Matlab/Freemat/Octave/Scilab: Recursion

In computer science, *recursion*[1] is the facility which allows a function (i.e. any type of *module*) to call itself. Recursion is a very useful facility in computer programming and it is available in Matlab/Freemat/Octave/Scilab. In this document three typical examples are used to demonstrate recursion in Matlab/Freemat/Octave/Scilab. In each of the examples the function is placed in an M-file[2].

Example 1 : Hello World

Consider the following function in the file helloworld.m. Note that it is recursive because the function is called helloworld() and it calls helloworld() from within its body.

```
function helloworld()
'Hello World'
helloworld()
```

When helloworld() is called from the main program the text 'Hello World' is printed continually (actually the computer complains of 'stack overflow' and gives up after a while). As illustrated by the example above, in practice we find that a recursive function should have a stopping condition (if the program is required to complete a process).

In the following example, the factorial[3] of a number is determined using a recursive method.

Example 2: Factorial function

```
function [fact]=factorial(n)
if (n==1) fact=1;
else fact=n*factorial(n-1);
end
```

The method uses the useful relationship:

$$n! = n(n-1)!\,;$$

the factorial is written in terms of the factorial of a smaller number. And the stopping condition $1! = 1$is also included.

[1] Recursion
[2] Matlab/Freemat/Octave/Octave: Functions: M-files
[3] Factorial

Example 3: Fibonacci sequence

As a third example, we consider a recursive algorithm for computing a term of the Fibonacci sequence[4]. The Fibonacci sequence is as follows

1, 1, 2, 3, 5, 8, 13, 21, ...;

It starts 1, 1 then each of the following terms is determined as the sum of the previous two terms.

```
function [fib]=fibonacci(n)
if (n<=2) fib=1;
else fib=fibonacci(n-1)+fibonacci(n-2);
end
```

The method uses the recurrence relationship:

$$f_n = f_{n-1} + f_{n-2}\ ;$$

the Fibonacci number of index *n* is written in terms of two earlier Fibonacci terms.

For example Fibonacci(10) returns the value 55.

[4] Sequences