Overview of the assignment m-files and the tasks

1. Getting started task worth 2 marks. Please do this today. In Matlab you type a command with the name ending in `_cr_zip_file`

2. The finding roots of cubics task worth 24 marks (3 files). There are 2 function files with names ending in `_cube.root.m` and `_cardano.m`. You test both in a script file with a name ending in `_test_cubic_iter.m`.

3. The partly graphics task worth 26 marks (1 script file). The script file has a name ending in `_cubic_plot.m`.

4. The finite difference methods task worth 18 marks. (3 files) This involves 2 function files with names ending in `_basic_fdm.m` and `_numerov_fdm.m`. You test both in a script file with a name ending in `_test_fdm.m`.

Functions and where to test them
Illustrating using the secant method

The secant method is a numerical method to attempt to solve $f(x) = 0$ using the iteration

$$x_{k+1} = x_k - f(x_k)\left(\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}\right), \quad k = 1, 2, \ldots$$

If you have a guess x_0 for the solution then you can let $x_1 = x_0 + h$, where h is small, and then use the above to generate x_2, x_3 etc.

A Matlab function file to do this called `secant_meth.m` might have the following as the first line.

```
function x=secant_meth(f, x0, tol, nit)
```

Here the inputs are `f`, which is a function handle, `x0`, which is the guess of the solution, `tol`, which is the used to stop the iteration and `nit`, which is the maximum number of iterations.

The function file

```
function x=secant_meth(f, x0, tol, nit)
x1=x0+sqrt(eps);
f0=f(x0);
f1=f(x1);
for k=1:nit
  x=x1-f1*(x1-x0)/(f1-f0);
  if abs(f1)<tol
    break;
  end
  x0=x1;
  f0=f1;
  x1=x;
  f1=f(x1);
end
```

You should add comments but these do not easily fit on the slide.

Note that `f()` is only evaluated once for each value of `k`.

Testing the function

In another file you could do the following to attempt to find the cube root of 2.

```
g =@(x) x^3-2;
x=secant_meth(g, 2, 1e-10, 20)
v=g(x)
```

Do not attempt to test by doing the following

```
function x=secant_meth(f, x0, tol, nit)
f =@(x) x^3-2;
x=2;
tol=1e-10;
nit=20;
% ..statements as before
```

If this was an assignment task and this was what submitted then the mark for the function file would be zero.

If you arbitrarily reset the values of the input arguments in statements in the function then there was not too much reason to have function in the first place.

Testing comment

If you do not test a function that you have just typed then there may be one or more syntax errors. If you submit a file which will not run due to syntax errors then there will be zero marks for that function and everything that depends on it.

Secant's method for the cube root

A vectorised version of secant's method to compute the cube root of y could involve the following.

```
function x=secant_cube_root(y)
f =@(x) x.^3-y;
x0=sqrt(sqrt(y));
x1=x0+sqrt(eps);
f0=f(x0);
f1=f(x1);

for k=1:20
  x=x1-f1.*(x1-x0)./(f1-f0);
  if norm(f1(:), inf)<1e-10
    break;
  end
  x0=x1;
  f0=f1;
  x1=x;
  f1=f(x1);
end
```

Testing the vectorised secant cube root function

```
y=2:8;
x=secant_cube_root(y)
v=x.^3-y

p=-1+1i;
z=secant_cube_root(p)
v=z.^3-p
```