PH40083: Advanced Problem Solving

Coursework 2 – Linear and Non-linear Least Squares

The coursework is focused on developing a deeper understanding on linear and non-linear least squares estimation and their uncertainties. The deadline is 4pm on Friday 1st of May. You need to upload your report with codes via Moodle.

- ***Report (in .PDF format): There is no official template nor required structure. Page limit is 5 pages (excluding Appendices). I expect that you code your own least squares solvers (e.g. MatLab, Python, C, ...) and that the results are discussed and presented in a clear manner with graphs that show the data and the fitting curves, and that in every exercise the estimated parameter values are presented with their estimated uncertainties (i.e. standard deviations extracted from the covariance matrix).***
- ***Codes: Add the codes that you have used as Appendices to the report (e.g. Appendix A: Code for Exercise 1, ...).***
- ***Marking: 60% of marks will be given based on sound, working programs and correct numerical results. Additional marks will be given for clear presentation of results, insightful analysis and discussion, and very good report writing style.***
- ***Many of you will be working remotely and using the programming tools available through UniDesk. I do understand that this may cause some problems, and I would appreciate that in such a case you would contact me as soon as possible.***
- ***How to work from home:*** ***https://www.bath.ac.uk/guides/preparing-yourself-to-work-from-home/***
- ***Using UniDesk and UniApps:*** ***https://www.bath.ac.uk/guides/work-remotely-with-uniapps-and-unidesk/***

1. Linear least squares (max 35 points) (data set: linear_LeastSquares.txt on Moodle)

In this part, you will study a linear model and linear least squares fitting. First, download the data set from Moodle. It corresponds to a measurement model of the following form

$$Y = aX + b + e,$$

where a and b are unknown parameters that we are looking for, and e models the random measurement noise with zero mean value and standard deviation $\sigma_{noise} = 5$. The file contains two columns of data. The first column has values for variable X and the second values for Y.

a) The data can be presented as an overdetermined linear system.

$$\underbrace{\begin{bmatrix} Y1 \\ Y2 \\ \vdots \\ YM \end{bmatrix}}_{Z} = \underbrace{\begin{bmatrix} X1 & 1 \\ X2 & 1 \\ \vdots & \vdots \\ XM & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\theta} + \underbrace{\begin{bmatrix} e1 \\ e2 \\ \vdots \\ eM \end{bmatrix}}_{e}$$

Write a computer program that calculates the linear least squares fit by explicitly using the linear least squares (LS) formulas given below

$$\hat{\theta}_{LS} = (H^T H)^{-1} H^T Z$$
$$\hat{\Gamma}_{LS} = \sigma_{noise}^2 (H^T H)^{-1}$$

What are the values of the estimated parameters, $\hat{\theta}_{LS}$? What are the values of the uncertainties based on the covariance matrix of the estimate, $\hat{\Gamma}_{LS}$?

b) Use a built-in fitting tool/toolbox (that is included in e.g. MatLab, Python,…) that can solve the same least squares problem, and compare the results. Explain what the fitting tool does with the data i.e. how does it work. Are there differences in the results? If there are, then explain what causes the differences?

c) Repeat part a) but perform the linear least squares fit with smaller numbers of data points. How do the solutions change?

d) Explain briefly what is least absolute residuals (or least absolute deviations) fitting. How does it differ from least squares fitting? Give an example case when it could be useful.

2. Weighted linear least squares (max 20 points) (data set: weighted_LS.txt)

Sometimes, we face a situation that not all the measured data points have the same accuracy. This can be considered in the data-analysis by using weighted linear least squares (WLS).

Download the data set from Moodle. It corresponds to a measurement model given below

$$Y = AX^2 + B\log(X) + C + e,$$

where A, B and C are the unknown constants that we want to estimate, e is the measurement noise, and log refers to the natural logarithm. The data set contains $M = 10$ values for X (first column) and Y (second column).

Here, we assume that e is additive white Gaussian noise (i.e. additive random noise with zero mean value) and that every measurement has a noise level that can be described by its variance. The weighted LS estimation can be performed with the help of these variances.

a) Calculate the linear least squares fit, first, by using the regular linear least squares (i.e. solve $\hat{\theta}_{LS}$ and $\hat{\Gamma}_{LS}$ using the formulas above) and assuming $\sigma_{noise} = 1$.

b) In this data set, however, the first 5 measurements are more accurate (with $\sigma_{noise} = 0.1$) than the last 5 (with $\sigma_{noise} = 10$). Use this knowledge to calculate the weighted least squares fit, $\hat{\theta}_{WLS}$ and $\hat{\Gamma}_{WLS}$. Compare the result with the one from part a).

3. Non-linear least squares (max 45 points) (data set: nonlinear_LS.txt)

Download the data set from Moodle. The first column in the data set has the values of variable X and the second column values of Y. Consider the following non-linear model

$$Y = X\exp(A) + B\exp(X) + e,$$

where A and B are the unknown constants to be estimated. We can present this in a general form as follows

$$Z = H(\theta) + e,$$

where $Z = [Y_1, Y_2, \ldots, Y_M]^T$, $M = 10$ is the number of measurements, $H(\theta)$ represents the non-linear system, θ is the vector of unknowns, and e is the measurement noise. The system can be solved by using the iterative Gauss-Newton method. The ith iteration takes the form

$$\hat{\theta}_{i+1} = \hat{\theta}_i + \kappa(J_i^T J_i)^{-1} J_i^T (Z - H(\theta_i)),$$

$$\hat{\Gamma}_i = \sigma_{noise}^2 (J_i^T J_i)^{-1},$$

where $\kappa > 0$ is a step parameter and J is the Jacobian matrix of the system.

a) What are the partial derivatives in the Jacobian matrix J ?

b) Write a computer program that performs the non-linear least squares estimation by explicitly evaluating the Gauss-Newton formulas. Try using an initial guess for the solution vector $\theta_0 = [A_0, B_0]^T = [\,3, 1\,]^T$, step parameter $\kappa = 0.01$, and assume $\sigma_{noise} = 1000$. How many iterations is needed to reach the solution? Discuss how could you choose the stopping criteria for the iterations?

c) Would it have been possible to use linear least squares to solve the problem? If "yes", what is the result. If "no", explain why.

d) Study the stability of the Gauss-Newton algorithm. For example, does the solution depend on the initial guess?

e) Many optimisation toolboxes include the Levenberg-Marquardt algorithm. Explain briefly how it works, and how it differs from the Gauss-Newton algorithm.