Day 21

- (Concept Question)
- Special operators
- Using a colon (:)
  - Extracting subsets of matrices
  - Creating vectors
  - Creating matrices
- Element-by-element operations using "."
- (Exercises)

Special operators

MATLAB uses special operators to allow shortcuts for matrix and vector operations.

Several things we have done with for loops can be done more easily with the colon operator (:).

We can extract subsets of matrices and vectors and we can create vectors and matrices with ":"

Extracting subsets of matrices

Consider the matrix A below.

The first column has theta in degrees, the second column has cosine of theta, and the third column has sine of theta.

Say we want to create a vector that is just the first column.

A <7x3 double>		
	1	2	3
1	0	1	0
2	15	0.9659	0.2588
3	30	0.8660	0.5000
4	45	0.7071	0.7071
5	60	0.5000	0.8660
6	75	0.2588	0.9659
7	90	0	1

Extracting subsets of matrices - COLUMNS

We can use a colon to "slice" a column out of a matrix.
For example:

```
theta = A(:,1);
```

The colon here means "all the rows"

Column #1

A <7x3 double>		
	1	2	3
1	0	1	0
2	15	0.9659	0.2588
3	30	0.8660	0.5000
4	45	0.7071	0.7071
5	60	0.5000	0.8660
6	75	0.2588	0.9659
7	90	0	1

The variable `theta` will then be just the first column from `A`

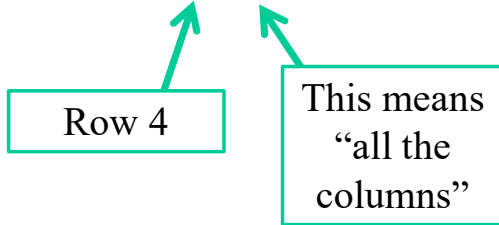| theta <7x1 double> |
|---|
| | 1 |
| 1 | 0 |
| 2 | 15 |
| 3 | 30 |
| 4 | 45 |
| 5 | 60 |
| 6 | 75 |
| 7 | 90 |

Extracting subsets of matrices - ROWS

We can also use a colon to "slice" out a row

For example:

```
theta4 = A(4,:);
```

Row 4

This means "all the columns"

A <7x3 double>	1	2	3
1	0	1	0
2	15	0.9659	0.2588
3	30	0.8660	0.5000
4	45	0.7071	0.7071
5	60	0.5000	0.8660
6	75	0.2588	0.9659
7	90	0	1

The variable `theta4` will then be just the fourth row of `A`

theta4 <1x3 double>	1	2	3
1	45	0.7071	0.7071

Extracting subsets of matrices - RANGES

Instead of using a whole row or a whole columns, we can also use a colon to specify ranges.

For example:

```
Asubset = A(2:5,2:3);
```

`Asubset` will be rows 2 through 5 and columns 2 through 3 of `A`

A <7x3 double>	1	2	3
1	0	1	0
2	15	0.9659	0.2588
3	30	0.8660	0.5000
4	45	0.7071	0.7071
5	60	0.5000	0.8660
6	75	0.2588	0.9659
7	90	0	1

Asubset <4x2 double>	1	2
1	0.9659	0.2588
2	0.8660	0.5000
3	0.7071	0.7071
4	0.5000	0.8660

Creating vectors

We can also use colons to create vectors quickly.

For example:

```
t = [0:4];
```

Creates a vector starting at 0 and ending at 4

t <1x5 double>				
1	**2**	**3**	**4**	**5**
0	1	2	3	4

Creating vectors

We can add a stepsize when we create vectors also:

For example:

```
t = [0:0.5:4];
```

The number in the middle is a step size.

This gives us a vector from 0 to 4 with a step size of 0.5.

t <1x9 double>								
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0.5000	1	1.5000	2	2.5000	3	3.5000	4

Creating matrices

We can combine vectors that we create this way into a matrix.

This gives us a quick way to create the A matrix that we used as an example today.

```
1 -    theta = [0:15:90]';
2
3 -    costheta = cosd(theta);
4 -    sintheta = sind(theta);
5
6 -    A = [theta,costheta,sintheta];
```

Creates a vector, theta, from 0 to 90 with step size of 15

The cos and sin functions will take vectors as inputs—and return vectors as outputs

This line combines the 3 vectors into a single matrix, A.

Element-by-element operations

If we want to multiply the elements of two vectors together, we need to use ".*" instead of just "*" for multiplication.

For example:

```
Command Window
>> a=[1:5]

a =

     1     2     3     4     5

>> b=[2:6]

b =

     2     3     4     5     6

>> a.*b

ans =

     2     6    12    20    30
```

The ".*" here tells MATLAB to multiply a and b "element by element"

4*5 = 20

Element-by-element operations

Element by element operations can be used with *, / or ^.

For example:

```
Command Window
>> a./b

ans =

    0.5000    0.6667    0.7500    0.8000    0.8333

fx >> |
```

```
Command Window
>> a.^2

ans =

    1     4     9    16    25

fx >> |
```