Numerical Methods (NUM101) — Coursework 2

This is the second of three courseworks which combine for 50% of the credits for this unit. The maximum number of marks for this coursework is 10, worth 16% of the overall credits. This coursework consists of **one** question. Parameter h of the coursework problem is **personalised**.

Instructions and rules

- Deadline: 26 April 2010.
- Material to be handed in:
  - (**Hardcopy**) A printed document containing the output generated by your script `TestMySecant.m` (see below for instructions) and comments (no essay!),
  - (**Hardcopy**) printout of program codes with comments.
  - (**Victory**) Upload two `m`-files, `MySecantIteration.m` and `TestMySecant.m` to the Victory assignment `CW2`.

  The working Matlab files (function and script) account for 70% of the credits. The properly formatted output of the hardcopy (see question) accounts for 30%, the remainder of the printout is only there to show the code, and give additional comments. The output may also show the result in a graph but this is optional.
- Credit for code part of question:

  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. If you copy code from other students you will be turned in for plagiarising.
- For questions, clarifications and further help contact:

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

[1] Examples for problematic code:
- code works correctly most of the time but fails for some valid arguments;
- code calls user function `f` unnecessarily often;
- magic numbers spread throughout the code;
- one part of the code is a repetition of another part.

Question 1: Testing the secant iteration

(a) Write a function `MySecantIteration` that can perform a secant iteration for any given function f. The secant iteration finds roots of f by applying the iteration

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k).$$

The first line of the function file `MySecantIteration.m` should look like this:

```
function [x,converged,res]=MySecantIteration(f,x0,x1,maxit,tol)
```

Inputs:

- `f`: user-supplied function f for which the root is sought,
- `x0,x1`: initial guesses for the iteration,
- `maxit`: maximum number of iterations to be performed,
- `tol`: tolerance. The function should return as soon as the modulus of the residual $|f(x_k)|$ of the most recent iterate is less than `tol`.

Outputs:

- `x`: final iterate (the root of f if the flag `converged` is `true`),
- `converged`: flag indicating success. If `converged==true` convergence was achieved (up to tolerance), otherwise no convergence was achieved within `maxit` iterations.
- `res`: a vector containing all residuals $f(x_k)$ at the iterates x_k.

[5 marks]

(b) Write a script `TestMySecant.m` that tests your function `MySecantIteration` with several examples: use the 4 functions

$$f_n(x) = \mathrm{e}^x - x^2 - x - 1 - nh$$

where $n = 1,2,3,4$ and h is your Jupiter number with a "0." prefix (that is, if your Jupiter number is 434,343 then $h = 0.434\,343$). Use

```
maxit=20;
tol=1e-8;
x0=1;   % initial guess 1
x1=1.2; % initial guess 2
```

as the other parameters of `MySecantIteration`.

For each n print the value of nh, the solution `x`, the convergence flag `converged` and the final 6 residuals returned by `MySecantIteration`. [2 marks]

(c) Theoretically, the residuals $r_k = f_n(x_k)$ of the secant iteration obey the formula

$$r_{k+1} = C r_k r_{k-1} + O(r_k r_{k-1}^2)$$

where the pre-factor C depends on f_n. Calculate estimates for C by using the final residuals returned by `MySecantIteration`:

```
C=res(end)/(res(end-1)*res(end-2));
```

and print the difference of this estimate to the theoretical value of C in the limit for small residuals (which you have to find out theoretically). [3 marks]

Total for Question 1: 10 marks