Numerical Methods (NUM101) — Coursework 3

This coursework consists of one part. it is worth 17% of the credits for this unit. The maximum number of marks for this coursework is 17.

Deadline	hand-in or upload?
12 May (Thu)	10:30 upload to Victory Assignment `CW3` **no** hand-in of printouts at CAM office **(changed!)**

Instructions and rules

- **(Changed!)** No material to be handed in to CAM.
- Material to be uploaded to Victory: function files `AppEval.m`, `AppDeriv.m`, `AppInt.m`, and, if you need them, function files of other functions that you call inside your main functions.
- Scripts and functions that generate Matlab errors receive provisionally 0 marks. The student then has to demonstrate in the lab session that the error is minor to justify partial credit.
- Credit:

  100% code performs computation correctly and efficiently, is well structured and commented;

  ≥80% code performs computation correctly and efficiently

  ≥60% code performs computation correctly but has problems[1];

  ≥40% code does not perform computations correctly but could be made to work with minor corrections;

  ≥20% the intentions behind the code are discernible with some effort.

- This is **individual** coursework. In cases where **two students collaborate** they must declare this at the top of their script and write on their printout "**Collaborated with xxxyyy**". (This must stand out visually!). These submissions are subject to particularly strict scrutiny for problematic constructions[2].
- For questions, clarifications and further help contact:

  Jan Sieber (`jan.sieber@port.ac.uk`, office LG.146).

[1] for example, the function works correctly most of the time but fails for some valid arguments

[2] more examples of problematic constructions (look also for warnings in the Matlab editor):

- hard-coded 'magic' numbers spread throughout the code,
- functions that should be general but only work for this example,
- one part of the code is a repetition of another part,
- stray brackets, misleading variable names or variable usages (say, using `x(:)` if `x` is scalar),
- arrays grow inside a loop,
- a variable is defined but not used.

Question 1: Numerical calculus — finding the values, derivatives and integrals of unknown functions

Write three functions, `AppEval`, `AppDeriv` and `AppInt` that approximate the value, the derivative and the integral of an unknown function f for which you are given only some unevenly spaced values. More specifically, the function `AppEval` has to be written in the file `AppEval.m`, and its first line looks like this:

```
function y=AppEval(x,xvals,yvals)
```

For any `x` from the interval $[0,1]$ it should return (an approximation of) the function value $y = f(x)$. You do not know the formula for f, but `xvals` and `yvals` are sample vectors of length approximately 120 (you cannot rely on the exact length but it will be larger than 100). For each element of `xvals`, the function value is in `yvals`: `yvals(i)` is `f(xvals(i))`. Note that the values in `xvals` cover the entire range from -0.1 to 1.1 but they are slightly unevenly spaced in a random manner.

Similarly, the function `AppDeriv` has to be written in the file `AppDeriv.m`, and its first line looks like this:

```
function y=AppDeriv(x,xvals,yvals)
```

For any `x` from the interval $[0,1]$ it should return (an approximation of) the derivative $y = f'(x)$ of the unknown function f. Again, `xvals` and `yvals` are sample vectors (same condition as for `AppEval`).

The function `AppInt` has to be written in the file `AppInt.m`, and its first line looks like this:

```
function y=AppInt(a,b,xvals,yvals)
```

For any `a` and `b` (you can assume $0 \le a < b \le 1$) it should return (an approximation of) the integral $\int_a^b f(x)\,\mathrm{d}x$. Again, `xvals` and `yvals` are sample vectors (same condition as for `AppEval`).

Total for Question 1: 17 marks

Hints and further instructions

- **Marking scheme** is in Table 1. The entries in the table mean that, for example, if the difference between `AppEval(x,xvals,yvals)` and the true $f(x)$ is less than 10^{-4} for all functions f and all values $x \in [0,1]$ that I test then you will get 2 points for `AppEval`. Similarly, you can read off the poitns awarded for other accuracies and for `AppDeriv` and `AppInt`. Note that the number of points for the same accuracy is higher for `AppDeriv`, and that there is a gap from 10^{-6} to 10^{-10} for the highest score for `AppEval` and `AppInt`.
- If you use Matlab's built-in interpolation functions (`spline`, `interp1`, `polyfit` and the like) your score will be halved (and rounded down).
- You find a Matlab function file `GenerateValues.m` in the Victory folder for `Coursework 3`. This function generates examples of `xvals` and `yvals`. I will use these samples for testing your functions. Download the file into your folder and call it (for example, on the commandline or in your own testing script) like this:

Question 1 continues overleaf

```
[xvals,yvals]=GenerateValues();
```

This defines two (random) vectors `xvals` and `yvals` of valid samples. If you want to have an idea of how the underlying function looks like, you may call

```
plot(xvals,yvals,'.-');
```

to see a plot of the function values. For the purposes of your own testing (to see if you get good accuracy) you can also call `GenerateValues` with a function of your own choice. For example, if you want to test your functions using $y = f(x) = \exp(x) - x$, you can call

```
[xvals,yvals]=GenerateValues(@(x)exp(x)-x);
```

Then the `xvals` will be random but the `yvals` will be the function values corresponding to `xvals`. In this way, you know, for example, that after

```
y=AppEval(x,xvals,yvals);
```

`y` must approximate `exp(x)-x`. This helps you estimate the accuracy of `AppEval`.

	points for e_{eval}	points for e_{deriv}	points for e_{int}
$\leq 10^{-1}$	0	1	0
$\leq 10^{-2}$	0	2	0
$\leq 10^{-3}$	1	3	2
$\leq 10^{-4}$	2	4	3
$\leq 10^{-5}$	3	5	4
$\leq 10^{-6}$	4	6	
$\leq 10^{-10}$	5		6

where

$$e_{\text{eval}} = \left|\texttt{AppEval(x,xvals,yvals)} - f(x)\right|$$
$$e_{\text{deriv}} = \left|\texttt{AppDeriv(x,xvals,yvals)} - f'(x)\right|$$
$$e_{\text{int}} = \left|\texttt{AppInt(x,xvals,yvals)} - \int_a^b f(x)\,\mathrm{d}x\right|$$

Table 1: Marking scheme