02610 Optimization and Data Fitting — Fall 2014

Assignment 1

List all your Matlab code in an appendix!

The report must be handed in no later than monday October 27, 2014 at 13:00 to secretary Mette E. Larsen, office 105, building 303B. You must also upload a version to Campusnet.

Question 1. Fitting to Air Pollution Data

t_i	y_i	t_i	y_i	t_i	y_i	t_i	y_i	t_i	y_i
		5	42.96	10	299.23	15	238.75	20	206.69
1	91.22	6	101.05	11	238.00	16	271.26	21	170.77
2	28.04	7	202.36	12	227.49	17	267.72	22	131.67
3	22.91	8	328.02	13	218.03	18	251.32	23	143.85
4	26.65	9	364.12	14	223.62	19	230.04	24	157.57

Measurements of NO concentration y_i as a function of time t_i. The units of y_i and t_i are μg/m^3 and hours, respectively.

We are given measurements of the air pollution, in the form of the concentration of NO, over a period of 25 hours on H. C. Andersen's Boulevard in central Copenhagen. The data is given in the table above. For further analysis of the air pollution we need to fit a smooth curve to the measurements, so that we can compute the concentration at an arbitrary time between 0 and 24 hours. Since the data repeats every day we will use periodic functions, and the fitting model takes the form:

$$M(\boldsymbol{x}, t) = x_1 + x_2 \sin(\omega\, t) + x_3 \cos(\omega\, t) + x_4 \sin(2\omega\, t) + x_5 \cos(2\omega\, t) + \cdots \qquad (1)$$

where $\omega = 2\pi/24$ is the period. Here, the vector $\boldsymbol{x} = (x_1,\, x_2,\, \ldots,\, x_n)^T$ consists of the unknown coefficients. The goal of data fitting in relation to this problem is to estimate the coefficients $x_1, x_2, \ldots, x_n$ such that we can evaluate the function $f(t)$ for any argument t. At the same time we want to suppress the influence of errors present in the data.

1.1: Matlab Code. Write a Matlab function with the call

```
[x_star,r_star] = NOfit(t,y,n)
```

that computes the least squares fit with the model (1) to the data points in the vectors $\boldsymbol{t}$ and $\boldsymbol{y}$. The order of the fit is n, which must be an odd number, and the underlying basis functions are the trigonometric functions:

$$f_1(t) = 1, \quad f_2(t) = \sin(\omega\, t), \quad f_3(t) = \cos(\omega\, t),$$

$$f_4(t) = \sin(2\omega\, t), \quad f_5(t) = \cos(2\omega\, t), \quad \ldots$$

The function returns the vector $\boldsymbol{x}^* \in \mathbb{R}^n$ with the coefficients to the least squares fit, and the corresponding least squares residual vector $\boldsymbol{r}^* \in \mathbb{R}^m$.

The Matlab code must include a reasonable amount of comments, as well as a sufficient description of the input- and output-parameters, and it must be listed in your report.

1.2: Testing the Software. Test your Matlab function on the data in the table and with $n = 3$ (this is NOT a good fit). List the found solution $\boldsymbol{x}^*$ and plot the least squares fit $M(\boldsymbol{x}^*, t)$ together with the data points. The plot must be shown in the report. The norm of the corresponding residual vector should be $\|\boldsymbol{r}^*\|_2 = 292.558$.

1.3: The Optimal Order of the Fit. Now we must determine the optimal order n of the fit, such that the fit follows the overall behavior of the data without following the random variations due to the noise. Implement these two tests covered in the lecture notes:

- test for random signs,
- test for correlation.

In your report you must document your investigation and describe how you reached your conclusion about the optimal n. Do you agree with the value of n determined by the above tests?

1.4: Estimating the Standard Deviation of the Solution Coefficients. The standard deviation for the ith solution coefficient x_i is the square root of the ith diagonal element of the covariance matrix for the solution, given by

$$\mathrm{Cov}(\boldsymbol{x}^*) = \varsigma^2 (A^T A)^{-1},$$

where ς is the standard deviation of the noise. For the optimal fit computed previously, you must compute the standard deviations for the elements of $\boldsymbol{x}^*$.
To estimate ς you should use the technique from the notes, which involves the scaled residual norm (or standard error) given by

$$s^* = \frac{\|\boldsymbol{r}^*\|_2}{\sqrt{m-n}}.$$

Describe in details how you determine ς (including a figure or table, if necessary), and list the standard deviations for the elements of $\boldsymbol{x}^*$.

Question 2. Algorithms for Nonlinear Unconstrained Optimization

In this question we consider the nonlinear unconstrained optimization problem

$$\min_{x \in \mathbb{R}^2} f(x) \tag{2}$$

with

$$f(x) = f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{3}$$

2.1: Contour plot and stationary points
Make a contour plot of $f(x)$ and locate all local minimizers, all local maximizer, and all all saddle points.

2.2: Gradient and Hessian.
Derive the gradient and the Hessian for $f(x)$. Verify that $\nabla f(x) = 0$ for all stationary points in 2.1. What property does the Hessian have for a) the local minimizers, b) the local maximizers, and c) the saddle points. Verify this numerically.

2.3 Steepest descent algorithm
Implement the steepest descent algorithm with a line search algorithm of your choice. You should motivate your algorithmic choices and explain the algorithm. Test this algorithm for various starting points. Use the contour plot to demonstrate the convergence sequence for this algorithm. Make tables and plots of the convergence rate. Is the convergence rate as expected.

2.4 Newton's algorithm
Implement Newton's algorithm with a line search algorithm of your choice. You should motivate your algorithmic choices and explain the algorithm. Test this algorithm for various starting points. Use the contour plot to demonstrate the convergence sequence for this algorithm. Make tables and plots of the convergence rate. Is the convergence rate as expected.

2.5 Quasi-Newton algorithm
Implement a BFGS-based Quasi-Newton algorithm with a line search algorithm of your choice. You should motivate your algorithmic choices and explain the algorithm. Test this algorithm for various starting points. Use the contour plot to demonstrate the convergence sequence for this algorithm. Make tables and plots of the convergence rate. Is the convergence rate as expected.

2.6 Gauss-Newton algorithm
Explain why we can consider $\min_x f(x)$ to be a least squares problem. Implement a Gauss-Newton algorithm with a line search algorithm of your choice. You should motivate your algorithmic choices and explain the algorithm. Test this algorithm for various starting points. Use the contour plot to demonstrate the convergence sequence for this algorithm. Make tables and plots of the convergence rate. Is the convergence rate as expected.

2.7 Levenberg-Marquardt algorithm
Implement the Levenberg-Marquardt algorithm. You should motivate your algorithmic choices and explain the algorithm. Test this algorithm for various starting points. Use the contour plot to demonstrate the convergence sequence for this algorithm. Make tables and plots of the convergence rate. Is the convergence rate as expected.

2.8 Comparison of algorithms
Test and compare the algorithms you have developed in this question. You should also compare your algorithms to relevant algorithms from Matlabs Optimization toolbox and IMM Optibox. Make tables and figures demonstrating convergence rates and tables that indicate the number of iterations, function evaluations, Jacobi evaluations, CPU time, etc for solution of the problem from various starting points. Discuss the pros and cons of the various algorithms.