PX390 Assignment 4: a chemical reaction in a circular flow reactor

November 2021

1 The description of the problem

An equilibrium chemical reaction characterised by the equation A $\underset{k_-}{\overset{k_+}{\rightleftharpoons}}$ B, where k_+ is the rate constant of the A $\longrightarrow$ B reaction, while k_- is the rate constant of the reverse reaction B $\longrightarrow$ A. In a homogeneous reaction mixture, the concentrations A and B follow the differential equations

$$\frac{\mathrm{d}A}{\mathrm{d}t} = -k_+A + k_-B \quad \text{and} \tag{1}$$

$$\frac{\mathrm{d}B}{\mathrm{d}t} = k_+A - k_-B. \tag{2}$$

This reaction is carried out in a circular reactor as shown in Figure 1, with a circumference of L where a pump is circulating a reaction medium at a constant speed v. It can be assumed that the reactor is thin. Reactant A is added to the vessel at a $S(x)$ source rate, while product B is removed at a rate that depends on the local concentration $B(x)$ and a spatially varying coefficient $\sigma(x)$. Both compounds undergo diffusion with the same diffusion coefficient D. To summarise, the differential equations describing the variations of concentrations are

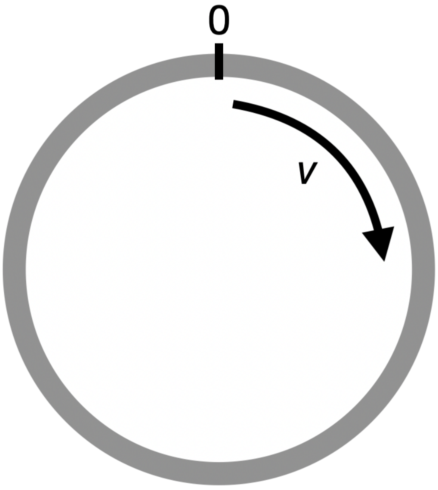

Figure 1: Schematic of the reactor.

$$\frac{\partial A}{\partial t} = D\frac{\partial^2 A}{\partial x^2} - v\frac{\partial A}{\partial x} - k_+A + k_-B + S \tag{3}$$

$$\frac{\partial B}{\partial t} = D\frac{\partial^2 B}{\partial x^2} - v\frac{\partial B}{\partial x} + k_+A - k_-B - \sigma B. \tag{4}$$

Your task is to model the steady-state solutions $A(x)$ and $B(x)$ after a sufficiently long time, where

$$\frac{\partial A}{\partial t} = \frac{\partial B}{\partial t} = 0 \quad \text{resulting in} \tag{5}$$

$$
\begin{aligned}
0 &= D\frac{\partial^2 A}{\partial x^2} - v\frac{\partial A}{\partial x} - k_+A + k_-B + S \\
0 &= D\frac{\partial^2 B}{\partial x^2} - v\frac{\partial B}{\partial x} + k_+A - k_-B - \sigma B.
\end{aligned}
\tag{6}
$$

The length of the spatial domain is L and periodic boundary conditions should be used, i.e. $A(L) = A(0)$, $B(L) = B(0)$ etc., as well as assuming $k_+ > 0$, $k_- > 0$, $v > 0$, $D > 0$, $S(x) \geq 0$ and $\sigma(x) \geq 0$ for all x.

You are required to use the `LAPACK` library to invert matrices, so the compiler flags are different this time. This library is installed on the lab computers and on `nenneke`. We will not provide any support for installing these libraries on your own computers, it might be quicker just to log in to `nenneke`. In order to compile your program (here, named `prog4.c`) on `nenneke` you need to execute the following commands:

1. `module purge`
2. `module load intel impi imkl`
3. `gcc prog4.c -lmkl -liomp5 -lm`

You also need to include

```
#include <mkl_lapacke.h>
```

in your source code. As before, you should test your code with simple test cases. We will, again, be marking it mostly based on how well it reproduces certain tests.

2 Some suggestions

- You do not need to prove that a stable steady state solution exist, you can assume that for sensible input parameters such a solution exist.
- The scheme used to numerically solve the differential equations must be convergent and stable. You can assume that first order methods will satisfy these criteria. To evaluate the spatial derivatives in equation 6 we recommend the central method for the diffusion term, and the upwinding method for the advection term.
- You will be using a grid to represent x and the functions. For N grid points, we recommend to set up the x grid such that $0, \frac{1}{N}L, \frac{2}{N}L, \ldots, \frac{N-1}{N}L$.
- The easiest way to call `LAPACK` is to use the matrix example codes on the PX390 website as a 'skeleton': this provides a framework for setting the values of the matrix at particular locations and solving the inverse problem.
- For reasons of numerical performance, make sure your matrix is a narrow banded matrix.
- One way to check that your answer is right is to write a time-evolution code.
- There are analytical solutions to these equations for simple cases.

3 Specifications

The input and output x grid has N ($N \geq 3$) grid points, starting at 0, but not including the "right" end of the domain; this grid will be taken to run from $x_0 = 0$ to $x_{N-1} = \frac{N-1}{N}L$. The grid points you use in setting up the matrix problem are up to you.

3.1 Input

The scalar input parameters will be read from a file called `input.txt`. The content of the file should be

1. L: right x boundary of domain,
2. N: number of grid points,

3. D: the diffusion coefficient of A and B,
4. v: advection velocity,
5. k_+: the forward rate constant of the reaction,
6. k_-: the backward rate constant of the reaction.

We will leave you to determine whether these should be read as integers or floating point numbers based on their meaning (note that the example input file does not necessarily have decimal points on all the values that should be floating point). It is sufficient to use `scanf` to read in these parameters and simply test that the reads were successful: we are not specifically forbidding the use of other ways of reading input but we found roughly 50% of students attempting a more complicated input scheme got it wrong. The functions $S(x)$ and $\sigma(x)$ (in that order) will be given at the N grid points as two columns of double precision numbers in a file called `coefficients.txt`. There is an example input parameters file and coefficients file on the assignment page.

3.2 Output

The code should write the solution to a file `output.txt`, in the following order: $x, A(x), B(x)$ in three column format for the grid points $x_0, x_1, \ldots, x_{N-1}$.