Technische Universität Braunschweig

Prof. Dr. Dirk Langemann & Jannis Marquardt, M.Sc.
Institute for Partial Differential Equations

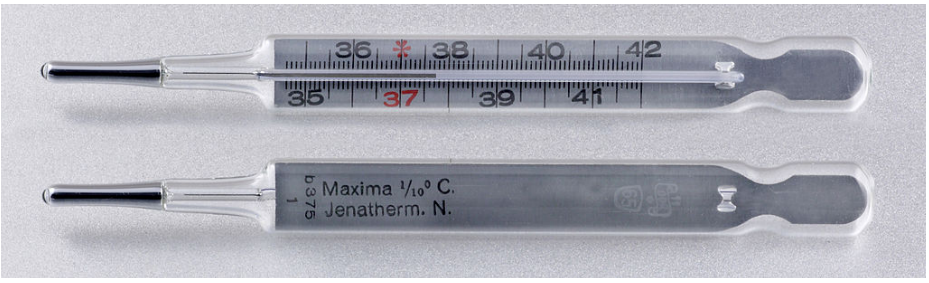
Assignment 2

- Programming exercises -

Upload your solution until Tue, 02 May 2023, 03:00 pm.

Programming exercise 2.1 *Regression* **(2 + 2 + 4 + 2 = 10 points)**

In this exercise, we will apply regression to a practical example. Imagine that you are a manufacturer of clinical thermometers. These thermometers use a certain liquid, which is contained in a small cylinder within the thermometer, to measure the temperature of i.e. a person. If the liquid gets warmer, it enlarges into the cylinder and contracts when it gets colder.



The height h of the liquid pillar within the cylinder is a linear (affine linear to be precise) function, depending on the temperature T:

$$h(T) = \alpha + \beta T \tag{1}$$

This function is specific for the liquid in the thermometer. Each liquid has its own specific choice of parameters α and β. In order to predict the height of the liquid in the cylinder (and use this knowledge to i.e. print a precise temperature scale on the outside of the thermometer), α and β have to be known for the liquid inside.

You, in your role as a manufacturer of such thermometers, decide to use a new liquid in the upcoming production line, for which you don't know the parameters α and β. Hence, these parameters have to be determined now. For that purpose, you heat the thermometer up to a certain temperature $\hat{T}$ and measure the height $\hat{h}$ of the liquid inside. Since you want to determine the parameters as best as possible, you do more that just two measurements (which would be enough to determine α and β through a linear system of two equations and two unknowns. In fact, you end up with a set of n temperature-height-pairs

$$\{(T_1, h_1), ..., (T_n, h_n)\}. \tag{2}$$

The first part of this assignment is to model this measuring process in Matlab:

(a) Create a new file `thermometer.m` and write a function

```
function h = thermometer(T)
```

which takes a temperature T as a parameter and returns the height h of the liquid in the cylinder. The actual relation of the temperature and the height is given as

$$h(T) = 2 + \frac{1}{5}T.$$

But since every measurement produces errors, the `thermometer`-function should also return an error afflicted height. This error afflicted height may be realised by considering the height function

$$\tilde{h}(T) = 2 + \frac{1}{5}T + \Theta(T),$$

where $\Theta : \mathbb{R} \mapsto (-0.75, 0.75)$ adds some noise (i.e. a random number) to each measurement result.
Hint: When you work with random numbers, each execution of your script will result in different values. This can be annoying while checking for errors in your code. You may use a 'seed' to generate the same random numbers for each execution.

Back to our imaginary plan of manufacturing thermometers:
After a lot of temperature-height-measurements, you ended up with a large number of data-pairs. In order to get an overview over the given data, you decide to visualise your measurements:

(b) Create a Matlab file `assignment_2.m` and plot the results of a height measurement with `thermometer(T)` for the temperatures

$$T \in \{t \in \mathbb{N}_0 \mid t \leq 80\}.$$

Plot each data point as a small dot and don't forget to label the axes of the plot.

After seeing this, you finally aim to find the parameters α and β from (1). The means to that purpose is the (linear) regression.

(c) Create a new file `regression.m` and write a function

```
function [alfa, beta] = regression(X),
```

where `X` is an array of datapoints as in (2) and `alfa` and `beta` are the parameters of an (affine) linear equation as given in (1).
As the name of the function suggests, you should use regression, applied to the dataset `X`, to determine `alfa` and `beta`. You are not allowed to use any build-in Matlab functions which would automatically compute these parameters.

(d) Call the regression function from the `assignment_02.m` file for the measurements which you have computed in (b). Plot the arising (affine) linear function (1) in a common plot with the datapoints.