

MATRICES AND VECTORS

A matrix ($m \times n$) with m rows and n columns, a column vector ($m \times 1$) with m rows and 1 column, and a row vector ($1 \times m$) with 1 row and m columns all can be used in MATLAB. Matrices and vectors, when defined, are placed in brackets. To create a row vector simply use a space to separate elements. To create a column vector separate the elements with a semicolon.

```
>> p=[1 4 6 7 8]

p =

    1    4    6    7    8

>> q=[1;2;5;1;6;]

q =

     1
     2
     5
     1
     6
```

Vectors can be multiplied, but remember for vectors order of multiplication is important. For our vectors, $p*q$ will yield a 1×1 matrix, a.k.a. a scalar, and $q*p$ will yield a 5×5 matrix.

```
>> r=p*q

r =

    94

>> s=q*p

s =

     1     4     6     7     8
     2     8    12    14    16
     5    20    30    35    40
     1     4     6     7     8
     6    24    36    42    48
```

Remember that the inner dimensions must agree. A 5×1 matrix cannot be multiplied by a 5×1 matrix. However, in some cases instead of performing a vector multiplication, we

MATLAB Tutorial – Matrices & Vectors

might want to multiply individual elements in two vectors by each other. We can do this by using the “.” notation. For this notation, the two vectors must have the same dimensions.

```
>> t=p.*q
??? Error using ==> .*
Matrix dimensions must agree.

>> t=p.*q'

t =

    1    8   30    7   48
```

Note that the apostrophe after q transposed q, converting q from a 5 x 1 to a 1 x 5 vector.

For addition and subtraction, the vectors must have the same dimensions.

```
>> p-q
??? Error using ==> -
Matrix dimensions must agree.

>> p-t

ans =

    0   -4  -24    0  -40

>> p+q
??? Error using ==> +
Matrix dimensions must agree.

>> p+t

ans =

    2   12   36   14   56
```

One tricky operation is the square function. This must be performed element by element for a vector “.”^” instead of just “^” because the “^” notation requires a square matrix.

```
>> p^2
??? Error using ==> ^
Matrix must be square.

>> p.^2
```

MATLAB Tutorial – Matrices & Vectors

```
ans =  
1 16 36 49 64
```

For functions that require an argument in parentheses, e.g. `abs()`, `sqrt()`, etc., it will perform the operations element by element.

```
>> sqrt(p)  
ans =  
1.0000 2.0000 2.4495 2.6458 2.8284
```

Matrices are made up of several vectors put together. A matrix can be defined as a collection of row vectors.

```
>> A=[1 3 5; 2 1 7; 9 0 2;]  
A =  
1 3 5  
2 1 7  
9 0 2
```

To extract elements from A, simply use `A(row,column)`. To obtain the element in the second row and third column, do the following.

```
>> A(2,3)  
ans =  
7
```

You can also extract parts or entire rows and columns with the colon notation. The general form to extract all elements from the a^{th} row to the b^{th} row and the c^{th} column to the d^{th} column is `A(a:b,c:d)`. For $a=2$, $b=3$, $c=1$, & $d=2$, we get.

```
>> A(2:3,1:2)  
ans =  
2 1  
9 0
```

MATLAB Tutorial – Matrices & Vectors

Operations with scalars can be performed with ease as well. For example, subtracting, adding, multiplying, etc. can be done with a scalar.

```
>> A-1  
  
ans =  
  
    0    2    4  
    1    0    6  
    8   -1    1
```

```
>> A*3  
  
ans =  
  
    3    9   15  
    6    3   21  
   27    0    6
```

An individual element in A can be redefined too. Simply redefine that single element and the others will remain unchanged.

```
>> A(2,2)=99  
  
A =  
  
    1    3    5  
    2   99    7  
    9    0    2
```

In this case, the second row, second column element was the only one that changed.

There are two other ways to define vectors, which are particularly helpful when making plots. The first is “`linspace(first,last,n)`”, which creates a row vector with the first element equal to “first”, the last element equal to “last”, and “n” total elements in the vector. The second is using colon notation “`first:increment:last`”, where a row vector is created with the first element equal to “first” and each subsequent element being an increment greater than the previous until the last one, which is less than or equal to “last”. If increment is left out, MATLAB will use a default value of 1.

```
>> linspace(2,10,5)  
  
ans =  
  
    2    4    6    8   10
```

MATLAB Tutorial – Matrices & Vectors

```
>> 2:5:10
```

```
ans =
```

```
2 7
```

```
>> 2:4:10
```

```
ans =
```

```
2 6 10
```

Notice the difference between the two.

Again this is helpful for plotting. Let's say you are going to make a plot of the sine function. You need to define an argument for the sine function in MATLAB. Let's look at the sine function from 0 to 2π with increments of $\pi/4$.

```
>> x=linspace(0,2*pi,9)
```

```
x =
```

```
0 0.7854 1.5708 2.3562 3.1416 3.9270 4.7124 5.4978 6.2832
```

```
>> y=sin(x)
```

```
y =
```

```
0 0.7071 1.0000 0.7071 0.0000 -0.7071 -1.0000 -0.7071 -0.0000
```

```
>> z=cos(x)
```

```
z =
```

```
1.0000 0.7071 0.0000 -0.7071 -1.0000 -0.7071 -0.0000 0.7071 1.0000
```

EXAMPLE

- 1) Using colon notation and the linspace command make vectors that:
 - a. Go from 0 to 10 with an increment of 2
 - b. Go from 4 to 10 with an 7 elements

- a.) The increment is given so the colon notation should be straight forward. For the linspace command there should be $(10-0)/2+1$ elements, which is 6.

MATLAB Tutorial – Matrices & Vectors

```
>> a1=0:2:10;  
>> a2=linspace(0,10,6);
```

b.) Here the number of elements is given, so the linspace command should be straightforward. The increment is $(10-4)/(7-1)$, which is one.

```
>>b1=4:10;  
>>b2=linspace(4,10,7);
```

2) If A is a 4 x 1 column vector and B is a 1 x 4 row vector, what is the size of the matrix resulting from the following calculations:

- a) $A*B$
- b) $B*A$
- c) $A.*B$
- d) A^2
- e) $A.^2$

3) If C is a 4 x 4 matrix and D is a 2 x 4 matrix, what is the result of the following calculations:

- a) $C*D$
- b) $D*C$
- c) $C.*D$

DO IT YOURSELF

1) Create a vector using both colon notation and the linspace command that:

- a) starts from 10 and goes to -10 with increments of 2.5
- b) starts from 25 and goes to 45 with 5 elements