

## 3. Типы массивов

### 3.1. Цель работы

Изучить основные типы массивов, используемых в MATLAB, и овладеть навыками их формирования.

### 3.2. Краткая теоретическая справка

В MATLAB тип массива (тип данных) определяется типом его элементов. По умолчанию мы имели дело с матрицами, элементы которых представлены константами в форме с плавающей точкой (ПТ) с двойной точностью. Такие массивы относятся к типу `double`.

Типы числовых и нечисловых массивов представлены в табл. 3.1.

Таблица 3.1. Типы массивов в MATLAB

| Символическое обозначение типа массива | Тип массива                                | Функция преобразования типа |
|--|--|-----------------------------|
| <code>double</code>                    | Числовой:<br>вещественный двойной точности | <code>double(X)</code>      |
| <code>single</code>                    | вещественный одинарной точности            | <code>single(X)</code>      |
| <code>int8</code>                      | целый 8-битовый со знаком                  | <code>int8(X)</code>        |
| <code>uint8</code>                     | целый 8-битовый без знака                  | <code>uint8(X)</code>       |
| <code>int16</code>                     | целый 16-битовый со знаком                 | <code>int16(X)</code>       |
| <code>uint16</code>                    | целый 16-битовый без знака                 | <code>uint16(X)</code>      |
| <code>int32</code>                     | целый 32-битовый со знаком                 | <code>int32(X)</code>       |
| <code>uint32</code>                    | целый 32-битовый без знака                 | <code>uint32(X)</code>      |
| <code>int64</code>                     | целый 64-битовый со знаком                 | <code>int64(X)</code>       |
| <code>uint64</code>                    | целый 64-битовый без знака                 | <code>uint64(X)</code>      |
| <code>logical</code>                   | Логический                                 | <code>logical(X)</code>     |
| <code>character (char)</code>          | Символьный                                 | <code>num2str(X)</code>     |
| <code>structure (struct)</code>        | Структура (массив записей)                 | —                           |
| <code>cell</code>                      | Массив ячеек                               | —                           |

### 3.2.1. Матрицы числового и логического типов

Преобразование матриц числового типа `double` в матрицы других числовых типов выполняется с помощью специальных встроенных функций (см. табл. 3.1).

При обработке матриц числового *целого* типа необходимо иметь в виду, что с ними *запрещено* выполнение арифметических операций, в том числе вычисление большинства встроенных функций, но *разрешено* выполнение операций отношения и логических операций:

```
>> A = [int8(159.7) int8(125.7) int8(-125.7)]
A =
    127    126   -126
>> B = [uint8(159.7) uint8(125.7) uint8(-125.7)]
B =
    160    126     0
>> C = [A<B; A==B; and(A,B)]
C =
     1     0     1
     0     1     0
     1     1     0
```

Элементами матрицы *логического* типа (`logical array`) являются логические константы, принимающие значения 1 (`true` — истина) или 0 (`false` — ложь), например, как в матрице C, или логические выражения (см. разд. 1.2.2):

```
>> x = [sin(3)<0.5  1; 0 (sin(3)<0.1)&(cos(3)<0.2)]
x =
     1     1
     0     0
```

### 3.2.2. Матрицы символьного типа

*Матрица символьного типа* (`char array`) — это разновидность нечисловых матриц, элементами которой являются символьные константы (см. разд. 1.2.2).

Строки и столбцы матрицы символьного типа формируются по-разному, а именно:

- элементы *строки* представляют собой *слитную* запись, поэтому необходимые пробелы должен предусмотреть пользователь:

```
>> a = ['Alla ' 'Woman ' 'Russian']
a =
Alla Woman Russian
```

- элементы *столбца* должны содержать *одинаковое* число символов, при этом пробел считается символом:

```
>> a = ['Alla  '; 'Woman  '; 'Russian']
```

```
a =
Alla
Woman
Russian
```

*Автоматическое* добавление пробелов в элементах столбца с выравниванием по левому краю выполняется с помощью функции:

```
char ('<char1>', '<char2>' ...)
```

где *<char1>*, *<char2>*... — элементы столбца с произвольным количеством символов.

Одна функция `char` описывает один столбец матрицы с символьными константами `'<char1>', '<char2>' ...`:

```
>> a = char('Alla', 'Woman', 'Russian')
a =
Alla
Woman
Russian
```

Матрицу символьного типа удобно формировать по столбцам, используя для каждого из них свою функцию `char` и предусматривая необходимое количество пробелов для разделения столбцов:

```
>> x = [char('a', 'aa', 'aaa') char(' bb', ' bbb', ' b') ...
char(' cc', ' ccc', ' c')]
x =
a   bb  cc
aa  bbb ccc
aaa b   c
```

Преобразование матрицы численного или логического типа в матрицу символьного типа выполняется с помощью функции `num2str(x)`:

```
>> x = [5 7; -1 9]
x =
     5     7
    -1     9
>> y = num2str(x)
y =
5 7
-1 9
```

Здесь `x` — матрица типа `double`, а `y` — матрица символьного типа:

```
>> [size(x); size(y)]
ans =
```

### 3.2.3. Структуры (массивы записей)

*Структура* (массив записей — struct array) — это разновидность нечислового массива, предназначенного для описания  $M$  объектов  $N$  параметрами.

Для описания структуры потребуется ввести ряд новых терминов:

- *поле* (field) — имя параметра, описывающего объект: скаляра, вектора, матрицы<sup>1</sup> или нечислового массива;

*Число полей* равно количеству параметров  $N$ .

- *значение поля* — значение параметра;
- *список полей* — список имен параметров;
- *запись* — список полей, одинаковый для всех  $M$  объектов;

*Число записей* равно количеству объектов  $M$ .

- *значение записи* — список полей с их значениями для одного объекта.

*Структура* (массив записей) — это упорядоченная совокупность значений записей, объединенная одним именем.

*Имя* массива записей выбирается так же, как обычно для переменной (см. разд. 1.2.2), а *размер* равен числу записей  $M$ .

Значение каждой  $i$ -й записи формируется отдельно по каждому  $n$ -му полю:

**<имя массива>( < $i$ > ).<имя  $n$ -го поля> = <значение  $n$ -го поля>**

Таким образом, для формирования  $M$  значений записей со списком из  $N$  полей, потребуется  $M \times N$  операторов присваивания.

Сформируем массив записей (структуру) с именем `personal` для описания трех объектов ( $M = 3$ ) — трех членов кафедры — четырьмя параметрами ( $N = 4$ ).

Запись включает в себя следующий список полей:

`surname` — фамилия — скаляр символьного типа;

`data` — дата рождения (число, месяц, год) — вектор численного типа;

`position` — должность — скаляр символьного типа;

`phd` — наличие ученой степени — скаляр логического типа.

Сформируем значения массива записей `personal` по каждому полю:

```
>> personal(1).surname = 'Иванов';  
>> personal(2).surname = 'Петров';  
>> personal(3).surname = 'Сидоров';
```

---

<sup>1</sup> Напомним (см. разд. 1.2.1), что по умолчанию в MATLAB любая переменная — матрица, а скаляр и вектор ее частные случаи.

```
>> personal(1).data = [1 2 1949];
>> personal(2).data = [5 7 1975];
>> personal(3).data = [5 8 1956];
>> personal(1).position = 'профессор';
>> personal(2).position = 'доцент';
>> personal(3).position = 'зав.лаб.';
>> personal(1).phd = true;
>> personal(2).phd = true;
>> personal(3).phd = false;
```

*Список полей* выводится по имени массива записей:

```
>> personal
personal =
1x3 struct array with fields:
    surname
    data
    position
    phd
```

*Значение i-й записи* выводится по имени массива записей с указанием индекса в круглых скобках. Например, значение 2-й записи:

```
>> personal(2)
ans =
    surname: 'Петров'
    data: [5 7 1975]
    position: 'доцент'
    phd: 1
```

*Значение поля в i-й записи* выводится по имени массива записей с указанием индекса в круглых скобках и имени поля. Например, значение поля surname первой записи:

```
>> personal(1).surname
ans =
Иванов
```

*Значения поля во всех записях* выводятся по имени массива с указанием имени поля. Например, поля surname:

```
>> personal.surname
ans =
Иванов
ans =
Петров
ans =
```

Сидорова

Удаление поля выполняется с помощью функции:

```
<имя массива> = rmfield(<имя массива>, '<имя поля>')
```

Например, удалим поле data:

```
>> personal = rmfield(personal, 'data')
personal =
1x3 struct array with fields:
    surname
    position
    phd
```

### 3.2.4. Массивы ячеек

*Массив ячеек* (cell array) — это наиболее сложный тип массива, элементами которого являются ячейки, представляющие собой массивы любой размерности, любого размера и типа.

Элементы массива ячеек указываются в *фигурных* скобках.

Сформируем квадратную матрицу ячеек 3×3:

```
>> A{1,1} = pi;
>> A{1,2} = [1 2 3;4 5 6];
>> A{1,3} = char('abs', 'angle');
>> A{2,1} = [ones(5,1)]';
>> A{2,2} = zeros(3,1);
>> A{2,3} = 'Alla';
>> A{3,1} = 7 ;
>> A{3,2} = rand(5,1);
>> A{3,3} = personal;
```

где `personal` — имя структуры, сформированной в *разд. 3.2.3*

*Вывод элементов* массива ячеек выполняется по его имени с указанием индексов:

```
>> A{1,2}
ans =
     1     2     3
     4     5     6

>> A{3,3}
ans =
1x3 struct array with fields:
    surname
    age
    position
```

pd

С элементами массива ячеек можно выполнять операции, разрешенные для данного типа массива и с учетом согласования их размерностей и размеров, например:

```
>> B = sum(A{1,2})+A{1,1}
B =
    8.1416    10.1416    12.1416
```

Графическое представление матрицы ячеек создается с помощью функции:

```
cellplot(A, 'legend')
```

### 3.2.5. Определение типа массива

Для определения типа массива служит функция:

```
class(<имя массива>)
```

Например, для массива A, сформированного в разд. 3.2.4:

```
>> class(A)
ans =
cell
```

## 3.3. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008, гл. 4.
2. Сергиенко А. Б. Цифровая обработка сигналов. 3-е издание — СПб.: БХВ-Петербург, 2010, Приложения 1—2.

## 3.4. Содержание лабораторной работы

Содержание работы связано с изучением типов массивов в MATLAB в режиме прямых вычислений.

## 3.5. Задание на лабораторную работу

Задание на лабораторную работу включает в себя следующие пункты:

1. Знакомство с матрицами числового и логического типов.

Ввести матрицу A с элементами:

$$A = \begin{pmatrix} 127,1 & -127,1 & 127,7 \\ -127,7 & 0 & 128,4 \\ -128,4 & 255,7 & 255,1 \end{pmatrix}$$

и выполнить с ней следующие действия:

- преобразовать в матрицу B целых 8-битовых чисел со знаком;

- преобразовать в матрицу  $C$  целых 8-битовых чисел без знака;
- преобразовать в матрицу  $D$  логического типа;
- определить тип матриц  $A, B, C, D$ .

Пояснить:

- как преобразовать матрицу  $A$  в матрицы  $B, C$  и  $D$ ;
- по какому правилу формируются элементы матрицы  $B$ ;
- по какому правилу формируются элементы матрицы  $C$ ;
- по какому правилу формируются элементы матрицы  $D$ ;
- как определяется тип матрицы.

## 2. Операции с матрицами числового типа.

Выполнить следующие операции:

- вычислить значения синуса всех элементов матриц  $A, B$  и  $C$ ;
- вычислить суммы матриц  $A$  и  $B, B$  и  $C$ ;
- проверить, являются ли элементы матрицы  $A$  числами, большими единицы, и определить вид и тип результата;
- выполнить логическую операцию "И" с матрицами  $B, C$  и определить вид и тип результата.

Сделать выводы по результатам выполнения операций.

## 3. Знакомство с матрицами символьного типа.

Выполнить следующие действия:

- сформировать трехэлементный вектор-строку  $X$  символьного типа с ФИО студента;
- сформировать трехэлементный вектор-столбец  $Y$  символьного типа с ФИО студента;
- сформировать матрицу  $F$  символьного типа  $2 \times 2$  с элементами:

$$F = \begin{pmatrix} \text{КИХ} & R=15 \\ \text{FIR} & \text{Order}=15 \end{pmatrix}$$

- преобразовать матрицу  $A$  (см. п. 1) в матрицу  $G$  символьного типа;
- определить типы матриц  $A$  и  $G$ .

Пояснить:

- как обеспечить не слитный вывод элементов вектора-строки;
- как обеспечить автоматическое добавление пробелов при выводе элементов столбца;
- как преобразовать матрицу числового типа в матрицу символьного типа.



4. Знакомство с массивами записей (структурами).

Сформировать массив записей (структуру) с именем `Filter` для описания четырех фильтров.

Каждая запись должна содержать три поля со следующими именами и их значениями:

`Type` (тип избирательности) — `lowpass`, `highpass`, `bandpass`, `stopband`;

`Order` (порядок фильтра) — `10`, `20`, `30`, `40`;

`Poles` (наличие полюсов) — `true`, `false`, `false`, `true`.

Выполнить следующие действия с массивом `Filter`:

- вывести список полей;
- вывести значение 1-й записи;
- вывести значения поля `Type` во всех записях;
- вывести значение поля `Type` в 3-й записи;
- удалить поле `Poles`.

Пояснить:

- с какой целью создается массив записей;
- что собой представляет запись и значение записи;
- каков размер массива записей `Filter`.

5. Знакомство с матрицами ячеек.

Создать матрицу ячеек  $S$   $3 \times 3$ , элементами которой являются сформированные ранее массивы:

$$S = \begin{pmatrix} A & B & C \\ D & F & G \\ X & Y & Filter \end{pmatrix}.$$

Выполнить следующие действия:

- последовательно вывести элементы матрицы  $S$  и определить их тип;
- вывести графическое представление матрицы  $S$ .

Пояснить:

- из каких элементов создается матрица ячеек;
- как эти элементы вводятся;
- как выводится графическое представление матрицы ячеек;
- что оно собой представляет.

## 3.6. Задание на самостоятельную работу

Самостоятельное задание рекомендуется для закрепления полученных знаний и включает в себя следующие пункты:

1С. Операции с элементами массива ячеек.

Сформировать массив ячеек *A*, рассмотренный в *разд. 3.2.4*, и привести пример арифметического выражения с элементами данного массива типа `double`.

2С. Операции с матрицами целого типа.

Привести пример выражения с матрицами целого типа.

3С. Операции с матрицами логического типа.

Привести пример логического выражения, в котором все переменные — матрицы.

4С. Операции с матрицами символьного типа.

Сформировать матрицу символьного типа размера  $3 \times 3$  посредством преобразования числовой матрицы типа `double`.

5С. Операции с массивом записей.

Привести пример массива записей для описания трех объектов двумя параметрами, один из которых представлен вектором, а другой — матрицей.

## 3.7. Отчет и контрольные вопросы

Отчет составляется в редакторе Word и содержит результаты выполнения каждого пункта задания, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. К какому типу относятся числовые массивы по умолчанию?
2. Как определить тип массива?
3. Какие типы числовых массивов используются в MATLAB?
4. Как преобразовать матрицу типа `double` в матрицы целых чисел разрядности 8, 16, 32 и 64 со знаком и без знака?
5. Какие операции возможны с матрицами числового целого типа?
6. Как преобразовать числовую матрицу в матрицу логического типа?
7. Что собой представляет матрица логического типа?
8. Какие типы нечисловых массивов предусмотрены в MATLAB?
9. Что собой представляет матрица символьного типа?
10. Что собой представляет массив записей?
11. В каких случаях целесообразно создавать массив записей (структуру)?

12. Что собой представляет матрица ячеек?
13. В каких случаях целесообразно создавать матрицу ячеек?
14. С какой целью выводится графическое представление матрицы ячеек?

|                     |            |
|---------------------|------------|
| Массив              | double, 1  |
| записей, 4          | int16, 1   |
| логический, 2       | int32, 1   |
| определение типа, 7 | int64, 1   |
| символьный, 3       | int8, 1    |
| структура, 4        | logical, 2 |
| тип, 1              | num2str, 4 |
| числовой, 2         | rmfield, 6 |
| ячеек, 6            | single, 1  |
| Функция             | uint16, 1  |
| cellplot, 7         | uint32, 1  |
| char, 3             | uint64, 2  |
| class, 7            | uint8, 1   |