# Statistics and Numerics

Lecture Prof. Dr. Jens Timmer

Exercises Helge Hass, Mirjam Fehling-Kaschek

# Exercise Sheet Nr. 5

### Exercise 1: Parameter estimation for a non-linear model

Consider the model

$$y(t) = A\left(1 - e^{-kt}\right) \tag{1}$$

with parameters $A$ and $k$ that are comprised as $p = (A,k)$.

a) Generate simulated data for normal distributed noise, i.e. $y_i^D = y(t_i) + \varepsilon \sim N(0,\sigma^2)$ with $\sigma = 0.1$.

- Implement a function `def MyModel(p,t)` : that returns function values for given parameter values and time points $t$.
- Implement a second function `def MyData(p,t)` : that returns data points by calling `MyModel` and adding noise.

Generate four data sets for $A = 5$, $k = 0.5$ and time points

- $t = \{0.5, 1\}$
- $t = \{17, 20\}$
- $t = \{0, 1, 3, 10, 15, 20\}$
- $t = \{1, 2, \ldots, 20\}$.

Plot the data sets along with the expected model trajectories $y(t)$ within $t = [0, \ldots 20]$.

b) Estimate parameters by the least squares method. The methods is based on finding parameters that minimize the weighted residual sum of squares:

$$\chi^2(A,k) = \sum_t \left(\frac{y^D(t) - y(A,k,t)}{\sigma}\right)^2, \tag{2}$$

where the residual is the distance between data points $y^D(t)$ and model prediction $y(A,k,t)$ and the weight is given by the uncertainty $\sigma$. In which case is the least squares method equivalent to the maximum likelihood method?

The minimization of $\chi^2(A,K)$ should be accomplished via a numerical optimization method. The python method *scipy.optimize.least_squares* is called via $least\_squares(res\_fun, x0)$ with $res\_fun$ corresponding to a function returning (weighted) residuals and $x0$ corresponding to initial values of the parameters to start the optimization.

1. Write the function `res_fun(p)` that returns a vector of residuals given a parameter vector $p$. This means, for every time point you return $\frac{y^D(t)-y(A,k,t)}{\sigma}$. Squaring and summation of the residuals is done internally by the optimizer.
2. Generate a random starting point, e.g. $A \in [0,10]$, $k \in [0,10]$ that is required by the optimizer.
3. Own arguments can be passed to the residual function, e.g. via $least\_squares(res\_fun, x0, args = (input\_1, input\_2))$ with corresponding $res\_fun(p, input\_1, input\_2)$. This can be used to pass $\sigma$, the data points and the time vector to get the model output (call MyModel(p,t) within $res\_fun$).

4. Cross-check your *res_fun* by calling it with different data points and parameters. Do the residuals make sense?

   - Estimate parameters for the different data sets. The output variable of the optimizer comprises the variable $x$ for the final parameter set and *cost* for the $\chi^2$ value.
   - Compare the parameter set used to generate the data sets with those found by the optimizer.
     Which parameter results in a better $\chi^2$ and trajectories that better fit the data?

**Exercise 2: Profile Likelihood**

Implement a method to compute the profile likelihood for the problem given in Exercise 1. Compute the parameter profiles for $p_i \in \{A, k\}$ following this procedure:

1. Generate an equidistant sample of values $x_j$, $j \in [1, 100]$, for the parameter of interest. The sample should be centered around the optimal parameter value found by the parameter estimation.
2. Reoptimize by fixing $p_i = x_j$, $j \in [1, 100]$. That means that only the remaining parameter is estimated by the least squares method while $p_i$ is passed to the residual function as its own argument (see Exercise 1b, point 3). Start the reoptimization of the remaining parameter at the optimal parameter value found in Exercise 1.
3. Plot the resulting profile likelihood ($\chi^2$ as a function of $x_j$).

Interpret the parameter profiles obtained for the four data sets.