

Saving output to ASCII text files

```
fid = fopen('<filename>', '<permission>');  
fprintf(fid, '<string>', <list of variables>);  
fclose(fid);
```

<permission> being **w** means that you can “write” to the file; if the file already exists, anything previously contained in it will be overwritten; if the file does not already exist, it will be created.

<permission> being **a** means that you can “append” to the file; if the file already exists, anything previously contained in it is retained and the new data is appended after the pre-existing content; if the file does not already exist, it will be created.

```
save <filename> <variable name> -ascii  
or  
save <filename> <variable name> -ascii -append  
to append the data after any pre-existing content of the file.
```

Iteration (loops)**For loop**

This construct is useful in situations where a set of statements needs to be executed repeatedly and the number of repetitions is known in advance.

```
for <loop counter variable> = <sequence of values>  
    <loop body>  
end
```

Here <sequence of values> is often given in the form **1 : <number of repetitions>** and <loop body> comprises a set of statements to execute

The <loop counter variable> (or loop “index”) steps through the <sequence of values> one by one, and at each step the statements in the <loop body> are executed.

The value of the loop counter variable should never be modified in the loop body.

While loop

This construct is useful in situations where a set of statements needs to be executed repeatedly but the number of repetitions is not known in advance. It is appropriate to use a while loop if we have some criterion for controlling the error of the approximation.

```
while <boolean expression>  
    <loop body>  
end
```

Here <loop body> is a set of statements to execute if <boolean expression> is true.

At the beginning, <boolean expression> is evaluated. If it is true, then the loop body is executed; if it is false, then the flow of control passes to the next statement following **end**.

After each pass through the loop, <boolean expression> is evaluated and, as long as it is true, the loop body is executed again.

It is possible to exit a loop early by using the **break** command, usually as part of an **if** structure.