

Array Manipulation

To build an array or matrix you can use the following term.

`AV=[1 2 3 4 5 6 7 8 9]` for a vector

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

or

`AM=[1 2 3; 4 5 6; 7 8 9]` for a 3x3 matrix

Now `AV` holds a 1x9 vector of the number 1-9,

1	2	3
4	5	6
7	8	9

`AM` holds a 3x3 matrix of the number 1-9.

A few commands;

`AV+1` will give you the numbers 2-10.

`sum(AV)` will give you 45.

`sum(AM)` will give you 12,15,18 this is because `sum` sums each column in turn.

`sum(sum(AM))` will give you 45.

The transpose is `AV'` or `AM'`

`AV(6)` gives you the sixth value in `AV`, in this case it will be 6. `AM(6)` will give you the sixth value of `AM`, which in this case is 8, because Matlab counts down each column first. Because `AM` is a matrix we can address each cell using its coordinates such as `AM(2,2)` which is 5.

Loops

There are two types of loops you could use in Matlab both perform the same task a number of times.

The FOR loop

The for loop is used to perform a number of actions a specific number of times.

```
AM=[1 2 3; 4 5 6; 7 8 9];
AN=[];
%Build a matrix and a blank variable
[x y]=size(AM);
%calculate the size of the matrix
%and store the width in x and the
%height in y

for i=1:x           %first loop from 1 to x
    for j=1:y       %second loop from 1 to y
        AN(i,j)=AM(i,j)+AM(x-i+1,y-j+1);
        %add the opposite number of the
        %matrix to its self and store
        %it in AN
    end
end
disp(AN);
```

The WHILE loop

The While loop is used if you want to perform an action until a condition is met.

```
a=0.000001;
x=1;
y=1;
%Set up the variable
while x > a % the loop will continue until a < x
    x=1/(y^2); % calculate a new x
    y=y+1;    %change the conditions this is
             %very important a while loop
end
disp(y);    %show the result
```

Conditions

To make decisions in your code you will need to use conditional statements such as IF this then ... ELSE ...

The IF statement will process the next section of code or if the statement is not true execute the code defined by the ELSE statement

```
IF condition is true    if A >= B
Do this                  disp(A);
ELSE                    else
Do this                  disp(B);
END                      end
```

Operator Meaning

>	greater than
<	less than
>=	greater than or equals
<=	Less than or equals
==	equal to
~=	not equal to

Operator Meaning

	or
&&	and
~	not

Finally

`random('unid',A,X,Y);`

Will give you a X by Y matrix filled with discrete random values from 0 to A.

`find(A==10)`

A really useful command, `find` will return the index (positions in the matrix) which satisfy the conditions passed to it