Data Smoothing

Exercise 3

Data Smoothing Report

Author
Cesare De Cal

Professor
Annie Cuyt

Assistant Professor
Ferre Knaepkens

November 6, 2019

1 Introduction

This exercise asks to use the linearly independent basis functions:

$$\Phi_{3,i}(x) =$$

to find the optimal combination

$$\Phi(x) = \lambda_0(x)$$

that minimizes
for the 20 data points (x_j, y_j) given in

j	x_j	y_j
0	0.0	-0.80
1	0.6	-0.34
2	1.5	0.59
3	1.7	0.59
4	1.9	0.23
5	2.1	0.10
6	2.3	0.28
7	2.6	1.03
8	2.8	1.50
9	3.0	1.44
10	3.6	0.74
11	4.7	-0.82
12	5.2	-1.27
13	5.7	-0.92
14	5.8	-0.92
15	6.0	-1.04
16	6.4	-0.79
17	6.9	-0.06
18	7.6	1.00
19	8.0	0.00

2 Tools

The following programming language and libraries have been used in this exercise:

- Item 1
- C Math Library
- GSL (GNU Scientific Library)

The following double-precision GSL data types have been used in the exercise:

- `gsl_vector ?`

The following GSL methods have been used in the exercise:

- `gsl_matrix_alloc(size1, size2)`
- `gsl_matrix_set_zero(matrix)`
- `gsl_matrix_set(matrix, row, column, value)`
- `gsl_matrix_get(matrix, row, column)`
- `gsl_vector_alloc(size)`
- `gsl_vector_set_zero(vector)`
- `gsl_vector_set(vector, index, value)`
- `gsl_vector_get(vector, index)`
- `gsl_matrix_memcpy(matrixToCopyFrom, matrix)`
- `gsl_linalg_SV_decomp(A, V, S, workspaceVector)`
- `gsl_vector_minmax(vector, minInVector, maxInVector)`

In order to factorize a matrix into the LU decomposition, and then solve the square system $Ax = y$ using the decomposition of A, I've used the following methods:

- `gsl_linalg_LU_decomp(A, permutation, signum)`
- `gsl_linalg_LU_solve(LU, permutation, b, x)`
- `gsl_permutation_alloc(size)`

The following method from the C Math library was used in this exercise to calculate the absolute value of a number:

- `fabs(x)`

3 Computation

First off, I compute the coefficients A of the linear system by using the linearly independent basis function. This is what A looks like:

$$\begin{bmatrix}
1.000000000000000e+00 & 0.000000000000000e+00 & 0.000000000000000e+00 & 0.000000000000000e+00 \\
6.400000000000002e-02 & 2.880000000000000e-01 & 4.320000000000001e-01 & 2.160000000000000e-01 \\
-1.250000000000000e-01 & 1.125000000000000e+00 & -3.375000000000000e+00 & 3.375000000000000e+00 \\
-3.429999999999999e-01 & 2.499000000000000e+00 & -6.068999999999998e+00 & 4.912999999999999e+00 \\
-7.289999999999998e-01 & 4.616999999999998e+00 & -9.747000000000000e+00 & 6.858999999999999e+00 \\
-1.331000000000000e+00 & 7.623000000000002e+00 & -1.455300000000000e+01 & 9.261000000000001e+00 \\
-2.196999999999999e+00 & 1.166100000000000e+01 & -2.063099999999999e+01 & 1.216700000000000e+01 \\
-4.096000000000001e+00 & 1.996800000000001e+01 & -3.244800000000000e+01 & 1.757600000000000e+01 \\
-5.831999999999998e+00 & 2.721599999999999e+01 & -4.233599999999999e+01 & 2.195199999999999e+01 \\
-8.000000000000000e+00 & 3.600000000000000e+01 & -5.400000000000000e+01 & 2.700000000000000e+01 \\
-1.757600000000000e+01 & 7.300800000000001e+01 & -1.010880000000000e+02 & 4.665600000000001e+01 \\
-5.065300000000001e+01 & 1.930290000000000e+02 & -2.451990000000000e+02 & 1.038230000000000e+02 \\
-7.408800000000001e+01 & 2.751840000000000e+02 & -3.407040000000000e+02 & 1.406080000000000e+02 \\
-1.038230000000000e+02 & 3.777390000000001e+02 & -4.581090000000000e+02 & 1.851930000000000e+02 \\
-1.105920000000000e+02 & 4.008960000000000e+02 & -4.844160000000000e+02 & 1.951120000000000e+02 \\
-1.250000000000000e+02 & 4.500000000000000e+02 & -5.400000000000000e+02 & 2.160000000000000e+02 \\
-1.574640000000000e+02 & 5.598720000000002e+02 & -6.635520000000001e+02 & 2.621440000000001e+02 \\
-2.053790000000000e+02 & 7.205670000000001e+02 & -8.426970000000001e+02 & 3.285090000000001e+02 \\
-2.874960000000000e+02 & 9.931679999999998e+02 & -1.143648000000000e+03 & 4.389759999999999e+02 \\
-3.430000000000000e+02 & 1.176000000000000e+03 & -1.344000000000000e+03 & 5.120000000000000e+02
\end{bmatrix}$$

4 Plot

5 Observations