

Introduction to Matlab

Part 2

Adapted from

<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>

Enter this matrix for use in examples:

<i>Test</i> =	90	100	87	43	20	58
	29	5	12	94	8	62
	75	21	36	83	35	24
	47	51	70	59	82	33

Do this!

Relational operators

You can make comparisons between scalars, between a matrix and a scalar, and between matrices.

The operators that let you make comparisons are as follows: $<$ (less than), \leq (less than or equal to), $>$ (greater than), \geq (greater than or equal to), $==$ (equal), \sim (not equal)

Try this!

Example:

type $3 < 4$

Matlab will display

ans =

1

type $3.5 == 3.5$

Matlab will display

ans =

1

type $10 \geq 7$

Matlab will display

ans =

0

Note: 0 means FALSE, 1 (or any non-zero) means TRUE

Relational operators cont.

Example:

type *Test* $\sim=$ 21

Matlab displays

ans =

1	1	1	1	1	1
1	1	1	1	1	1
1	0	1	1	1	1
1	1	1	1	1	1

type *small* = *Test* <25 & *Test* >=5

Matlab displays

small =

0	0	0	0	1	0
0	1	1	0	1	0
0	1	0	0	0	1
0	0	0	0	0	0

Note: When comparing a matrix and a scalar or two matrices, Matlab returns a matrix of the same size, filled with 1's and 0's indicating where the statement is true, and where it's false. Comparisons between matrices can only be made between matrices OF THE SAME SIZE.

Relational Operators cont.

Results from a relational query can then be used in further statements: you can use a *matrix* of true/false (1/0) entries to reference another matrix!

Example: type $Test(smaller) = 1000$

Matlab displays

Try this!

Test =

90	100	87	43	1000	58
29	1000	1000	94	1000	62
75	1000	36	83	35	1000
47	51	70	59	82	33

NOTE: To NOT have Matlab display the results of every statement, follow the statement with a semicolon (;)

type $Test(smaller) = 0;$ Matlab goes to the next prompt ">>"

type $Test$

Matlab will display Test as above, with all 1000's replaced by 0's.

Boolean operators

You can use Boolean operators to ask for more than one condition (statement) to be valid at the same time.

These operators are: $\&$ (AND), $|$ (OR), and \sim (NOT)

If you join two statements with $\&$, they must both be true for the overall statement to be true. E.g. $A \& B$ - A, B must both be true

If two statements are joined with $|$, one or both must be true for the statement to be true. E.g. $A | B$, A, B, or A and B must be true.

The \sim operator negates any statement. So, if $A \& B$ is true, $\sim(A \& B)$ is false.

If and for statements

The **if** statement allows you to execute some series of steps if a condition is true, or a different series if the condition is false. **For** statements allow you to repeat a series of statements some number of times.

The syntax for the if statement is as follows:

```
if logical statement  
    statements to be executed  
end
```

The syntax for the for statement is similar:

```
for counter = starting value: increment: ending value  
    statements to be executed  
end
```

NOTE: The statements inside an if statement or for loop are usually indented for readability

If statements cont.

Example:

```
Type if Test(4,4)>0  
    display('Condition is true')
```

End

Matlab displays

ans =

Condition is true

```
type if Test(1,1)>0 | Test(2,4)==0  
    display('TRUE')
```

else

display('FALSE')

end

Matlab displays

ans =

TRUE

Now, change the conditions in the second example so that the result is 'FALSE'.

Try this!

NOTE: Else and elseif can be used to extend if statements - if the initial condition is false, the else condition will execute. If the initial condition is false, the elseif condition will execute if its logical statement is true.

For statements cont.

Example:

```
Type for k = 1:10  
    k  
end
```

Matlab displays

k =

1

k =

2 and so on, through k = 10.

```
Type for counter = 2:2:10
```

```
    counter
```

```
end
```

Remember - the colon operator indicates every integer between the start and end OR
start : increment : end

notice that counter increments by 2 with each time thru the loop

Try this!

Scripts

Instead of typing commands in to Matlab at the command prompt, you can put them in an M-file, and run them by typing the file name instead of retyping the commands each time.

To start a new M-file (simply, a file with a .m extension), type *edit &* in the Matlab command window.

A new window will open that will let you edit a script file

To edit an M-file you've created previously, type *edit myfile.m &*

Once you've finished entering commands, and saved the file, at the Matlab prompt, you simply type the filename e.g. *myfile* and the commands within that file will execute.

Scripts cont.

Example:

At the Matlab prompt, type *edit* &
Type the following into your file:

```
Mymatrix = [1:5; 14:2:22; 50:-7:22];  
for i = 1:3  
    for j = 1:5  
        if Mymatrix(i,j) < 5 | Mymatrix(i,j) > 20  
            newmatrix(i,j) = Mymatrix(i,j);  
        elseif Mymatrix(i,j) == 20  
            newmatrix(i,j) = 100;  
        else  
            newmatrix(i,j) = 0;  
        end  
    end  
end
```

*%use the percent sign to
%indicate comments*

*%semicolon means
%Matlab won't display*

Scripts cont.

Once you have typed everything in, save and close the editor

To run your file, simply type *myfile* at the Matlab prompt

Once it's complete, type *newmatrix* to see the results.

If you want to go back and edit the file, you can now type *edit myfile.m*

After this tutorial, you should be able to:

- Compare scalars and matrices using relational operators: $>$, \geq , $<$, \leq , $==$ and $\sim=$
- Combine statements using Boolean operators: $\&$ (AND), $|$ (OR), and \sim (NOT)
- Use if statements to execute a series of steps only if a condition is true
- Use a for loop to execute statements a given number of times
- Put statements into an M-file to run later