Using MATLAB and M-files

You should create a directory for your own M-files in the MATLABR11 directory. Call it Mfiles, say. You can now change your default directory to this one using the command:
File → Set Path.. . Browse and set the current directory to C:\MATLABR11\Mfiles.

You can now create your own Mfiles. Use the File→ New → M-file command. Type in the program as normal and save it in your directory. It should have an extension .m, to denote an M-file. The function below defines a polynomial, called fun1, save it as fun1.m. It is just 3 lines long.

```
function y = fun1(x)
% define the function y = f(x)
y = x.^2 + 2*x +1;
```

Now use the quad function to integrate it (command window):

```
>> Q= quad('fun1',0,2)
```

This integrates the function from 0 to 2. The answer is Q = 8.6667.
Verify by integrating that the result is correct.

There is a demonstration is MATLAB called `quaddemo` that integrates a function called humps(x). This function has local maxima at x = 0.3 and x = 0.9. A plot of the graph is shown. The function is as follows:

```
y = 1 ./ ((x-.3).^2 + .01) + 1 ./ ((x-.9).^2 + .04) - 6;
```

Q = 29.8583

The humps function and others can be found in the \MATLABR11\toolbox\matlab\demos directory.

```
% Create the Runge function 1/(1+25x^2)

x = -1:0.01:1;
y = 1./(1 + 25*x.*x);
plot(x,y)
pause
% points for poly of degree 5

x5 = -1:0.4:1;
y5 = 1./(1 + 25*x5.*x5);
p5 = polyfit(x5,y5,5);      % fit poly of degree 5
p5                          % print coefficients of poly
% set up points for plotting
xi = -1:0.02:1;
yi = polyval(p5,xi);        % Evaluate poly p5 at points xi
plot(x,y,'r',xi,yi,'g')
pause

% points for poly of degree 10

x10 = -1:0.2:1;
y10 = 1./(1 + 25*x10.*x10);
p10 = polyfit(x10,y10,10);    % fit polynomial of degree 10
zi = polyval(p10,xi);         % Evaluate poly p10 at points xi
plot(x,y,'r',xi,yi,'g',xi,zi,'b')  % plots with red, green blue lines
```