UNIVERSITY *of* LIMERICK
OLLSCOIL LUIMNIGH

Faculty of Science & Engineering
Department of Mathematics & Statistics

END OF SEMESTER ASSESSMENT PAPER

MODULE CODE: MS4101

SEMESTER: Autumn 2015

MODULE TITLE: Mathematics Laboratory

DURATION OF EXAMINATION: 2 1/2 hours

LECTURER: Dr. J. Kinsella

PERCENTAGE OF TOTAL MARKS: 70%

EXTERNAL EXAMINER: Prof. J. King

INSTRUCTIONS TO CANDIDATES:

- **Complete successfully the tasks described overleaf for full marks — 70%.**
- **You may consult the lecture notes for the course and your own hand-written notes during the assessment.**
- **You will be advised how to upload your work at the end of the assessment.**

- You are asked to solve the following problems using Matlab.
- The tasks are inter-related so you should perform them in sequence.
- Before you begin, start a `diary` file whose name is `123456789.txt` where `123456789` should be replaced by your student ID.

1 Perform the following plotting and scripting tasks. 35

(a) Write a **function** m-file `myfun.m` that inputs x and returns the function $e^x - 3x^2 + 1$, together with its first and second derivatives at x.

- `myfun.m` has a single input parameter `x` and three output parameters; `f`, `fp` and `fpp`.

(b) Write a **script** m-file `Run.m` that performs the following tasks.

(i) Plot the function `myfun` on the interval $[-5, 5]$ with a suitable number of intermediate points.

(ii) Include the x–axis by plotting a horizontal line from $(-5, 0)$ to $(5, 0)$.

(iii) Annotate the plot with the title: **Plot of $\mathbf{e^x - 3x^2 + 1}$ on the interval $[\mathbf{-5, 5}]$**.

(iv) Use LaTeX formatting commands to typeset the parts of the title using mathematical notation.

(v) Label the x–axis and y–axis suitably.

(c) Based on your plot, add a line in your **script** m-file `Run.m` that creates a vector `myroots` which contains the approximate location of the root or roots of `myfun` in the interval $[-5, 5]$.

(d) Add a further line in `Run.m` that creates and displays a string variable `rootstr` which states the approximate location of the roots. (Use the `num2str` function to include the vector `myroots`.)

2 Use the Matlab Symbolic Toolbox `syms` and `vpasolve` commands to find all the roots of the function `myfun` in the interval $[-5, 5]$. 5

(a) Check that the roots found are correct by evaluating `myfun` at each point.

(b) Add the relevant commands to your **script** m-file `Run.m`.

3 Perform the following Matlab programming tasks. 60

Halley's Method (similar to Newton's Method) computes a succession of increasingly accurate estimates of a root of a specified function. The update rule is:

$$x_{new} = x - \frac{2f(x)f'(x)}{2[f'(x)]^2 - f(x)f''(x)}.$$

(a) Write a **function** m-file `Halley.m` that uses this update rule to approximate a root of a specified function.

(b) `Halley.m` should have input parameters:

- `fun` (the name of the function m-file containing the function whose root is to be found).
- `x0` (the starting point).
- `acc` (the required accuracy).

(c) `Halley.m` should have output parameter `r`, the root found.

(d) `Halley.m` should use a `while` loop to apply the update rule above until the difference between successive x-values is less than the value of the input parameter `acc`.

(e) Amend your stop criterion so that the `while` loop also checks for `abs( fun(x)) < acc` and stops if this occurs.

(f) Now add a line to your **script** m-file `Run.m` that calls the **function** m-file `Halley.m`.

(i) Use the value `@myfun` for the input parameter `fun`.

(ii) Choose values for `x0` based on your plot of `myfun`.

(iii) Take the value `1.0e-8` for the input parameter `acc`.

- Run your **script** m-file `Run.m`.
- Close your diary file.
- Create a `zip` file named `123456789.zip` (where `123456789` should be replaced by your student ID) containing all the files in your current working folder.
- Upload your zip file when asked to do so at the end of the assessment.