Nonlinear Problems in applications (MTH311) — Coursework 1

This is the first of two courseworks which combine for 100% of the credits for this unit. The maximum number of marks for this coursework is 35, worth 35% of the credits. The second coursework is worth 65 marks and 65% of the credits. This coursework consists of **one** question. Parameter `p(2)` (see Question) is **personalised**.

Instructions and rules

- Deadline: 16 April 2010.
- Material to be handed in:
  - (**Hardcopy**) Printed document containing graphs and solutions with necessary headings, annotations and comments (no essay!),
  - (**Hardcopy**) printout of program codes with comments.
  - (**Victory**) Upload a script `cw1.m` and all functions (also `m`-files) necessary to run the script to the Victory assignment `CW1`. When I run the script `cw1.m` it should recover all graphs and numerical outputs of your document.

  The working scripts and functions account for 80% of the credits. The hardcopy (20%) is only there to show the output, the code, and give additional comments.
- Credit for the coding part of each question:

  100% code performs all computations correctly, is well structured and commented;
  ≥80% code performs all computations correctly but has problems with its structure[1];
  ≥60% code performs most of the computations correctly;
  ≥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 are unable to implement some of the functions or scripts you may use a copy of these functions' or scripts' `m`-files from other students **if you declare at the top of your coursework document** which `m`-files are borrowed. You will then get zero credit only for these functions. If you do not declare 'borrowed code' but use it to get your results then you are plagiarising.

  Working code that you wrote yourself will always get more credit than a neat plot obtained by calling other students' functions.
- For questions, clarifications and further help contact:

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

[1] examples of bad structure:
– 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.

Question 1: A chemical reaction $A \to B \to C$

A chemical reaction involving three substances is governed by a set of nonlinear differential equations:

$$\begin{aligned}
\dot{u}_B(t) &= -u_B(t) + p_1\,[1 - u_B(t)]\,\mathrm{e}^{T(t)} && (1)\\
\dot{u}_C(t) &= -u_C(t) + p_1\mathrm{e}^{T(t)}\,\big[1 - u_B(t) - p_4 u_C(t)\big] && (2)\\
\dot{T}(t) &= -(1+p_2)T(t) + p_1 p_3 \mathrm{e}^{T(t)}\,\big[1 - u_B(t) + p_4 u_C(t)\big]. && (3)
\end{aligned}$$

The quantities u_B and u_C measure the amounts of the substances B and C, and T is a measure for the temperature in the reactor. The parameter p_1 is a measure for the reaction rate $A \to B$ and can be varied (by catalyst). The parameter p_2 is the cooling provided to the reactor (this is also controllable). The values of p_3 and p_4 are given as

$$p_3 = 8 \qquad\qquad p_4 = 0.05. \qquad (4)$$

See below for values of p_1 and p_2.

(a) **Equilibria** Find the equilibria of the reaction (1)–(3) for parameters p_1 in the range from 0 to 0.2 (p_2 is fixed, see **Hints** how to get your personalised value of p_2). Plot three graphs showing the curve of equilibria, one where the y-axis is u_A, one where the y-axis is u_B, one where the y-axis is T (x-axis is always p_1). [20 marks]

(b) **Stability** Indicate the type of each equilibrium along the curves you obtained in part a. For example use a dot for equilibria that are stable, a cross for equilibria that have one unstable eigenvalue, a square for for equilibria that have two unstable eigenvalues (or use colors instead of different symbols). [5 marks]

(c) **Fold** The curve from part (a) should show two folds. Calculate the parameter value p_1 and the values of u_A, u_B and T that are exactly on the fold to five significant digits. [10 marks]

Total for Question 1: 35 marks

Hints:

- You may assume that all equilibria of (1)–(3) lie on a single curve such that one can find them all by continuation.
- On Victory in the folder `Coursework 1` you will find a Matlab function file `abc.m`, which defines the Matlab function `abc(x,p)` where `x`= (u_B, u_C, T) and `p`= (p_1 , p_2, p_3, p_4). This function is the right-hand-side of system (1)–(3), similar to the worked example `ab` from the lecture. You can use this function in your calculations such that you do not have to convert (1)–(3) into Matlab code.
- Also in the folder is a table with your personalised initial values of p_2 and a file `getpar.mat`, which you can load into Matlab. Download `abc.m`, `cw1getpar.mat` and `Cw1init.m` into your current Matlab working directory. Open `Cw1init.m` in the editor, change the line

  ```
  myID='cam12345';
  ```

  to contain your user ID (keep the quotes!) end execute `Cw1init.m`. After this the vector `p` is defined with your personalised value for p_2 ($p_1 = 0$, and p_3 and p_4 are as in equation (4)).

Question 1 continues overleaf

- You should also have the function `abc` available. Try `abc([0;0;0],p)` on the command-line. Beware that the Matlab installation in the Lab may have an annoying bug and claim that it cannot find `abc.m` even though it is in the current working directory (check by typing `ls`).

  You can work around this bug by working on a directory on a USB stick instead of the `N:` drive. If you insist on working on `N:` drive:

  – open `abc.m` into the editor,
  – click on the `Run` toolbar button in the editor,
  – a pop-up window may show up (click on `Change directory`),
  – ignore the error message on the command-line,
  – try again `abc([0;0;0],p)` on the command-line.

  You may have to do this for all newly written or downloaded functions and scripts in your current directory.

- On Victory in the folder `Useful Functions` you will find Matlab functions that are potentially useful, and that you can call as part of your own scripts and functions after downloading them. Beware that they are written by me and only provided for your convenience. This means that they may not give meaningful error messages if you call them with inconsistent arguments. Report difficulties to me.