

2. Двумерные массивы чисел: матрицы и векторы

В системе MATLAB все одномерные массивы трактуются либо как вектор-строки, либо как вектор-столбцы, зададим вектор-столбец:

```
b = [ 1; 2; 3]
```

состоящий из трех строк, так как точка с запятой в операции конкатенации означает переход на новую строку.

Для массива *b* функция *length (b)* возвращает число 3, так как действительно этот массив состоит из трех элементов. Функция *length* не различает вектор-строки и вектор-столбцы.

Матрицу *a* размером 3 x 2 (первым указывается число строк, вторым - число столбцов) можно сформировать вертикальной конкатенацией вектор-строк:

```
a = [ [1 2]; [3 4]; [5 6] ]
```

или горизонтальной конкатенацией вектор-столбцов:

```
a = [ [1; 3; 5] , [2; 4; 6] ]
```

Чтобы узнать *размеры* двумерного массива и «геометрию» векторов (вектор-столбцы или вектор-строки), нужно использовать функцию *size*. Для рассмотренного выше двумерного массива *a* получается следующий результат:

```
size( a )
```

```
ans =
```

```
3 2
```

где первым показывается число строк, а вторым - число столбцов.

Теперь применим эту функцию к одномерным массивам. Вот что из этого получается для сформированного выше вектор-столбца *b*, состоящего из трех строк и одного столбца:

```
size( b )
```

```
ans =
```

```
3 1
```

Наконец, попробуем применить эту функцию к переменной, состоящей из единственного числового значения, то есть к скаляру:

```
var = 5;
```

```
size( var )
```

```
ans =
```

```
1 1
```

Вычисления с массивами. В традиционных языках программирования вычисления с массивами осуществляются поэлементно в том смысле, что нужно запрограммировать каждую отдельную операцию над отдельным элементом массива. В М-языке системы MATLAB допускаются мощные групповые операции над всем массивом сразу. Именно групповые операции системы MATLAB позволяют чрезвычайно компактно задавать выражения, при вычислении которых реально выполняется гигантский объем работы.

Для поэлементного перемножения и поэлементного деления массивов одинаковых размеров применяются операции, обозначаемые комбинациями двух символов: *.* * и *./*. Использование комбинаций символов объясняется тем, что символами * и / обозначены специальные операции линейной алгебры над векторами и матрицами.

Кроме операции *./*, называемой *операцией правого поэлементного деления*, есть еще *операция левого поэлементного деления* *.*. Объясним разницу между этими операциями. Выражение *A./B* приводит к матрице с элементами *A(k,ra) /B(k,m)*, а выражение *a.\B* приводит к матрице с элементами *B(k, m)/A(k, m)*.

Знак * закреплен за перемножением матриц и векторов в *смысле линейной алгебры*.

Типичные задачи *аналитической геометрии в пространстве*, связанные с нахождением длин векторов и углов между ними, с вычислением скалярного и векторного произведений, легко решаются разнообразными средствами системы .MATLAB. Например, для нахождения *векторного произведения* предназначена специальная функция cross:

```
u = [ 1 2 3]; v = [ 3 2 1];
```

```
cross( u, v )
```

```
ans =
```

```
-4 8 -4
```

Скалярное произведение векторов вычисляется с помощью функции общего назначения sum, вычисляющей сумму всех элементов векторов (для матриц эта функция вычисляет суммы для всех столбцов). Скалярное произведение, как известно, равно сумме произведений соответствующих координат (элементов) векторов. Таким образом, выражение

```
sum(u .* v )
```

```
ans =
```

```
10
```

вычисляет скалярное произведение двух пространственных векторов (имеющих по три координаты) u и v, которое равно 10.

Длина вектора вычисляется с помощью скалярного произведения и функции извлечения квадратного корня:

```
len1 = sqrt( sum( u .* u ) );
```

Угол между векторами легко вычисляется на основе определения скалярного произведения, гласящего, что оно равно произведению длин векторов на косинус угла между ними. Отсюда находим выражение для вычисления угла между ранее заданными векторами u и v:

```
len1 = sqrt(sum(u .* u)); len2 = sqrt(sum(v .* v));
```

```
phi = acos ( sum(u .* v) / ( len1 * len2 ) )
```

```
phi = 0.7752
```

Функция find. Функция find определяет индексы элементов массива, которые удовлетворяют заданному логическому условию. Как правило, она используется с целью создания шаблонов для сравнения и образования массивов индексов. В наиболее употребительной форме функция k = find(x <условие>) возвращает вектор индексов тех элементов, которые удовлетворяют заданному условию.

```
A=magic(4)
```

```
A =
```

```
16 2 3 13
```

```
5 11 10 8
```

```
9 7 6 12
```

```
4 14 15 1
```

```
k=find(A>8);
```

```
A(k)=100
```

```
A =
```

```
100 2 3 100
```

5	100	100	8
100	7	6	100
4	100	100	1

Функция вида $[i, j] = \text{find}(x)$ позволяет получить индексы ненулевых элементов прямоугольного массива. Функция вида $[i, j, s] = \text{find}(x)$ возвращает, кроме того, и их значения в виде вектора s .