1 Help on functions

A function-file always starts with the line 'function [output]=functionname(input)' and implies a subroutine call where only the specified input and output variables are exchanged. All variables in the function are local, which means the function cannot access MATLAB workspace variables that you don't pass to it. After the execution of the function, you cannot access the internal variables, only the specified output of the function.

Example: The following function called '*mult*' has the two vectors a and b as input parameters. It computes the elementwise product $a. * b$ and returns it as vector c:

```
function [c]=mult(a,b)
c=a.*b;
```

To use the function, first save it under the filename 'mult.m'. Then, within Matlab type:

```
c=mult(a,b);
```

For a function which has only input but no output parameters, the function header would look like the following:

```
function []=plotx(input)
```

2 MATLAB Help on functions

```
FUNCTION Add new function.
    New functions may be added to MATLAB's vocabulary if they
    are expressed in terms of other existing functions. The
    commands and functions that comprise the new function must
    be put in a file whose name defines the name of the new
    function, with a filename extension of '.m'. At the top of
    the file must be a line that contains the syntax definition
    for the new function. For example, the existence of a file
    on disk called STAT.M with:

        function [mean,stdev] = stat(x)
        %STAT Interesting statistics.
        n = length(x);
        mean = sum(x) / n;
        stdev = sqrt(sum((x - mean).^2)/n);

    defines a new function called STAT that calculates the
```

mean and standard deviation of a vector. The variables
within the body of the function are all local variables.
See SCRIPT for procedures that work globally on the work-
space.

A subfunction that is visible to the other functions in the
same file is created by defining a new function with the FUNCTION
keyword after the body of the preceding function or subfunction.
For example, avg is a subfunction within the file STAT.M:

```
function [mean,stdev] = stat(x)
%STAT Interesting statistics.
n = length(x);
mean = avg(x,n);
stdev = sqrt(sum((x-avg(x,n)).^2)/n);

%-------------------------
function mean = avg(x,n)
%AVG subfunction
mean = sum(x)/n;
```

Subfunctions are not visible outside the file where they are defined.

You can terminate any function with an END statement but, in most
cases, this is optional. END statements are required only in M-files
that employ one or more nested functions. Within such an M-file,
every function (including primary, nested, private, and subfunctions)
must be terminated with an END statement. You can terminate any
function type with END, but doing so is not required unless the
M-file contains a nested function.

Normally functions return when the end of the function is reached.
A RETURN statement can be used to force an early return.

See also script, return, varargin, varargout, nargin, nargout,
         inputname, mfilename.