Modelling and Programming — IT & Health

Week 5

Deliverables

- Report 1 (in final form). *Deadline*: 23:59 Tuesday 3 November 2009.
- Report 2: A function for finding the part of a (t, C_p) data set that exhibits a linear relationship.
- Report 2: Section 1. Software for solving linear systems of equations.

You can hand in Report 1 digitally using DTU CampusNet or in hard-copy in class.

Functions in MATLAB

Writing small self-contained programs with well-defined input and output is an essential element of programming. Such programs are referred to as *functions*. Except for the language we use when we write functions in MATLAB, they are no different from functions in mathematics. The user provides an input and the function returns an output (sometimes there is no input, only output). To complete Report 2, you have to write a number of functions. The following exercises will help you learn about MATLAB functions:

- **P**: Exercises 1–2 and 4 of Section 10.4.

Make sure you understand the syntax for creating a new function (see **P**: Figure 10.2) and where to write the *help text* for a function (see **P**: Section 10.3.5). The *help text* is where you document your software in MATLAB.

The first function you need to write for Report Assignment 2 is one that finds the part of a (t, C_p) data set that exhibits a linear relationship. Let us call the function `find_linear_tail`. It takes two vectors as input. A vector `t` of times and a vector `ln_C` of logarithms of the concentrations measured at these times. The output of the function is an array of indices `idx` that point to the elements of the data set which exhibit a linear relationship. This array of indices is found using the algorithm (a–c) described in the report assignment.

To implement the algorithm (a–c) in the report assignment, you have to test what side of a straight line from $A = (\mathtt{t}_i, \mathtt{ln\_C}_i)$ to $B = (\mathtt{t}_{i_{\text{end}}}, \mathtt{ln\_C}_{i_{\text{end}}})$ a data point $C = (\mathrm{t}_j, \mathrm{ln\_C}_j)$ lies on. If you need help to do this, do the following exercises:

- Draw three points in a triangle on a piece of paper. From left to right name the points A, C, and B. Write down how to find the two-dimensional vectors $\vec{a}$ from A to B and $\vec{b}$ from A to C.
- Suppose θ is the angle between the vectors $\vec{a}$ and $\vec{b}$, explain why the sign of $\sin\theta$ reveals what side of the line AB the point C lies on. Standing at A and looking toward B, does a positive $\sin\theta$ say that the point C is on the left or on the right hand side?
- Use the following formula
$$\det\begin{bmatrix}\vec{a} & \vec{b}\end{bmatrix} = |\vec{a}||\vec{b}|\sin\theta$$
to write a MATLAB function called `left_right` which takes A, B, and C as input and returns a positive value if B is on the left hand side and a negative value if B is on the right hand side. (*Hint*: There is a MATLAB command called `det` which finds the determinant of a square matrix.)

The function `left_right` is useful for implementing `find_linear_tail`. Remember to provide *help text* for your functions.

Array conditionals and looping

Your m-file for the function find_linear_tail is probably still empty except, perhaps, for the first line which constructs the function and the help text which documents it. In continuation of the previous section, the following exercises will provide more help for implementing find_linear_tail:

- Write a for-loop in the find_linear_tail function which iterates over j from $i+1$ to $i_{\text{end}}-1$ updating C and storing the values returned by left_right(A,B,C) in a vector lr.

The results in lr are what you need to do the test in step c of the algorithm. As long as *all* the points between A and B are on the same side, the index from where the data exhibits a linear relationship has not been reached and we must increment i and start over. This is a loop that continues as long as a *condition* is true (see **P**: Example 8.0.1).

- Write a conditional statement that tests whether all the elements of lr are greater than 0. (*Hint*: Write doc all in the MATLAB command prompt.)
- Write the condition which tests whether all points between A and B are on the same side.
- Write a while-loop in the find_linear_tail function which uses this condition to find i such that the data from i to i_{end} exhibits a linear relationship. Make sure that your while-loop cannot turn into an infinite loop.
- Let the find_linear_tail function return the indices from i to i_{end} in idx.

After doing these exercises, you should have an implementation of the find_linear_tail function which is the first part of the Report 2 deliverables.

Gaussian elimination

Section 1 of Report Assignment 2 is about solving linear systems of equations using Gaussian elimination. Do the following exercises to implement Gaussian elimination:

- Write a MATLAB function called Gauss_elim which uses the MATLAB \ operator to solve a system of linear equations.
- Write a MATLAB script that tests the function Gauss_elim using Examples 2.4, 2.6, and 2.7 of **LA**. How does your function handle the problem described in Example 2.7 of **LA**?

Curriculum

The curriculum for Week 3 is

*	Book appendix. *Estimation of Absorption Kinetics from Plasma Concentration Data.*
LA	Sections 2.1–2.3. *Gaussian Elimination.*
P	Chapter 10. *Subprograms.*

Section 9.1 of **C** provides a soft introduction to the subject of linear systems of equations. So if you feel that you need a few extra words or a couple of extra examples about this subject, read Section 9.1 of **C** as well as Sections 2.1–2.3 of **LA**.

JRF 2009