

## **Labs for the MA2895 Matlab assessments**

MA2895\_CB=Numerical Analysis Project. It is worth 10 credits.

The class test is currently scheduled for Monday in week 22 and will be worth 30%.

The assignment tasks are worth the other 70% with a deadline likely to be at the start of week 28.

In the labs you will be learning and revising Matlab, practising Matlab, getting ready for the assessments and working on the assignment when you have your individual version.

---

It is very important to practice.

## Week 17 tasks

1. Run the set-up instructions in section 1.1. You will get a MA2715 icon and some folders including

`h:\Matlab_Level12\`

This is a possible folder for your Matlab files in level 2.

2. You will practice the create, edit, run cycle in section 2.
3. You will get some practice with matrices, see section 3.
4. You will be using again `if-else` constructions, see section 4.3.
5. You will use `break` and `continue` statements within loops, see section 4.4 which is about `for`-loops.

## The create, edit, run cycle

Usually you create your commands in an editor and save as a file with a name ending in `.m`. For example you might have the following.

---

```
format compact
A=[4, 1, 1; 1, 4, 1; 1, 1, 4]
y=[1; 2; 3]
b=A*y
x=A\b
```

---

You can then run the block of commands in one go.

You further edit if you need to change things in some way and you re-run and repeat the process until you have achieved what you aimed to do. In particular your work is saved.

# Matrices and how things are stored

---

```
A=[1, 2, 3; 4, 5, 6; 7, 8, 9];
```

```
A
```

```
A(:)'
```

```
A(6)
```

---

The output generated is as follows.

```
A =
```

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

```
ans =
```

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

```
ans =
```

```
    8
```

The matrix is stored column-by-column to explain why A(6) is 8.

# Making decisions

## if and if--else statements

If you only want something to be done if a condition is true then the syntax is as follows.

---

```
if testcondition
    statements to do if testcondition is true
end
```

---

If you also want something to be done when a condition is not true then the syntax is as follows.

---

```
if testcondition
    statements to do if testcondition is true
else
    statements to do if testcondition is false
end
```

---

## Examples of for-loops

---

```
for k=1:8
    fprintf('%d, %2d, %3d\n', k, k^2, k^3);
end
```

---

Here 1:8 is the row vector [1, 2, 3, 4, 5, 6, 7, 8] and k successively takes each value in the list.

If we want steps other than 1 then we can have the following.

---

```
for k=1:2:8
    fprintf('%d, %2d, %3d\n', k, k^2, k^3);
end
fprintf('\n')
for k=8:-3:1
    fprintf('%d, %2d, %3d\n', k, k^2, k^3);
end
```

## Comments about fprintf

```
fprintf('%d, %2d, %3d\n', k, k^2, k^3);
```

Here the following is the specification of the output format.

```
'%d, %2d, %3d\n'
```

There are 3 parts starting with % and there are 3 numbers to display.

$k$  is displayed according to the part %d.

$k^2$  is displayed according to the part %2d.

$k^3$  is displayed according to the part %3d.

In the part %3d the width is 3 and d means decimal digits.

\n gives a new line.

## `break` and `continue` statements in loops

`break` and `continue` statements can only appear in loops.

If a `break` statement is executed then you leave the loop.

If a `continue` statement is executed then you jump to the next case of the loop.

In section 4.4 the `break` statement is used in an example to leave a loop when a Newton iteration has converged.

In section 4.4 the `continue` statement is used in an example to skip statements which are no longer of interest in the current stage of the loop.