>disp(S);</code>	<code>% отображение суммы 210 на экране</code>

***Практические задания:***

1. Составить программу, по которой компьютер выводит заданное слово N раз.
2. Составить программу, по которой компьютер многократно вычисляет сумму  $A + B$  при различных значениях  $A$  и  $B$ . В конце каждого этапа появляется запрос о продолжении или прекращении вычислений: «Завершить программу? (Д/Н)».
3. Вычислить значения функции  $y=ax^2+bx+c$  при  $x = -5, -4, \dots, 5$ . Значения  $a, b$  и  $c$  вводятся с клавиатуры.
4. Составить программу, по которой компьютер генерирует последовательность из 10 случайных чисел и подсчитывает сумму четных чисел.

***Варианты заданий***

<b>№ варианта</b>	<b>Значение <math>n</math></b>	<b>Значение <math>m</math></b>
1	4	6
2	5	7
3	6	2
4	7	5
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	7
12	5	9

**ЛАБОРАТОРНА РАБОТА № 9**  
**«ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ В МАТЛАВ.»**

## ПРОГРАММЫ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ»

**Цель работы:** используя команды и функции MatLab, научиться создавать элементарные программы циклической структуры.

### **Методические рекомендации:**

Все необходимое подробно изложено в лабораторной работе № 7 и № 8.

### **Практические задания:**

1. В военкомате набирали призывников в танковые войска, необходимо было набрать 10 человек. Известно, что тем, у кого рост выше 1м 70см в танке тесно и то, что в день военкомат может принять не более 100 человек. Составить программу, помогающую производить отбор.
2. Вычислить значения функции  $y = a / x + b$  при  $x=0, 1, 2, 3, \dots, 10$ . Значения  $a$  и  $b$  вводятся с клавиатуры.
3. Составить программу, по которой компьютер печатает последовательность вида (значение переменной  $n$  задается с клавиатуры):
  - a) 1, 2, 3, 4, ...,  $n$ ;
  - b) 2, 4, 6, 8, ...,  $2n$ ;
  - c) 10, 100, 1000, ...,  $10n$   
(  $n < 10$ );
  - d) 1, 3, 9, 27, 81, ...,  $3^n$

### **Варианты заданий**

№ варианта	Значение $n$	Значение $a$	Значение $b$
1	4	6	2
2	5	7	3
3	6	2	5
4	7	5	7
5	7	2	4
6	8	2	6
7	4	3	8
8	5	3	2
9	3	6	3
10	4	5	2
11	9	7	3
12	5	9	4

## ЛАБОРАТОРНАЯ РАБОТА № 10

## «ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ В МАТЛАВ. ВЫЧИСЛЕНИЕ КОРНЕЙ ПОЛИНОМОВ»

**Цель работы:** используя команды и функции MatLab, научиться вычислять корни полиномов.

### **Методические рекомендации:**

*Полиномы и операции над ними:*

- POLYVAL - вычисление полинома;
- POLYVALM - вычисление матричного полинома;
- CONV - умножение полиномов;
- DECONV - деление полиномов;
- POLYDER - вычисление производных;
- ROOTS - вычисление корней полиномов;
- POLY - вычисление характеристического полинома;
- RESIDUE, RESI2 - разложение на простые дроби.

Приведем несколько примеров по применению этих команд.

### **Пример 1.**

Построим график полинома  $p(x)=x^3-x+2$  на отрезке  $-1 \leq x \leq 1$ . Это выглядит так:

```
p=[1,0,-1,2]; x=-1:.01:1; f=polyval(p,x); plot(x,f), grid
```

Полином не имеет корней на заданном отрезке. Это подтверждает и команда `roots(p)`, которая даст

```
ans = -1.5214 0.7607 - 0.8579i 0.7607 + 0.8579i.
```

### **Пример 2.**

Разделим предыдущий полином на  $x-3$ :

```
[q,r]=deconv(p,[1,-3])
```

Тогда

```
q = 1 3 8, r = 0 0 0 26.
```

Другими словами, частное  $q(x)=x^2+3x+8$ , а остаток  $r=26$ .

### **Практические задания:**

Вычислить полиномы:  $n=10$

1.  $P_n(x) = a_0 + a_1*x + a_2*x^2 + \dots + a_n*x^n$
2.  $P_n(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + \dots,$

3.  $x + \frac{x^2}{2^2} + \frac{x^3}{3^2} + \frac{x^4}{4^2} + \frac{x^5}{5^2} + \dots = f(x)$
4.  $f(x) = x^3 + 4x^2 - 3x + 2$
5.  $P(x) = 1 + \frac{3}{2}x + \left(\frac{3}{2}x\right)^2 + \left(\frac{3}{2}x\right)^3 + \dots + \left(\frac{3}{2}x\right)^n$
6.  $f(x) = 1 - 3x^3 + (3x^3)^2 - (3x^3)^3 + (3x^3)^4 - (3x^3)^5 + \dots + (-1)^n (3x^3)^n$
7. Задать полином  $p(x) = x^3 - 3.55x^2 + 5.1x - 3.1$  и найти графически его единственный вещественный корень. Сколько знаков вам удастся получить, уменьшая интервал около корня?

**Варианты заданий**

№ варианта	Значение <i>a(или c)</i>	Значение <i>x</i>
1	4	6
2	5	7
3	6	2
4	7	5
5	7	2
6	8	2
7	4	3
8	5	3
9	3	6
10	4	5
11	9	7
12	5	9

## СПИСОК ЛИТЕРАТУРЫ

1. Курбатова Е.А. MATLAB 7. Самоучитель. – М., 2012.
2. Алексеев Е.Р., Чеснокова О.В. MATLAB 7. Самоучитель. – М., 2012.
3. Поршнева С.В. MATLAB 7. Основы работы и программирования. Учебник. – М., 2012.
4. Гандер В., Гржебичек И. Решение задач в научных вычислениях с применением Maple и MATLAB. – М., 2014.
5. Демидова Л.А., Кираковский В.В., Пылькин А.Н. Алгоритмы и системы нечеткого вывода при решении задач диагностики городских инженерных коммуникаций в среде MATLAB. – М., 2014.
6. Иглин С.П. Математические расчеты на базе Matlab. – СПб., 2012.
7. Мартынов Н.Н. Matlab 7. Элементарное введение. – М., 2012.
8. Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. Серия: Самоучитель. – М., 2012.
9. Подкур М.Л., Подкур П.Н., Смоленцев Н.К. Программирование в среде Borland C++ Builder с математическими библиотеками MATLAB. - М., 2013.
10. Дьяконов В.П. MATLAB 6.0/6.1/6.5/6.5 + SP1 + Simulink 4/5. Обработка сигналов и изображений. - М., 2014.



*Подписано в печать 18.02.2015 г. Тираж 500 экз.  
Формат изд. 60x84/16. Объем 3 усл. печ. л.  
Отпечатано в типографии “ИП Волков А.И.”  
Райымбека 212/1, оф. 319. Тел.: 330-03-12, 330-03-13*