Lab #5. Inversion with linear constraints

In this lab, you will practice inversion with explicit (exact) constraints and Laplacian smoothness constraints.

Step 1: Add switches to your code or make new versions to add linear constraints using two methods discussed in the course notes: exact model constraints (section 7.1 in the notes) and Laplacian smoothness constraints (section 7.2; eqs. 7.13 and 7.14).

Using these programs, invert the following integral for a function $m(x)$ sampled on interval $x \in [0,1]$:

$$\int_0^1 x e^{-yx} m(x) dx = d(y) . \quad (1)$$

In the table at the end of this description (also in file asg5.dat), values of this integral are given for 25 values of $y \in [0.1,10]$. Random Gaussian error with mean zero and standard deviation 0.000005 was included in $d(y)$.

In the program, denote the number of data points (N_d in my text) by N = 25 and the number of model parameters (N_m in the text) by M. Use an identifier for M, so that it can be changed in the following tests. Define an equally spaced grid of M model-grid points (using `linspace()`), so that the first of them is at x = 0 and the last x = 1.

Step 2: Using this grid of $m(x_i)$, construct kernel matrix **L** that would express equation (1) as a linear forward model **Lm** = **d**. For the quadrature (approximation of the integral (1) by a sum), use the trapezoidal or Simpson's integration rule.

Step 3: Also define two constraint matrices:

1) Two-row matrix **B** such that equation **Bm** = **0** implements two constraints on the resulting model function: $m(0)$ = 0, $m(1)$ = 0. If your model is a vector (matrix column in Matlab) like this:

$$\mathbf{m} = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{N_m-1} \\ m_{N_m} \end{pmatrix}, \tag{2}$$

then the constraint matrix will be

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}. \tag{3}$$

2) Roughness matrix **R** (see section 7.2) such that expression $\mathbf{m}^{\mathrm{T}}\mathbf{R}\mathbf{m}$ approximates the integral of minus the Laplacian of the model function: $-\int_0^1 \frac{d^2}{dx^2} m(x)dx$. The integral of the Laplacian will be negative, and therefore the roughness term $\mathbf{m}^{\mathrm{T}}\mathbf{R}\mathbf{m}$ will be positive for functions $m(x)$ oscillating within the interval. Simply use eq. (3.35) in class notes for the Laplacian.

Step 4: Using matrices **L**, **B**, and **R**, and several forms of gridding, solve for $m(x)$ for the following constraints on the model:

1) Take a small $M < N$ and perform the least-squares inversion;
2) Try the simple even-determined quadrature, i.,e. $M = N = 25$, and solution of the form $\mathbf{m} = \mathbf{L}^{-1}\mathbf{d}$. I think you will find that the solution will fail; is this so?

The solution should fail because there exist functions $m(x)$ for which $\int_0^1 xe^{-yx}m(x)dx = 0$ (null space of the forward kernel). These functions can be removed by additionally constraining the solution:

3) For $M = N = 25$, try the damped least squares (Marquardt-Levenberg) inversion. This is a “soft” (approximate) constraint requiring that the model $m(x) \approx 0$ not exactly but “on average”.
4) Try the smoothest-model inversion using matrix **R** (section 7.2 in the notes).

5) Try exact constrained inversion using matrix **B**. The method is described in section 7.1 in class notes. Verify that the constraints $m(0) = 0$ and $m(1) = 0$ are achieved exactly.

Data table:

y	$d(y)$
0.1	-0.1439
0.2	-0.1299
0.3	-0.1170
0.4	-0.1052
0.5	-0.0944
0.6	-0.0844
0.7	-0.0753
0.8	-0.0699
0.9	-0.0593
1.0	-0.0523
1.2	-0.0400
1.4	-0.0297
1.6	-0.0211
1.8	-0.0140
2.0	-0.0081
2.5	+0.0025
3.0	+0.0088
3.5	+0.0123
4.00	+0.0140
5.0	+0.0144
6.0	+0.0130
7.0	+0.0112
8.0	+0.0099
9.0	+0.0078
10.0	+0.0065