MA1710: Key points in week 2 Matlab session

Start by experimenting with `for`–loops.

Create the following with the editor and run.

```
for k=1:5
  disp('What does this do?')
end
```

```
for i=2015:-1:2011
  disp('What does this do?')
end
```

The same output in both cases.

Use the variable in the loop and use `fprintf`

Next you try the following.

```
for k=1:5
  fprintf('k=%d\n', k)
end
```

```
for i=2015:-1:2011
  fprintf('i=%d\n', i)
end
```

These illustrate that the variable can be used.

The `for`–loop syntax

```
for variable_name=list_of_values
  Instructions to do for each value in the list.
  The instructions typically use variable_name.
end
```

Examples to evaluate sums

To compute

$$\sum_{n=1}^{1000} \frac{1}{n^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{1000^2}.$$

we can use the following statements.

```
s=0;
for n=1:1000
  s=s+1.0/(n*n);
end
```

`s` varies in the loop and at any intermediate stage is stores the sum of the terms considered so far.

Computing a product using a `for`–loop

$$10! = 1 \times 2 \times 3 \times \cdots \times 9 \times 10.$$

This can be compured with the following statements.

```
n=10;
p=1;
for r=2:n
  p=r*p;
end
```

The mechanism is similar to what is done to compute a sum. Here, at any intermediate stage, `p` is the product of the numbers considered so far.

The `if`–statement

```
if  logical_condition
  Statements to do if the condition is true.
end
```

As some of the statements to solve a quadratic you might have the following.

```
d=b*b-4*a*c;
if d>=0
  disp('The quadratic has real roots')
end
```

The `if-else` construction

```
if  logical_condition
  Statements to do if the condition is true.
else
  Statements to do if the condition is false.
end
```

As some of the statements to solve a quadratic you might have the following.

```
d=b*b-4*a*c;
if d>=0
  s=sqrt(d);
  x1=(-b-s)/(2*a);
  x2=(-b+s)/(2*a);
else
  fprintf('d=%e,', d);
  fprintf(' the quadratic has complex roots\n');
end
```

Factorials and a `break` statement

You can leave a loop before the end with a `break` statement and usually this will involve a test which has the reason for leaving the loop. An example of using `break` is as follows.

```
for n=1:30
  v=factorial(n);
  fprintf('n=%2d, n!=%14d=%22.14e\n', n, v, v);
  if v>=1e12
    break;
  end
end
```

Here `factorial` is a Matlab function. In this case the `break` statement is executed the first time that a factorial exceeds 10^{12}.