



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Программное обеспечение вычислительной тех-
ники и автоматизированных систем»

Учебно-методическое пособие по дисциплине

«Теория вычислительных процессов и структур»

Автор
Загребнева А.Д.

Ростов-на-Дону, 2018



Аннотация

Учебно-методическое пособие предназначено для студентов очной формы обучения направления 02.03.03 «Математическое обеспечение и администрирование информационных систем».

Авторы

К.т.н.

Доцент каф. ПОВТиАС

Загребнева А.Д.



Оглавление

1. Лабораторная работа №1: Основы работы в системе MATLAB	4
1.1. Теория.....	4
1.2. Контрольные вопросы.....	12
1.3. Используемая литература.....	13
2. Лабораторная работа №2: Матричные операции в системе MATLAB	13
2.1. Теория.....	13
2.2. Контрольные вопросы.....	20
2.3. Используемая литература.....	21
3. Лабораторная работа №3: Двумерная графика в системе MATLAB	21
3.1. Цель работы.....	21
3.2. Контрольные вопросы.....	26
3.3. Используемая литература.....	27

1. ЛАБОРАТОРНАЯ РАБОТА №1: ОСНОВЫ РАБОТЫ В СИСТЕМЕ MATLAB

1.1. Теория

MATLAB (сокращение от англ. «MatrixLaboratory») - пакет прикладных программ для решения задач технических вычислений, а также используемый в этом пакете язык программирования. MATLAB работает на большинстве современных операционных систем, включая Linux, Mac OS, Solaris и Windows. Язык MATLAB представляет собой высокоуровневый язык программирования, включающий основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования. Основной особенностью языка MATLAB является его широкие возможности по работе с матрицами, которые создатели языка выразили в лозунге «Думай векторно» (англ. Thinkvectorized).

При вызове MATLAB на дисплей выводится заставка, которая сменяется Командным окном (CommandWindow) и окнами Текущий каталог (CurrentFolder) и Рабочая область (Workspace). В верхней части экрана размещена лента, на вкладках которой сгруппированы связанные команды. В окне Workspace отображаются все переменные и массивы, которые были использованы за время работы MATLAB, и их характеристики. Все символы команд, которые пользователи набирают с клавиатуры, результаты выполнения этих команд и информация об ошибках отображаются в CommandWindow.

Работа в командном окне среды MATLAB – CommandWindow

Вводкоманд

❶ В CommandWindow выводится командная строка, начинающаяся с символа `>>`. В командной строке в режиме диалога можно набрать команду (оператор) или выражение и, нажав клавишу Enter, получить ответ (answer).

1. Попробуйте умножить числа 3 и 5 с помощью команды `3 * 5`.

❶ Если не указана переменная, MATLAB хранит вычисления в переменной с именем `ans`.

```
>> 7 + 3
```

```
ans =
```

```
10
```

Теория вычислительных процессов и структур

2. Попробуйте присвоить переменной m значение $3 * 5$: $m = 3 * 5$
 - ❗ Знак равенства (=) в MATLAB является оператором присваивания. Когда вы вводите $x = 3 + 4$, MATLAB сначала вычисляет $3 + 4$, а затем присваивает результат (7) переменной x . Обратите внимание, что в окне рабочей области (Workspace) отображаются все переменные, находящиеся в настоящее время в рабочей области.
3. Присвойте переменной m значение $3 * 5$: $m = 3 * 5$.
Введите команду $m = m + 1$.
Создайте переменную с именем y , которая имеет значение $m / 2$.
 - ❗ Если имеется необходимость выполнить команду без вывода результата, то в конце команды ставится символ «;».
4. Введите $k = 8-2$; с точкой с запятой в конце.
 - ❗ Вы можете отображать пошагово все введенные ранее команды в Командном окне при помощи клавиш \uparrow и \downarrow . Обратите внимание, что CommandWindow должно быть активным для этого.
5. Нажмите \uparrow , чтобы вернуться к команде $m = 3 * 5$ и отредактировать команду $m = 3 * k$
 - ❗ В окне CommandHistory отображаются все введенные в окне CommandWindow за время работы команды, пролистывая это окно можно повторить ввод той или иной команды.
 - ❗ Пошагово эти команды можно отображать в Командном окне при помощи клавиш \uparrow и \downarrow .
6. Откройте окно CommandHistory и просмотрите все выполненные в Командном окне команды, выбрав вкладка HOME→Layout→CommandHistory→docked. Сделайте скриншот для отчета. Закройте окно CommandHistory, выбрав вкладка HOME→Layout→CommandHistory→close.

Использование переменных, встроенных функций и констант

- ❗ При вводе переменных необходимо учитывать, что их имена представляют собой последовательности латинских букв и цифр, начинающихся с буквы (знак $_$ относится к буквам), а также и то, что строчные и заглавные буквы не тождественны.
 - ❗ Для очистки CommandWindow необходимо выполнить команду *clc*. Для очистки Workspace полностью используется команда *clear*, для частичной очистки - команда *clear<список имен>*.
1. Очистите все переменные, введя команду *clear*.

2. Теперь очистите командное окно с помощью команды `clc`.
 - MATLAB содержит встроенные системные константы, которые создаются в системе при загрузке. Основные из них:
 - i или j – мнимая единица (корень квадратный из -1);
 - π – число $\pi \approx 3.1415926\dots$;
 - ϵ – погрешность операций над числами с плавающей точкой (2^{-52});
 - realmin – наименьшее число с плавающей точкой;
 - realmax – наибольшее число с плавающей точкой;
 - inf – значение машинной бесконечности;
 - NaN – указание на нечисловой характер данных (Not-a-Number).
3. Создайте переменную с именем x со значением $8\pi / N$, где N – номер варианта.
 - MATLAB содержит множество встроенных математических функций, таких как `abs` (абсолютное значение) и `eig` (вычисление собственных значений) и т.д. Из этого многообразия следует отметить некоторые наиболее используемые. Аргументами большинства из приведенных ниже функций являются не только скаляры, но и массивы:
 - a^x – степенная функция;
 - x^a – показательная функция
 - $\text{abs}(x)$ – абсолютная величина;
 - $\text{real}(x)$, $\text{imag}(x)$ – действительная и мнимая часть числа;
 - $\text{conj}(x)$ – комплексно-сопряженное значение;
 - $\text{sign}(x)$ – знак числа (для комплексных $x/|x|$);
 - $\text{sqrt}(x)$ – квадратный корень;
 - $\text{exp}(x)$ – экспонента;
 - $\text{log}(x)$, $\text{log}_2(x)$, $\text{log}_{10}(x)$ – натуральный логарифм, логарифмы по основанию 2 и 10;
 - $\text{sin}(x)$, $\text{cos}(x)$, $\text{tan}(x)$, $\text{cot}(x)$, $\text{csc}(x)$, $\text{sec}(x)$ – тригонометрические функции (синус, косинус, тангенс, котангенс, косеканс, секанс).
 Список элементарных математических функций может быть получен по команде `helpfun`.
4. Присвойте переменной y значение синуса x .
5. Присвойте переменной z квадратный корень из -9 .
6. Даны $x = 1,5$; $y = 2$; $z = 3$. Вычислить a , b из таблицы ниже для вашего варианта.

Номер варианта	a	b
1	2	3
1	$a = \frac{z + y/(x^2 + 4)}{e^{-x-2}/(x^2 + 4)}$	$b = \frac{x}{y}(\operatorname{arctg} z + 1/6)$
2	$a = \frac{3,5 + e^{y-1}}{1 + x^2 y - \operatorname{tg} z }$	$b = \frac{(y - x)^2}{2} + \frac{ y - x ^3}{3}$
3	$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ z }}{1 + \frac{x^2}{2,5} + \frac{y^2}{4}}$	$b = \frac{1 + \cos(y - 2)}{\frac{x^2}{2} + \sin^2 z}$
4	$a = z + \frac{x}{y^2 + \left \frac{x^2}{y + x^3/1,3} \right }$	$b = 1 + \operatorname{tg}^3\left(\frac{z}{2x + 2y}\right)$
5	$a = \frac{2,3 \cos(x - 1/6)}{1/2 + \sin^2 y} + z$	$b = x^y + \frac{z^2}{3 + z^2/5}$
6	$a = \frac{1,5 + \sin^2 z}{\left x - 2x/(1 + x^2 y^2) \right }$	$b = \cos^2\left(\operatorname{arctg} \frac{y}{z}\right) + \sin^2 x$

Теория вычислительных процессов и структур

7	$a = \ln \left (y) \frac{1,5y}{z + x^2/4} \right $	$b = x - \frac{x^2}{3!} + \frac{x^5}{5!}$
8	$a = \frac{\sin(x^2 - 2y + z)}{2,6x^y}$	$b = \cos^2 \left(x^2 + \frac{y}{z} \right)$
9	$a = \frac{5\cos(x - 1/6)}{0,5 + \sin^2 x}$	$b = \frac{y^2}{3 + z^2/7} - 3x$
10	$a = \frac{3,5 + \operatorname{tg}(x^2 + y)}{\left x - 4x/(1 + xy^2) \right }$	$b = \sin^2 \left(\operatorname{arctg} \frac{1}{z} \right)$
11	$a = \frac{2,6 + \operatorname{tg}(x - y)}{\left x - 2x/(x^2 + y^2) \right }$	$b = x(\operatorname{tg} z + e^{x-3})$
12	$a = \ln \left \frac{y + x^2/4}{5z} \right $	$b = 1,5 + \frac{(y - x)^3}{2} + \frac{ y - x }{x}$
13	$a = \frac{\cos(x^3 + 2y - 2z)}{\operatorname{tg} y - 1,5}$	$b = \frac{1 + \sin(y - 2)}{x^2 \frac{1}{2 + \sin^2 x}}$
14	$a = \frac{5\sin(x + 1/3)}{1/2 \cos x + 1}$	$b = \left(1 + \operatorname{tg}^2 \frac{z}{y + 2} \right)$
15	$a = \frac{3 + \sin^3(x^2 + y)}{2,5 + \left x - 4x/(1 + x^2 y^2) \right }$	$b = 1 + \frac{z^2}{3 + z^2/5}$
16	$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ y }}{1,5 + x^2 + y^2}$	$b = \cos^2 \left(\operatorname{arctg} \frac{1}{z+1} \right)$
17	$a = \frac{1,5 - e^{2-y}}{2x - \sqrt{y - \operatorname{tg} z}}$	$b = x + \frac{x^2}{3+z} + \frac{x^2}{5+z}$
18	$a = \frac{3,3 + y^2 + (x^2 + 2)}{e^{-0,5} + 1/(x^2 + 4)}$	$b = \cos^2 \left(x^2 + \frac{y}{1+z} \right)$
19	$a = y + \frac{3,5x}{y^2 - \sqrt{\frac{x^2}{2y + x^2}}}$	$b = 2 + \frac{y^2}{3 + \frac{z^2}{1+x}} - 3x$
20	$a = \frac{2\cos(x - 1/6)}{1/2 + \sin^2 y}$	$b = \sqrt{\sin \left(\operatorname{arctg} \frac{1}{x+z} \right)}$

Создание вектора и матрицы

Непосредственное задание элементов матрицы

- 1. Все переменные MATLAB являются массивами, что означает, что каждая переменная может содержать несколько элементов. Единственное число, называемое скаляром, на самом деле является массивом 1 на 1, то есть содержит 1 строку и 1 столбец.

- 2. Вы можете создавать массивы с несколькими элементами, используя квадратные скобки.

```
>>x = [3 5]
```

```
x =
```

```
3 5
```

1. Создайте массив x с двумя элементами в одной строке: 7 и 9.
2. Когда вы разделяете числа пробелами (или запятыми), MATLAB объединяет числа в вектор строки, который представляет собой массив с одной строкой и несколькими столбцами (1-на- n). Когда вы разделяете их точкой с запятой, MATLAB создает вектор столбца (n -на-1).

```
>>x = [2; 3]
```

```
x =
```

```
2
```

```
3
```

2. Создайте массив x с двумя элементами 7 и 9 в одном столбце. Попробуйте вспомнить предыдущую команду и измените пробел между числами на точку с запятой (;).
3. Сформируйте вектор-строку $v1$ размером 1 на 3 и вектор-столбец $v2$ размером 3 на 1, значения элементов которых в зависимости от номера варианта N равны: $v1 = [N \ 2N \ 3N]$, $v2 = [4N \ 5N \ 6N]$.

- 4. Информация: вы можете комбинировать пробелы и точки с запятой для создания матриц, которые представляют собой массивы с несколькими строками и столбцами. При вводе матриц вы должны вводить их подряд за строкой.

```
>> x = [3 4 5; 6 7 8]
```

```
x =
```

```
3 4 5
```

```
6 7 8
```

4. Создайте матрицу x со следующими значениями:

```
5 6 7
```

```
8 9 10
```

5. Создайте матрицу y со значениями, зависящими от номера варианта N :

N 3N

2N2N

3N N

- 7 В MATLAB вы можете выполнять вычисления внутри квадратных скобок.

```
>> x = [abs(-4)4 ^ 2]
```

```
x =
```

```
4 16
```

6. Создайте вектор строку x 1 на 2, которая содержит $\sqrt{10}$ в качестве своего первого элемента, а π^2 - второго элемента.

Линейный массив равноотстоящих узлов

- 7 Часто в практике используются векторы, элементы которого равномерно распределены по пространству, например, как вектор ниже:

```
>> y = [5 6 7 8]
```

```
y =
```

```
5 6 7 8
```

Для таких векторов существует удобный метод создания – использовать оператор «:» и указать только начальную и конечную точки:

```
>> y = 5: 8
```

```
y =
```

```
5 6 7 8
```

Оператор «:» использует по умолчанию интервал 1, но вы можете указать свой собственный интервал, как показано ниже.

```
>> x = 20: 2: 26
```

```
x =
```

```
20 22 24 26
```

1. Создайте вектор строку x, начинающуюся с 1, заканчивающуюся на (N+5), а интервал равен 0.5. Здесь N – номер варианта.

- 7 Если вы знаете количество элементов в векторе, вы можете использовать функцию *linspace(a,b,n)*, генерирующую n точек, равномерно распределенных в интервале (a, b).

```
>> x = linspace (0,1,5)
```

```
x =
```

```
0 0,250 0,500 0,750 1,000
```

2. Создайте вектор строку x, начинающуюся с 0, заканчивающуюся N, содержащую l элементов. Здесь N – номер варианта.

- 7 Функция *linspace* и оператор «:» создают векторы-строки. Вы можете преобразовать вектор-строку в вектор-столбец, используя оператор транспонирования (').

```
>> x = 1: 3;
>> x = x'
x =
    1
    2
    3
```

Или с помощью команды «>> x = (1: 2: 5) '».

- Создайте вектор столбец x , начинающийся с -5 , заканчивающийся N , с интервалом равным 2 . Здесь N – номер варианта.

Использование стандартных функции для создания массива

- MATLAB содержит множество функций для создания матриц специального вида, аргументами которых являются размерности создаваемых матриц. Приведем некоторые из них:

zeros(n), *zeros(m,n)*, *zeros(size(A))* - формирование массива нулей (одномерного, двумерного, соразмерного с массивом A); допустимо формирование массива и большей размерности *zeros(m, n, p, ...)*;

ones(n), *ones(m,n)*, *ones(size(A))* - формирование массива единиц;

rand(n), *rand(m,n)*, *rand(size(A))* - формирование массива чисел с равномерным законом распределения в $(0,1)$;

randn(n), *randn(m,n)*, *randn(size(A))* - формирование массива чисел с нормальным законом распределения ($Mx=0$, $Dx=1$);

eye(n), *eye(m,n)*, *eye(size(A))* - формирование единичной матрицы ($n \times n$, $m \times n$, соразмерной с матрицей A);

magic(n) - матрица магического квадрата.

Например, команда *rand(2)* создаст матрицу случайных чисел размерности 2×2

```
>> x = rand (2)
x =
    0,8147 0,1270
    0,9058 0,9134
```

- Создайте вектор-строку x , содержащую $(N+4)$ случайных числа, равномерно распределенных на $(0,1)$.
- Создайте единичную матрицу размером 5 на 5 .

Импорт данных

Сохранение и загрузка переменных

- Вы можете сохранять переменные своей рабочей области в формате MATLAB, который называется MAT-файлом, используя команду *save*.

```
>>save('mydata','x')
```

Приведенная выше команда сохраняет переменную *x* в MAT-файл с именем *mydata.mat*.

❶ Вы можете загружать переменные из MAT-файла с помощью команды *load*:

```
>>load('mydata.mat')
```

1. Сохраните переменные рабочего пространства в файл с именем *datafile.mat*.
2. Используйте *clear*, чтобы очистить рабочее пространство.
3. Загрузите переменные из файла *datafile.mat*.

Импорт данных в MATLAB с помощью «Import Tool»

1. Создайте текстовый файл *elements.txt* следующего содержания:

```
1 1.5 0.04
3 1e-2 1.5e10
```

и положите его в рабочую директорию MATLAB (указана и задается в окне *CurrentFolder*). Выберите файл и нажав правую кнопку мыши, выберите из списка «*Importdata...*». Задайте переменной название «*A*». Импортируйте данные в MATLAB в качестве матрицы, выбрав опцию «*NumericMatrix*» и нажав «*ImportSelection*».

Получение справочной информации

- ❶ В командном окне всегда можно обратиться к помощи, набрав команду *help*. Выбрав *help <раздел (topic)>*, можно получить помощь по командам соответствующего раздела. Для помощи в синтаксисе необходимо набрать *help<команда (функция)>*.
 - ❷ Вводя в командной строке *doc<команда (функция)>* можно получить детальную информации о любой функции MATLAB.
 - ❸ Полную документацию о программе можно получить, нажав на ленте кнопку «*Help*». В документации MATLAB содержится много хороших примеров и информации, которые могут помочь вам при работе над вашими собственными задачами.
1. Используйте документацию для функции *randi*, чтобы выполнить следующую задачу. Создайте матрицу *x*, состоящую из 5-ти строк и 7-ми столбцов, которая содержит случайные целые числа в диапазоне от 1 до $(N+10)$, где *N* – номер варианта.

1.2. Контрольные вопросы

1. Перечислите основные окна в MATLAB и объясните их назначение.
2. Какой символ используется для подавления вывода результатов выполнения операторов?
3. Переменная *ans*. Какие значения принимает перемен-

- ная ans при делении 1/0 и 0/0?
4. Как вызвать предыдущую команду в MATLAB?
 5. Какие системные константы MATLAB вы знаете?
 6. Какие встроенные функции MATLAB вы знаете?
 7. По каким правилам формируются имена переменных в MATLAB?
 8. Укажите способы формирования матрицы на языке MATLAB.
 9. Назовите встроенные функции для создания матриц в MATLAB.
 10. Каким образом сохраняются значения переменных в файле расширением «.mat» и как можно восстановить значения переменных, используемых в предыдущих сеансах?
 11. Как можно получить справочную информацию в MATLAB?

1.3. Используемая литература

1. Малеев, Р. А. Компьютерные технологии: учеб. пособие / Р. А. Малеев, В. В. Регада, О. Н. Регада. – Пенза : Изд-во ПГУ, 2014.
2. Методические разработки с использованием математических пакетов. - Режим доступа: <http://www.exponenta.ru>, <http://www.matlab.ru>.
3. Лазарев, Ю. Моделирование процессов и систем в MATLAB : учеб. курс / Ю. Лазарев. – СПб. : Питер ; Киев : Изд. группа ВУВ, 2005.

2. ЛАБОРАТОРНАЯ РАБОТА №2: МАТРИЧНЫЕ ОПЕРАЦИИ В СИСТЕМЕ MATLAB

2.1. Теория

2.1.1. Индексирование и изменение матриц

Используйте индексирование для извлечения и изменения строк, столбцов и элементов массивов MATLAB.

Индексирование в матрицах

1. Для обращения к любому элементу заданной матрицы A необходимо указать в скобках после имени матрицы номера соответствующих ему строки и столбца через запятую.

`>> x = A(5,7);` % элемент матрицы A в 5-й строке и 7-ом столбце

1. Создайте матрицу A , состоящую из 5-ти строк и 5-ти столбцов, которая содержит случайные целые числа в диапазоне от 1 до 100 с помощью функции *randi*.

Присвойте в зависимости от номера варианта (Таблица 1) элементу матрицы, находящемуся на пересечении n -й строки и m -

го столбца, значение, равное номеру варианта L .

Таблица 1. Соответствие номера варианта номерам строки и столбца.

№ строки (n)	№ варианта				
	1	1	2	3	4
2	6	7	8	9	1
3				0	
	1	1	1	1	1
4	1	2	3	4	5
	1	1	1	1	1
5	6	7	8	9	0
	2	2	2	2	2
	1	2	3	4	5
	1	2	3	4	5
	№ столбца (m)				

- Вы можете использовать ключевое слово «end» при обращении к последнему индексу строки или столбца матрицы
`>> x = A(end,7);` % элемент последней строки и 7-ом столбце матрицы A
- Используя ключевое слово *end*, присвойте переменной x элемент матрицы в последней строке и m -ом столбце данных переменных.
- Обратите внимание, что вы можете использовать арифметику с ключевым словом *end*. `>> x = A(end-1,end);` % элемент в предпоследней строке и последнем столбце матрицы A
- Присвойте переменной y элемент матрицы в n -ой строке и предпоследнем столбце.

Извлечение нескольких элементов

- С помощью двоеточия легко выделить часть матрицы. Например,

```
>> x = A(2, :) % вектор из 2-ой строки матрицы A
```

```
>> x = A(1:3,:) % матрица из 1-ой, 2-ой и 3-ей строки матрицы A
```

```
>> x = A(1:2,3) % вектор из первых двух элементов 3-его столбца матрицы A
```

- Создайте переменную x , которая содержит m столбец матрицы A, созданной ранее. Создайте переменную y , которая содержит n строку матрицы A.

2. Создайте матрицу B , содержащую четные строки матрицы для четных вариантов и нечетные строки матрицы для нечетных вариантов.
3. Создайте матрицу C , состоящую из последних 3-ех элементов последней строки матрицы A .
4. При работе с векторными элементами можно в круглых скобках указывать только один индекс.
 $\gg x = y(3)$ % 3-ий элемент вектора y
 $\gg x = y(3:end)$ % 3-ий – последний элементы вектора y
4. Создайте вектор z , содержащий 2-ой – 4-ый элементы вектора x .

Изменение значений матрицы

- ❶ Для удаления элемента вектора достаточно присвоить ему пустой массив — пару квадратных скобок $[\]$. Чтобы вычеркнуть одну или несколько строк (столбцов) матрицы нужно указать диапазон удаляемых строк (столбцов) для одной размерности и поставить двоеточие для другой размерности. Например, для удаления последнего столбца матрицы A достаточно ввести команду:

```
 $\gg A(:, end) = [\ ]$  % удаление последнего столбца в матрице
```

A

1. Найти матрицу $A1$, удалив из матрицы A n -ю строку. Найти матрицу $A2$ удалением из матрицы A m -го столбца.
- ❶ Матрицу можно определить из стандартных блоков. Например,

```
 $\gg A = [[3; 4] [-1; 2] [7; 0]]$ 
```

A =

```
3 -1 7
4 2 0
```

2. Образовать матрицу $A3$, добавив к матрице A 6-й столбец, равный вектору x .
- ❶ Двоеточие применяется также для замещения элементов матрицы. Следующая команда позволяет заменить первую строку матрицы A второй:

```
 $\gg A(1, :) = A(2, :)$  % 1-ая строка матрицы  $A$  заменена 2-ой строкой
```

3. Образовать матрицу $A4$, в которой переставлены первая и n -ая строки матрицы A (если $n=1$, то переставить n -ую и последнюю строки матрицы).

2.1.2. Операции с матрицами

Набор арифметических операций в MATLAB состоит из стандартных операций сложения-вычитания, умножения-деления,

операции возведения в степень и дополнен специальными матричными операциями.

Алгебраические операции с матрицами

- Запись операций сложения (вычитания) и умножения матриц естественна. Если операция применяется к матрицам, размеры которых не согласованы, то будет выведено сообщение об ошибке. Для поэлементного выполнения операций умножения, деления и возведения в степень применяются комбинированные знаки (точка и знак операции). Например, если за матрицей стоит знак (^), то она возводится в степень, а комбинация (.^) означает возведение в степень каждого элемента матрицы. При умножении (сложении, вычитании, делении) матрицы на число соответствующая операция всегда производится поэлементно.

>> B = A + 2 % добавление скалярного значения ко всем элементам матрицы A

>> C = A + B % сложение матриц

>> z = [3 4] * [10; 20] % умножение векторов

>> z = [3 4] .* [10 20] % поэлементное умножение векторов

- Добавьте число N , где N – номер варианта, к каждому элементу матрицы A и сохраните результат в переменную B .
- Сложите матрицы A и B , перемножьте матрицы A и B , перемножьте поэлементно матрицы A и B , вычислите матрицу N^*A , где N – номер варианта.
- Найти матрицу C , транспонированную к матрице A .

2.1.3. Использование функций при работе с матрицами

- Полный перечень стандартных функций работы с матрицами можно посмотреть, набрав в командной строке `help elmat`. Отметим некоторые полезные функции:

`size(A)`, `size(A, dim)` – размерность матрицы;

`length(v)` – размерность массива (только для векторов);

`cross(x,y)` – векторное произведение (x , y – трехмерные векторы);

`kron(A,B)` – тензорное произведение (произведение Кронекера);

`sum(A)`, `sum(A, dim)` – сумма элементов матрицы;

`prod(A)`, `prod(A, dim)` – произведение элементов матрицы;

`min(A)`, `min(A, dim)`, `max(A, dim)` – минимальный и максимальные элементы матрицы;

$sort(A)$, $sort(A, dim)$, $sort(A, direction)$ – сортировка элементов матрицы A ;

$sortrows(A)$, $sortrows(A, direction)$ – сортировка строк матрицы;

$diag(v)$, $diag(A)$, $diag(A, k)$ – создание диагональной матрицы или выделение диагонали;

$triu(A)$, $triu(A, k)$ – выделение верхней треугольной части матрицы;

$trll(A)$, $trll(A, k)$ – выделение нижней треугольной части матрицы;

$rank(A)$ – вычисление ранга матрицы;

$norm(A, p)$ – вычисление нормы, для вектора норма вычисляется $\sum(\text{abs}(A).^p)^{1/p}$;

$inv(A)$, A^{-1} – обратная матрица.

1. Для двух векторов $a = \{0, N, N + 1\}$ и $b = \{N + 1, N, 0\}$, где N

– номер варианта, вычислить $a \cdot b$, $a \times b$, $|a|$.

2. Создайте с помощью функции $randi$ матрицу A , состоящую из 5-ти строк и 5-ти столбцов и содержащую случайные целые числа, равномерно распределенные в диапазоне от -50 до 50.

3. Определите максимальный и минимальный элементы матрицы A и его индексы. Проверьте правильность нахождения индексов.

4. Вычислите определитель матрицы A .

5. Найти матрицу $A1$, обратную матрице A . Если эта операция невыполнима, переопределите какой-либо элемент матрицы A , чтобы сделать ее допустимой.

6. Выполните сортировку m -го столбца матрицы A .

7. Найдите сумму всех элементов матрицы A .

8. Найдите вектор d , содержащий числа, которые находятся на главной диагонали матрицы A , сформируйте диагональную матрицу D , содержащую нулевые элементы и диагональ A .

2.1.4. Решение систем линейных уравнений

Обращение матрицы используется для решения системы линейных уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n, \end{cases}$$

при задании ее в матричной форме как

$$\mathbf{Ax} = \mathbf{b}, \quad (1)$$

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

где $\mathbf{A} = [a_{i,j}]$, $i, j = 1..n$,

Система (1) имеет единственное решение, равное $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, если ее определитель отличен от нуля $\det(\mathbf{A}) \neq 0$.

Приведем пример решения системы малой размерности:

```
>> A = [0 1; 2 1], B = [1 3]', X = A\B
```

```
A = 0 1
```

```
2 1
```

```
B = 1
```

```
3
```

```
X = 1
```

```
1
```

1. Решите систему уравнений

$$\begin{cases} x + y - z = 36, \\ x + z - y = 13, \\ y + z - x = N, \end{cases}$$

где N – номер варианта. Найти обратную матрицу C , соответствующую данной системе. Проверить, что матрица C является обратной.

2.1.5. Логические операции с матрицами

Логические операции и переменные

- Операторы отношения, такие как $>$, $<$, $==$ и \sim = выполняют сравнения между двумя значениями. Результат сравнения для равенства или неравенства равен 1 (true) или 0 (false). Вы можете сравнить вектор или матрицу с одним скалярным значением с использованием реляционных операторов. Результатом является логический массив того же размера, что и исходный массив.

```
>> [5 10 15]> 12
```

```
ans =
```

```
0 0 1
```

1. Создайте вектор $v1$ из 5 элементов, содержащий случайные целые числа, равномерно распределенные в диапазоне от 1 до 50.
2. Создайте вектор x , содержащий результат тестиро-

вания, будет ли каждый элемент $v1$ больше N , где N – номер варианта.

- 7 Соответствующие элементы двух массивов можно сравнить с использованием реляционных операторов. Эти два массива должны иметь одинаковый размер, а результат - логический массив того же размера.

```
>> [5 10 15]> [6 9 20]
ans =
    0 1 0
```

3. Создайте вектор $v2$ из 5 элементов, содержащий случайные целые числа, равномерно распределенные в диапазоне от 1 до 50.
4. Создайте вектор x , содержащий результат тестирования, будет ли каждый элемент $v1$ больше вектора $v2$.

Сочетание логических условий

- 7 MATLAB содержит логические операторы, которые объединяют несколько логических условий, таких как AND (&) и OR (|).

```
>>x = (pi> 5) & (0 <6)
x =
    0
```

1. Создайте массив $v3$, элементы которого принимают истинные значения, когда элементы $v1$ и $v2$ больше $(N+10)$, где N – номер варианта.

Логическая индексация

- 7 Вы можете использовать логические операции при индексировании массива. В этом случае MATLAB выбирает все элементы массива, для которых условие является истинным.

```
>> v = v1 (v1> 6) % все элементы превышающие шесть
v =
    6,6678
    9,0698
```

1. Создайте массив $v4$, содержащий все элементы $v1$ большие $(N+10)$, где N – номер варианта.
2. Создайте массив $v5$, содержащий все элементы вектора $v2$, соответствующие ситуации, когда элементы $v1$ больше $(N+10)$, где N – номер варианта.

- 7 Вы можете использовать индексацию для переопределения значений в массиве.

```
>> x(x == 999) = 0; % замена всех значений в массиве x
равных 999 значением 0
```

3. Измените $v1$ так, чтобы любое значение, меньшее $(N+10)$, где N – номер варианта, было заменено значением 0.

2.1.6. Команды проверки и сравнения

- i Элементарные логические операции дополнены набором функций MATLAB, позволяющих проверить для матриц некоторые условия, см. ниже. Проверка для матриц проводится по столбцам. Результатом проверки для вектора является число.

find – поиск значений согласно заданному условию; определение индексов;

all – проверка того, что все элементы не равны нулю;

any – проверка того, что хотя бы один элемент не равен нулю;

isempty – выявление пустого массива;

isequal – проверка равенства матриц;

issparse – проверка матрицы на разреженность;

nonzeros – вывод ненулевых элементов массива;

isnumeric – проверка, является ли массив числовым;

isinf – выявление бесконечных элементов массива;

isnan – выявление элементов нечислового типа;

isletter – проверка на символ;

isstr – проверка на строковую переменную;

strcmp – сравнение двух строк.

4. Создайте массив *v6*, содержащий индексы элементов вектора *v1* равные 0 с помощью функции *find*.
5. Создайте массив *v7* содержащий ненулевые элементы вектора *v1*.

2.2. Контрольные вопросы

1. Какие способы обращения к элементам, строкам и столбцам заданной матрицы существуют в системе MATLAB?
2. Как осуществляется структурирование матрицы: выделение заданной подматрицы, удаление строк или столбцов?
3. Как осуществляется конкатенация матриц, т.е. объединение нескольких малых матриц в одну большую?
4. Каким критериям должны отвечать матрицы при сложении, вычитании и перемножении?
5. Перечислите и объясните действие операторов, используемых при арифметических операциях с матрицами.
6. Как в MATLAB решается система линейных уравнений, заданная в матричной форме?
7. Как используются логические операции при индексировании матрицы?
8. Назовите несколько команд проверки и сравнения матриц в MATLAB.

2.3. Используемая литература

1. Малеев, Р. А. Компьютерные технологии: учеб. пособие / Р. А. Малеев, В. В. Регода, О. Н. Регода. – Пенза: Изд-во ПГУ, 2014.
2. Методические разработки с использованием математических пакетов. - Режим доступа: <http://www.exponenta.ru>, <http://www.matlab.ru>.
3. Лазарев, Ю. Моделирование процессов и систем в MATLAB : учеб. курс / Ю. Лазарев. – СПб.: Питер ; Киев : Изд. группа ВУВ, 2005.
4. Говорухин, В.Н., Цибулин В.Г. Компьютер в математическом исследовании: учеб. курс. – СПб.: Питер, 2001.

3. ЛАБОРАТОРНАЯ РАБОТА №3: ДВУМЕРНАЯ ГРАФИКА В СИСТЕМЕ MATLAB

3.1. Цель работы

Построение графика кривой на плоскости

Для построения в системе MATLAB графика функции одной переменной $y(x)$ в декартовой (прямоугольной) системе координат используется команда

plot (X1, Y1, S1, X2, Y2, S2, ...)

где X_i, Y_i – заданные векторы одинакового размера, элементы которых являются координатами точек i -й кривой; S_i – обязательные символьные переменные, относящиеся к i -й кривой. Любая из S_i может содержать до трех специальных символов, определяющих тип линии, соединяющей отдельные точки графика (таблица 1), тип точки графика (таблица 2) и цвет линии (таблица 3).

Таблица 2. Тип линии.

Обозначение	Тип линии	Обозначение	Тип линии
–	Сплошная	– .	Штрих-пунктир
:	Двойной пунктир	– –	Штриховая

Таблица 3. Тип точки графика.

Обозначение	Тип точки	Обозначение	Тип точки
-------------	-----------	-------------	-----------

.	Точка	V	Треугольник (вниз)
0	Окружность	A	Треугольник (вверх)
X	Крест	<	Треугольник (влево)
+	Плюс	>	Треугольник (вправо)
*	Звездочка	P	Пятиугольник
S	Квадрат	H	Шестиугольник
D	Ромб		

Таблица 4. Цвет линии.

Обозначение	Цвет линии	Обозначение	Цвет линии
Y	Желтый	G	Зеленый
M	Фиолетовый	B	Синий
C	Голубой	W	Белый
R	Красный	K	Черный

В случае, если значения переменных S не указано, то, по умолчанию, принимаются следующие параметры графика: тип линии графика – сплошная; тип точки графика – пиксель; цвет каждого последующего графика устанавливается в следующем порядке: синий, зеленый, красный, голубой, фиолетовый, желтый, черный.

Рассмотрим пример построения функции $y = 10\sin(x + \pi/4)$ для значений аргумента, изменяющихся от 0 до 2π с шагом, равным $\pi/100$. Соответствующий программный код приведен ниже:

```
X = [0:pi/100:2*pi];
Y = (sin(X+pi/4)).*(sin(X+pi/4));
plot(X,Y);
xlabel('x');
ylabel('y');
grid
title('Функция y = sin^2(x+\pi/4)')
```

Сначала сформируем вектор значений аргумента X , затем вычислим вектор соответствующих значений функции Y и, наконец, построим график зависимости $Y(X)$ с помощью функции **plot**.

Функция **grid** позволяет добавить на график сетку из координатных линий. Названия осей задаются с помощью функции **xlabel** и **ylabel**, а заголовок графика выводится с помощью процедуры **title**. Обратите внимание на то, что для вывода в заго-

ловке графика греческой буквы «п» перед символами «**pi**» добавлена обратная косая черта «\». Аналогично можно вывести и другие прописные и строчные греческие буквы `\alpha`, `\beta`, `\gamma`, `\Gamma`, `\Sigma`, `\Pi` и т.д. Для задания индексов используется символ «^» перед верхним индексом или символ «_» перед нижним индексом.

В результате на экране появится окно, в котором будет построен заданный график (Рисунок 1).

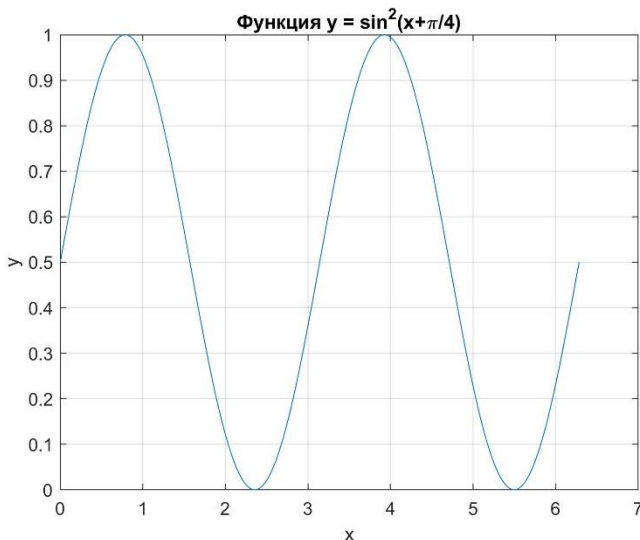


Рисунок 1. График функции $y = 10 \sin(x + \pi/4)$.

Для того, чтобы убрать сетку введите `grid off`.

При построении графика выбор масштаба и построение осей совершаются автоматически, а для изменения масштабов применяется команда **axis**. Чтобы определить интервалы изменения координат самостоятельно, нужно выполнить команду

axis([xmin, xmax, ymin, ymax])

Здесь числа `xmin` и `xmax` задают интервал изменения горизонтальной координаты (минимальное и максимальное значения), `ymin` `ymax` — соответственно интервал изменения вертикальной координаты. Для изменения масштаба в примере наберите:

`axis([0 2*pi 0 1]).`

Для добавления текста в определенное место графика, например, для обозначения кривой графика, используется команда

text(x, y, 'string')

где **x**, **y** определяют координаты точки начала текста;

'string' –строковая константа, определяющая вводимый в график текст. Приведем пример текста, вставленного на графике в точку с координатами $x=1.2$, $y = \sin(1.2+\pi/4)*\sin(1.2+\pi/4)$:

```
text(1.2,sin(1.2+pi/4)*sin(1.2+pi/4),'\leftarrow  
sin^2(x+\pi/4)',...
```

```
'FontSize', 12). % задания требуемого размера шрифта в  
пунктах
```

Обратите внимание на то, что строковая константа начинается с обратной косой черты «\» и команды **leftarrow**, которая задает стрелку, направленную влево. Аналогично можно добавить и другие виды стрелок (команды **\uparrow**, **\rightarrow** и т.д.), а также знаки «больше или равно», «меньше или равно», «не равно» (команды **\geq**, **\leq**, **\neq**). Здесь для задания требуемого размера шрифта в пунктах используется строковая константа 'FontSize'. Изменить написание текста можно с помощью строковой константы 'FontWeight', например:

```
title('Функция  $y = \sin^2(x+\pi/4)$ ','FontWeight', 'normal').
```

Координаты точки ввода текста в график удобно задавать с помощью мыши, используя команду **gtext**('string'). При исполнении этой команды на графике появляется большое перекрестие, установив которое в нужное место графика, достаточно нажать любую клавишу или любую кнопку мыши, и на этом месте появится надпись. Например, вывести текст можно с помощью команды:

```
gtext('Тестовый вариант').
```

График функции в новом масштабе представлен на Рисунке

2.

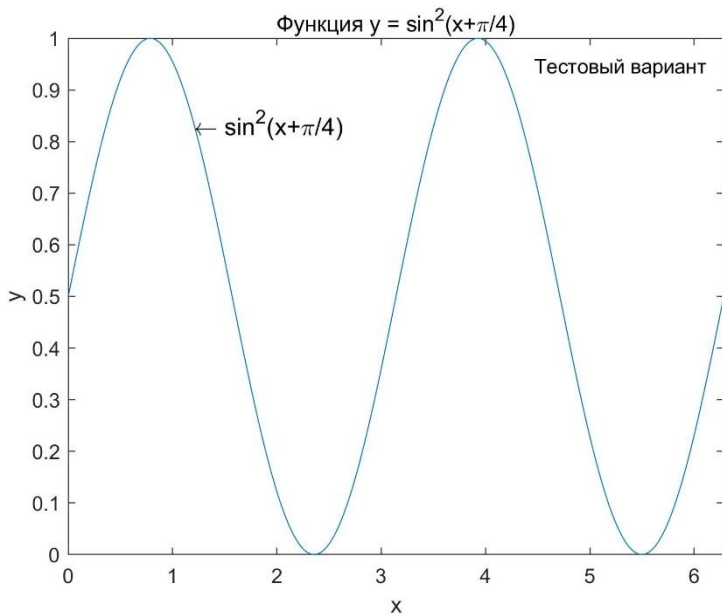


Рисунок 2. График функции $y = 10 \sin^2(x + \pi/4)$.

1. Построить график функции для заданного варианта задания (таблица 4). Добавить к графику сетку из координатных линий, названия осей и заголовков, обозначьте кривую, вставьте номер варианта. Включить в отчет по лабораторной работе тексты созданных самостоятельно m -файлов на языке MATLAB и полученный график функции в хорошем разрешении.

Таблица 4. Функция для изображения.

Номер варианта	Функция	Диапазон изменения x	Шаг изменения x
1	$Y = -\sin(x/2)$	$0^\circ \div 360^\circ$	10°
2	$Y = 13x^2$	$0 \div 24$	8
3	$Y = \cos(4x - 60^\circ)$	$0^\circ \div 180^\circ$	5°
4	$Y = \frac{1}{x + 4}$	$-1 \div 3$	0,5
5	$Y = \frac{x + 44}{x - 23}$	$-8 \div 8$	2
6	$Y = (3x^2 + 43)$	$0 \div 40$	4
7	$Y = \operatorname{tg}(x) - 1$	$0^\circ \div 90^\circ$	2°
8	$Y = \frac{x}{x - 31}$	$-10 \div 30$	5
9	$Y = (x^3 - x)$	$0 \div 200$	20
10	$Y = \sin(4x - 45^\circ)$	$0 - 180^\circ$	2°
11	$Y = \sin(3x + 60^\circ)$	$0 - 360^\circ$	4°
12	$Y = 1 - e^{5x}$	0-5	0,1
13	$Y = \sin(5x)$	$0 - 360^\circ$	4°
14	$Y = \operatorname{tg}(x + 30^\circ)$	$0^\circ \div 90^\circ$	2°
15	$Y = \cos(5x - 15^\circ)$	$0 - 180^\circ$	1°
16	$Y = \frac{2x}{x + 100}$	$-80 \div 80$	10
17	$Y = 1 - e^{2x}$	0-5	0,1
18	$Y = 1 - x^2$	0-1	0,01
19	$Y = 1 + e^{4x}$	0-5	0,1
20	$Y = \cos(4x)$	$0 - 360^\circ$	2°

3.2. Контрольные вопросы

1. Какие команды используются в MATLAB для построения одного и нескольких графиков в одних осях?
2. Как задаются различные стили графиков?
3. Как добавить к графикам сетку из координатных линий, названия осей, легенду и заголовок?
4. Каким образом в MATLAB можно вводить верхние и нижние индексы с помощью команды text?
5. Каким образом строятся графики в логарифмическом масштабе?
6. Как можно отредактировать созданный график в MATLAB?

7. Каким образом в MATLAB можно сохранить график в файле?
8. Как можно открыть график из файла в MATLAB?

3.3. Используемая литература

1. Малеев, Р. А. Компьютерные технологии: учеб. пособие / Р. А. Малеев, В. В. Регода, О. Н. Регода. – Пенза : Изд-во ПГУ, 2014.
2. Методические разработки с использованием математических пакетов. - Режим доступа:<http://www.exponenta.ru>, <http://www.matlab.ru>.
3. Лазарев, Ю. Моделирование процессов и систем в MATLAB : учеб. курс / Ю. Лазарев. – СПб. : Питер ; Киев : Изд. группа ВУВ, 2005.