Nonlinear Problems in applications (MTH311) — Coursework 2

This is the second of two courseworks which combine for 100% of the credits for this unit. The maximum mark for this coursework is 65, worth 65% of the credits. This coursework consists of **two** questions. The parameters `p(2)` in Question 1 and `p(3)` in Question 2 are **personalised**.

Instructions and rules

- Deadline: 14 May 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 two scripts, `cw21.m` and `cw22.m`, and all functions (`m`-files or `mat`-files) necessary to run the scripts to the Victory assignment `CW2`. Running the script `cw21.m` should recover all graphs and numerical outputs of your document for Question 1. Running the script `cw22.m` should recover all graphs and numerical outputs of your document for Question 2.

  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 and efficiently, 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 but fails for some, or is unnecessarily inefficient[2];

  ≥**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.

[2] example for unnecessary inefficiency: un-structured Jacobian in Question 2

Question 1: Fold continuation for $A \to B \to C$ reaction

(*See Coursework 1 instructions for background explanation of the nonlinear system.*) As part (c) of coursework 1 you calculated 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. The other parameters are your personalised value of parameter p_2, and

$$p_3 = 8 \qquad\qquad p_4 = 0.05. \tag{1}$$

Find the curve in the p_1-p_2 plane on which folds of equlibria occur. [15 marks]

Total for Question 1: 15 marks

Hints:

- You may assume that the folds form a single curve in the p_1-p_2 plane and that they are all connected to the first fold you found in part (c) of coursework 1.
- Starting from the initial fold the curve extends in both directions, for initially increasing p_2 and initially decreasing p_2. This means that you may have to run the continuation twice.
- The matlab function file `abc.m`, which defines the Matlab function `abc(x,p)` is still on Victory in the folder `Coursework 1`. You can use this function in your calculations.
- (*You should have done this already for Coursework 1*) In the folder `Coursework 1` 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 (1)). Alternatively, read off your personal values from the table in victory and put the comand

  `p=[`*your values as column vector*`];`

  at the top of your script `cw21.m`.

  You should also have the function `abc` available. Try `abc([0;0;0],p)` on the command-line.
- 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.

Question 2: Resonance of excited beam with magnets

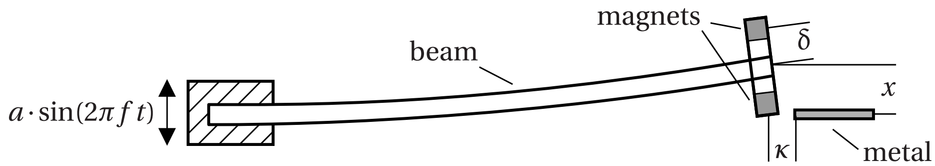

Figure 1: Excited beam with magnets — set-up used in energy harvesting devices

Figure 1 shows a set-up that is used for energy harvesting by putting it on a vibrating platform (say, a helicopter) and putting a coil around the metal strip. The vibration amplitude $x(t)$ is governed by the differential equation

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -Cv - Kx + M(x+\delta) + M(x-\delta) - (2\pi f)^2 a \cdot \sin(2\pi f t). \end{aligned} \tag{2}$$

The parameters are the spring constant, K, the damping, C (which is controlled by taking energy out of the system), the maximum displacement of the platform, a, the vibration frequency, f, the distance of the magnets from the beam, δ, the gap between beam and metal strip, κ, and the strength of the magnetic force per unit distance squared, γ. The magnetic force M has the formula

$$M(s) = \gamma \frac{s}{(s^2+\kappa^2)^{3/2}}.$$

We look for *periodic* solutions of (2) with period $1/f$, that is,

$$x(0) = x(1/f), \quad v(0) = v(1/f). \tag{3}$$

Find all periodic solutions of (2) with period $1/f$ for forcing frequencies f in the range between 2 (Hz) and 5 (Hz) by the following procedure.

(a) Discretise the boundary-value problem using the trapezoidal or the midpoint rule with $N = 20$ and $N = 60$ grid points and find the solutions of the resulting large nonlinear system $F(X, f) = 0$ (where the dimension of X is $2 \times N$) with continuation.

Plot a diagram of the resonance peak in the plane for $N = 60$: x-coordinate is the frequency f, y-coordinate is $\max\{|x_1(t)|\}$. (See example `Duffing` on Victory.)

[25 marks]

(b) The remaining points are awarded if the structure of the Jacobian of F is exploited: the number of function calls to `MagBeamRhs` per Newton iteration should be proportional to N (not N^2). The function F and its Jacobian must be set up in a way that they work for the example problems `bratu`, `Duffing` and `BucklingBeam` (see Victory) without change. *Help on how to do this will be given in the lecture.* [25 marks]

Total for Question 2: 50 marks

Hints

Jacobian For the continuation you have to set up two functions:

- the right-hand-side F of the nonlinear problem corresponding to the trapezoidal rule for (2) with periodic boundary conditions,
- the Jacobian of this right-hand-side (let's call it `dF`).

Initially, you may use just `dF=@(x)MyJacobian(F,x,h);` for the Jacobian. However, this is extremely inefficient. A working solution using based on this **earns only 25 points**.

Downloads You can download the file `MagBeamRhs.m` from Victory folder `Coursework 2`. This is a function file that implements the right-hand-side of (2). Also on Victory are the file `cw22.m`, which you may use as the start of your script file to hand in, and a file `cw2getpar.mat`, which loads your personal parameters (you may alternatively look up your parameters from the table in folder `Coursework 2`).