

Report Assignment 2: Software for Pharmacokinetics

The second report assignment is about software development. As an example, we imagine that you have a pharmaceutical company as your customer. The pharmaceutical company develops drugs, and they measure a lot of data to find out how the drugs interact with the body after intake. The company is in possession of mathematical models that describe this interaction. The models involve a number of unknown parameters. Your task is to develop a piece of software that can help the company estimate parameters using the data they acquired.

When you deliver your software, it must be tested and well-documented. The test must ensure the correctness of your program. This means that apart from functions that estimate parameters your software must also include MATLAB scripts that test correctness. This is done by comparison to known results. All scripts and functions that are part of your software must be documented both in terms of technical and end-user documentation. This documentation is provided by writing appropriate comments in your code. These comments must be transformed into an HTML document using the MATLAB tool `m2html`.

Introduction to Pharmacokinetics

Pharmacodynamics is about what a drug does to the body. Pharmacokinetics, on the other hand, is about what the body does to the drug. More specifically, pharmacokinetics is about absorption, distribution, metabolism, and excretion of drugs in the body over time. The imaginary pharmaceutical company, which you are now working for, develops drugs that are taken as an oral dose (extravascular dose). Your job is to estimate the following five parameters that describe the absorption kinetics following an extravascular dose:

k – elimination rate

k_a – absorption rate

t_{lag} – lag time (the time between intake and start of absorption)

t_{max} – time of peak concentration.

C_{max} – peak concentration

These parameters are very important for the pharmaceutical company to determine the optimal oral dosing of their drugs. You need to estimate them given data that describe the plasma concentration C of the drug as a function of the time t after intake.

For this report you may assume that the absorption of the drugs is a first-order process in which one-compartment linear disposition kinetics apply. This means that you can use the *method of residuals* to estimate the parameters. The method of residuals is used as follows in absorption kinetics:

1. Do a semilog plot of the data (t, C) , where the C -axis uses the logarithmic scale.
2. Find the part of the data that exhibits a linear relationship between t and $\ln C$.
3. Do linear regression to find the equation of the straight line

$$\ln \overleftarrow{C} = -k t + c_1$$

that fits this part of the data.

4. For the remaining data points compute the residuals $\overleftarrow{C} - C$.
5. Do linear regression to find the equation of the straight line

$$\ln(\overleftarrow{C} - C) = -k_a t + c_2$$

that fits the residuals (exclude the two smallest residuals as they are too noisy).

6. Find the intersection point of the two lines. The time of the intersection is t_{lag} .
7. Find the maximum point $(t_{\text{max}}, C_{\text{max}})$ of the curve.

This method was previously carried out manually in the pharmaceutical company. Consequently, they have a lot of money to save from your automation of the method, but they cannot provide much help with respect to how it should be automated. It is your responsibility to design the software such that it implements each of the steps described above. They do, however, provide a rule of thumb which may help you automate Step 2. To find the part of the data that exhibits a linear relationship between t and $\ln C$, you can do the following:

- (a) Start with the first point (index i) and the last point (index i_{end}).
- (b) Find the straight line (or vector) that connects the two points.
- (c) Check if all data points in-between (index $i + 1$ to index $i_{\text{end}} - 1$) are on one side of the line.
 - If yes, the relationship is not linear. Go to the next point ($i = i + 1$) and start again.
 - If no, the relationship between the data points (from index i to index i_{end}) is linear.

Designing the Software

When designing a piece of software, it is important to consider what is input and what is output. This is something you would usually learn by consulting your customer. In this case, the input is a vector of time values and a corresponding vector of plasma concentrations. The value pairs at corresponding index positions of these two vectors are the (t, C) points. The output you need to provide is a semilog plot of these points, the residuals $(t, \overleftarrow{C} - C)$, and two lines inserted: a least squares fit to the linear part of the data and a least squares fit to the residuals. In addition, your software should output the five parameters that describe the absorption kinetics of the considered drug.

The report you hand in must include a description of your software design. The description should clearly illustrate how you intend to go from input to output. It is a good idea to draw a diagram and explain it. A box in the diagram could, for example, illustrate that a MATLAB function is called while an arrow could illustrate that data is passed from one function to another. Think about your design before you start implementing your software.

It is a great advantage in software development if you are able to identify self-contained subtasks that are useful in many different contexts. This facilitates code re-use and often reduces the workload. The following three sections suggest how you can use software developed at exercises as a part of your software package.

1 Linear Systems of Equations

Solving linear systems of equations is, among other things, useful for finding the intersection of two lines and for doing linear regression. This is needed for Steps 3, 5, and 6 in the method of residuals described above. Consequently, Gaussian elimination is very useful for solving parts of this report assignment. *Develop a documented and tested piece of software for solving linear systems of equations.*

2 Roots and Extrema

To find the two parameters t_{\max} and C_{\max} , you need a method for finding the first maximum along a curve. An extremum (maximum or minimum) is found where the derivative of a function is zero (where the root is). A maximum is where the derivative is positive before the zero-point of the derivative and negative after the zero-point. Since we do not have a functional expression that we can take the derivative of, we need to find a numerical approximation of the derivative.

To compute a numerical approximation of the derivative, we use the approximation

$$\frac{dC}{dt} \approx \frac{\Delta C}{\Delta t} = \frac{C_{i+1} - C_i}{t_{i+1} - t_i},$$

where i is an index into the vectors of data t and C . This provides approximate derivatives at the times: $t_i + \frac{1}{2}\Delta t_i$. Please note that the derivative vectors will have one data point less than the original data vectors. Find the index j for which the sign changes in the derivative vector from positive to negative, then you can use the line between the two points:

$$\left(t_j + \frac{1}{2}\Delta t_j, \frac{\Delta C_j}{\Delta t_j}\right) \quad \text{and} \quad \left(t_{j-1} + \frac{1}{2}\Delta t_{j-1}, \frac{\Delta C_{j-1}}{\Delta t_{j-1}}\right)$$

to find t_{\max} and afterwards you can use the line between (t_j, C_j) and (t_{j-1}, C_{j-1}) to find C_{\max} . *Develop a documented and tested piece of software for finding the first maximum along a curve.*

3 Linear Least Squares Problems

Linear regression, that is, how to fit a straight line to a set of data points using the least squares solution, is needed for Steps 3 and 5. It involves solving a linear system of equations. *Use your implementation of a solver for linear systems of equations to develop a documented and tested piece of software that does linear regression (linear least squares fitting).*

Testing the Software

Software testing is a profession in its own right (see, for example, the Wikipedia article¹ about it). The main purpose of testing is to detect software failures so that defects may be uncovered and corrected. In the context of this report, you are required to test the correctness of your software by using it to solve problems with known solutions. Correctness is then tested by comparing the results that your software returns to the known solutions. For example: use Examples 2.4, 2.6, and 2.7 of **LA** to test your solver for linear systems of equations; use the curve from Figure 4.5 in **LA** to test your program for finding the first maximum along a curve; and use Example 4.5 of **LA** and Problem 25 in Section 12.7.4 of **C** to test your solver for linear least squares problems.

In order for you to test the remainder of your software, the pharmaceutical company provides the example in Table 1 which includes a solution:

$$\begin{aligned} k &\approx 0.14 \\ k_a &\approx 0.53 \\ t_{\text{lag}} &\approx 0.5 \\ (t_{\max}, C_{\max}) &\approx (4.5, 0.97) \end{aligned}$$

¹http://en.wikipedia.org/wiki/Software_testing

Time t (hr)	Observation	Treatment of data	
	Plasma concentration C (mg/L)	Extrapolation \hat{C}_p (mg/L)	Residuals $\hat{C} - C$ (mg/L)
1	0.38	1.90	1.52
2	0.73	1.65	0.92
3	0.91	1.40	0.49
4	0.97	1.23	0.26
5	0.97	1.07	0.10
6	0.92	0.95	0.03
8	0.71	0.71	–
10	0.53	0.53	–
12	0.40	0.40	–
14	0.30	0.30	–

Table 1: Plasma concentration as a function of time as observed after oral administration of a 100 mg dose of a drug [Rowland and Tozer. *Clinical Pharmacokinetics: Concepts and Applications*, 3rd ed., Lippincott Williams & Wilkins, 1995].

However, the solution has been calculated by hand, so you will probably not be able to obtain the exact same results. Nevertheless, you can use it as a rough guide to check that your software provides plausible answers.

As a final test of your software, the pharmaceutical company would like you to submit the results that your software returns if you give the following observed data as input:

Time t (hr)	0.25	0.5	1	2	3	4	6	8	10	12
Plasma concentration C (mg/L)	1.6	2.7	3.7	3.5	2.7	2.0	1.02	0.49	0.26	0.12

Finalising Your Software Delivery

Since this report is slightly atypical, we summarise in this section what you are expected to hand in. Your report is expected to include the following:

1. Software design specification.
 2. Test results.
 3. Conclusion.
- A. HTML software documentation.

You are welcome to add other relevant details as long as your report does not exceed the page limit. The HTML software documentation is an appendix and is not to be included in the page count.