Applications of matrix computations, Spring 2018   Exercise 3

Lecturer: Samuli Siltanen.
Teaching assistants: Vesa Kaarnioja and Jesse Railo.

Deadline: Monday 16.04. at 10.00. *Sending your solutions late is not allowed!*
     The exercises are held in the room C321 of Exactum on Wed 14.15–16.00 (group 1), Thu 12.15–14.00 (group 2), Fri 10.15–12.00 (group 3), and Fri 14.15–16.00 (group 4). You gain credits from this course only by completing the weekly exercises. Participating in the exercise group is not obligatory for completing the course, but it is highly recommended.

     ATTENTION! Return your solutions as a single PDF file containing your Matlab codes (as text), results, images, comments, and explanations at the **Moodle page of the course** before the deadline (further instructions are given at the exercise groups if needed). Please include your name, student number, and your exercise group number in the beginning of the PDF file.

1. Define three sets of vectors as follows:

$$
\left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1000 \\ -1 \end{bmatrix} \right\}, \qquad
\left\{ \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \\ -3 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\}, \qquad
\left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1000 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right\}.
$$

   (a) Use Matlab to determine which of the above sets are linearly independent.

   (b) Look up from the literature what is Gram-Schmidt orthogonalization. Explain it briefly (at most one page of text, preferably less).

   (c) Use Matlab and Gram-Schmidt orthogonalization to produce orthonormal bases out of the sets of vectors above that you found to be linearly independent.

2. Download the file `DataFitExample1.m` from the course website. You will need to modify the file in the exercises below. Choose $a_0 = 0.8$ and $b_0 = 10$ in the ideal model $y = a_0 x + b_0$.

   (a) Take the simulated measurement noise amplitude to be `noiselevel= 1;` and take the x points to be `10:10:100;` leading to 10 data points. Compute the mean of ten least-squares estimated parameter values of a and b. How close is the mean to the actual values? Repeat the above but take the mean over 100 recovered parameter values of a and b. Is the mean now closer? (Hint: use `for`-loops.)

   (b) Take the simulated measurement noise amplitude be `noiselevel= 5;` and take the x points to be `10:1:100;` leading to 91 data points. Compute the mean of ten recovered parameter values of a and b. How close is the mean to the actual values? Repeat the above but take the mean over 100 recovered parameter values of a and b. Is the mean now closer? How do the results compare to those from (a)?

3. The following picture contains the temperature of water in an electric kettle as a function of time, data measured at the lecture on 28.3.2018:
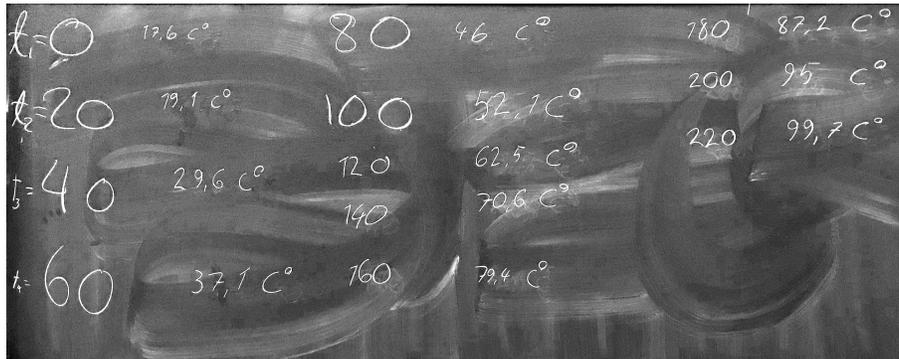

(a) Create data vectors `timevec` and `tempvec` based on the measurement so that the time instants are in increasing order and the temperatures still correspond to the correct measurement times.

(b) Fit a linear model to the data in the least-squares sense. Draw a figure that illustrates the data points and the linear model using the commands `plot` and `hold on`.

(c) Using known constants of physics and the fact that the amount of water in the experiment was 1 liter, estimate the power of the electric kettle in watts.

4. Download the photo `SeaView.jpg` from the course website. Read it in Matlab using the command `im = imread('SeaView.jpg','jpg');`. Furthermore, convert the pixel values to floating point numbers by writing `im = double(im);` Then you will have the following image in your workspace as a matrix:


(a) Plot the image using the command `imagesc(im)`. Next improve the plot by using these commands: `colormap gray` and `axis equal`. Can you explain the meaning of the two commands?

(b) Make the image brighter by applying `gam` with a suitable value between zero and one. You can choose the best value according to your own taste and follow the instructions in this blog post:

blogs.helsinki.fi/smsiltan/2014/05/26/how-to-make-blackwhite-images-look-good/.

5. In the pictures of two-dimensional square "patients" below, thin lines depict pixels and thick lines X-rays. The length of the side of a pixel is always one.

(a) Give a numbering to the nine pixels ($x \in \mathbb{R}^9$) and to the six X-rays ($m \in \mathbb{R}^6$) shown below. Construct the matrix A_1 for the tomographic measurement model $m = A_1x$.
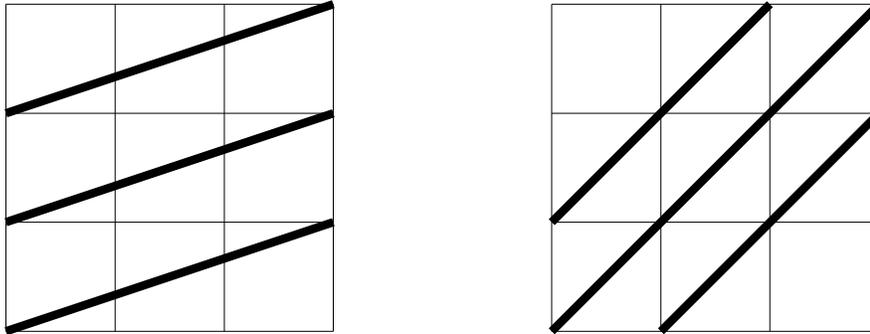
(b) Using the same numbering of pixels than in (a), construct the matrix A_2 for the measurement model $\widetilde{m} = A_2x$ corresponding to the six X-rays shown above *and* to the six X-rays shown below. Now $\widetilde{m} \in \mathbb{R}^{12}$.
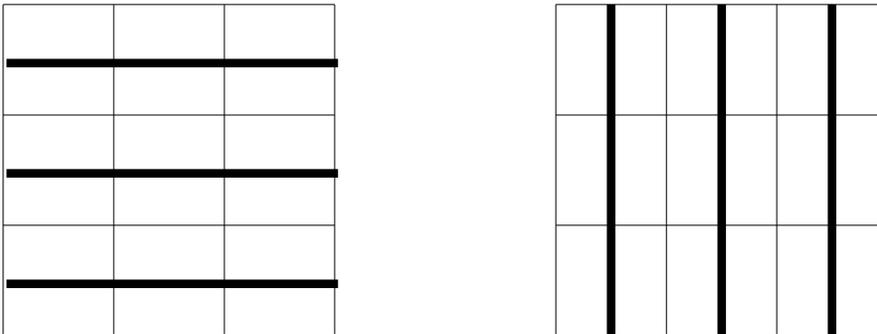
(c) Give a non-negative value to each component of x. That is, create a "patient" of your own choosing. Then simulate two data vectors by computing $m = A_1x$ and $\widetilde{m} = A_2x$ for the *same* vector x. Then calculate least-squares reconstructions $A_1 \backslash m$ and $A_2 \backslash \widetilde{m}$ using the Matlab operator "\" (called *könökeppi* in Kuopio). Are the reconstructions different?

(d) Show the original x and the reconstructions from (c) side by side in the same figure as 3×3 pixel images using the `imagesc` command.

6. Consider the systems

$$Ax = b$$

of linear equations given below. Here $x = [x_1 \ x_2]^T$. Use Matlab to plot (in black color) the lines in the (x_1, x_2) plane related to the equations. Also, plot the solution (or least-squares solution) as a red dot. Hint: try the commands `x = inv(A)*b` or `x = A\b` or `x = pinv(A)*b`.

(a) Take $A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

(b) Take $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Here a true solution does not exist (why?), so use least-squares solution.

(c) Take $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ -1 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}$.

Here a true solution does not exist (why?), so use least-squares solution.

(d) Define A and b as in (c) . Download the Matlab files `Qfunctional.m` and `Qminimum_v2.m` from the course website. Modify the file `Qminimum_v2.m` so that you can observe the path $\tilde{x} + tw$ traveling through the location $\tilde{x}$ of the minimum of Q.