Numerical Methods (NUM101) — Optional Coursework `numdiff5`

This coursework consists of one part. It is worth 17 marks which will replace your worst coursework mark of the three regular courseworks (overall 17% of the credits for this unit).

Deadline	hand-in or upload?
18 May (Wed)	22:30 upload to Victory Assignment `numdiff5` **no** hardcopy to CAM
18 May (Wed) or before	demonstration of working demo script and function in lab to lecturer

Instructions and rules

- Material to be uploaded to Victory: function file `numdiff5.m`, working demo script `demo.m`, and, if you need them, function files of other functions that you call inside your main function.
- Marks will be awarded for your work only if you demonstrate the usage and the inner workings of your solution to the lecturer in the lab.
- 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. **No declared collaboration is permitted**.
- 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. Other 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: Higher order derivatives

Write a function `numdiff5` that approximates the **fifth** derivative of an unknown function f. The function `numdiff5` has to be written in the file `numdiff5.m`, and its first line looks like this:

```
function y=numdiff5(f,x)
```

For any `x` it should return (an approximation of) the fifth derivative $y = f^{(v)}(x)$ of the unknown function f. You can assume that the function f can be evaluated everywhere.

Total for Question 1: 17 marks

	points for $e_{\texttt{numdiff5}}$
$\leq 10^{-2}$	3
$\leq 10^{-3}$	5
$\leq 10^{-4}$	7
$\leq 10^{-5}$	9
$\leq 10^{-6}$	11
$\leq 10^{-7}$	13
$\leq 10^{-8}$	15
$\leq 10^{-9}$	17

where

$$e_{\texttt{numdiff5}} = \left|\texttt{numdiff5(f,x)} - f^{(v)}(x)\right|$$

Table 1: Marking scheme

Hints and further instructions

- **Marking scheme** is in Table 1. The entries in the table mean that, for example, if the difference between `numdiff5(f,x)` and the true $f^{(v)}(x)$ is less than 10^{-4} for all functions f and all values x that I test then you will get 7 points
- 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 `GenerateFunc.m` in the Victory folder for `numdiff5`. This function generates examples of functions f. I will use these samples for testing your `numdiff5`. Download the file into your folder and call it (for example, on the commandline or in your own testing script) like this:

  ```
  f=GenerateFunc();
  ```

  This defines a function `f` for which

  ```
  y=numdiff5(f,x);
  ```

  should return the $y = f^{(v)}(x)$. You can call

  ```
  fplot(f,[-10,10]);
  ```

  to see a plot of the function.