Applications of matrix computations Exercise 6, October 17–21, 2016

Lecturer: Samuli Siltanen.
Teaching assistants: Santeri Kaupinmäki and Jonatan Lehtonen.

Please send your solutions to application.matrixcomputation@gmail.com by Monday, October 24, at 10 AM.

1. Load the acceleration signal contained in the file *lecture_acc.mat* as follows:

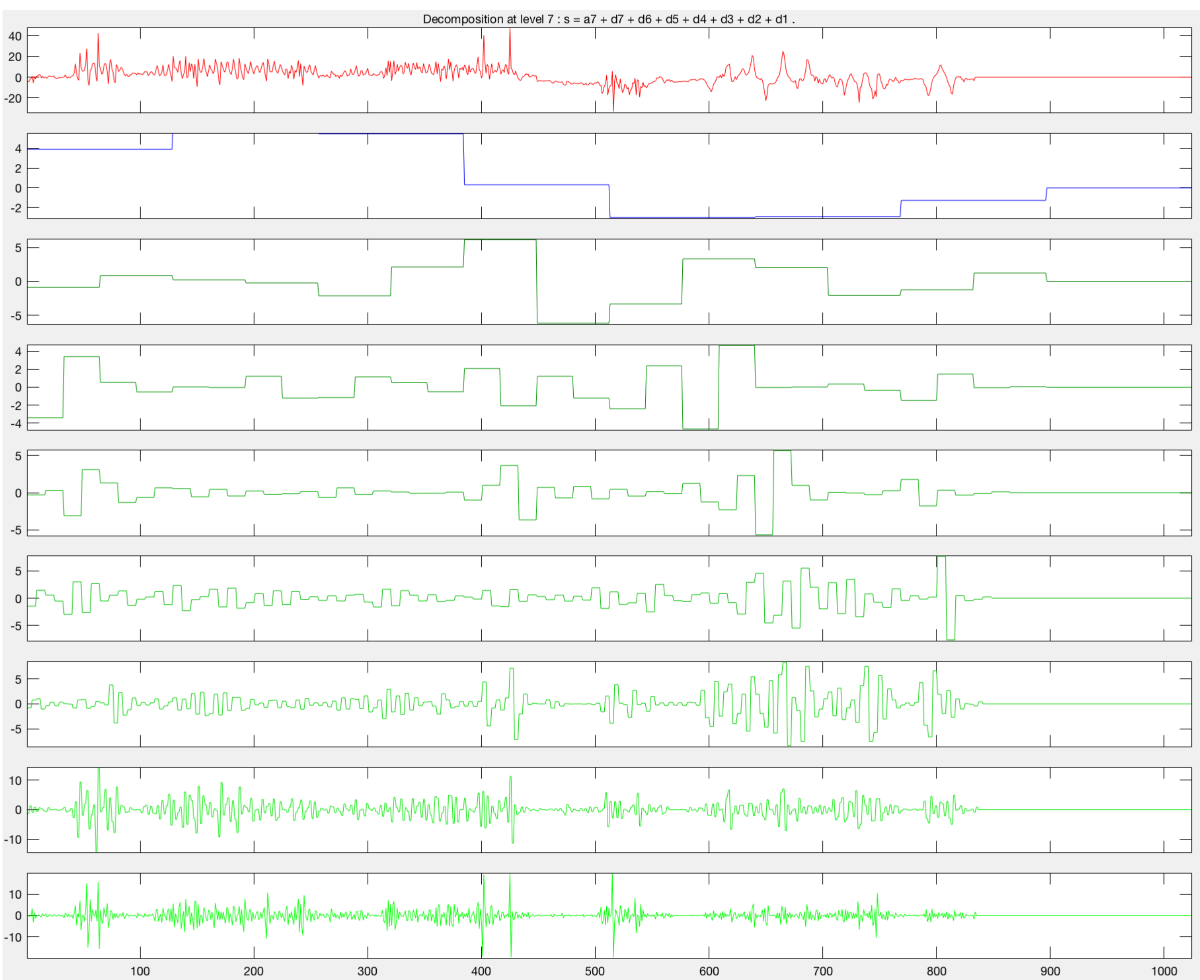
```
load lecture_acc.mat a1
s = zeros(1,1024);
s(1:size(a1,1)) = a1(:,1);
```

   Compute the Haar wavelet transform to level 7 (the routine *Wavelet_tr_test.m* is helpful), and decompose the signal into 8 parts:

$$s = a_7 + d_7 + d_6 + d_5 + d_4 + d_3 + d_2 + d_1.$$

   The decomposition is done by setting suitable parts of the wavelet transform to zero and applying the inverse transform. (Hint: each coefficient from the transform of s should be used for exactly one of the 8 parts; also, there's a reason why the parts are called a_7 and $d_1, \ldots, d_7$.)

   Present your result in this kind of multiple plot where the first plot is the signal s, and then the parts of the decomposition are plotted in the order given above. Explain how each part of the decomposition is obtained.



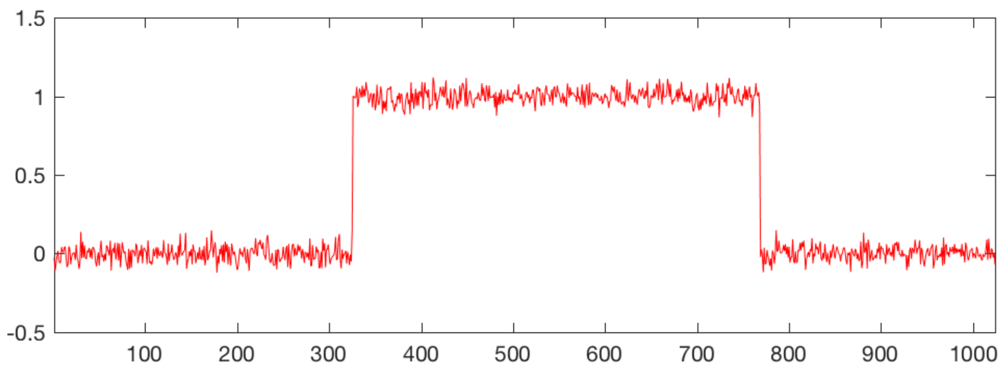
2. **Vanishing moments.**

   (a) Construct a one-dimensional signal (vector in $\mathbb{R}^{1024}$) that is zero in the first third of the elements, grows linearly in the second third and again vanishes in the final third. (I know that 1024 is not divisible by three. You need to do it approximately.)

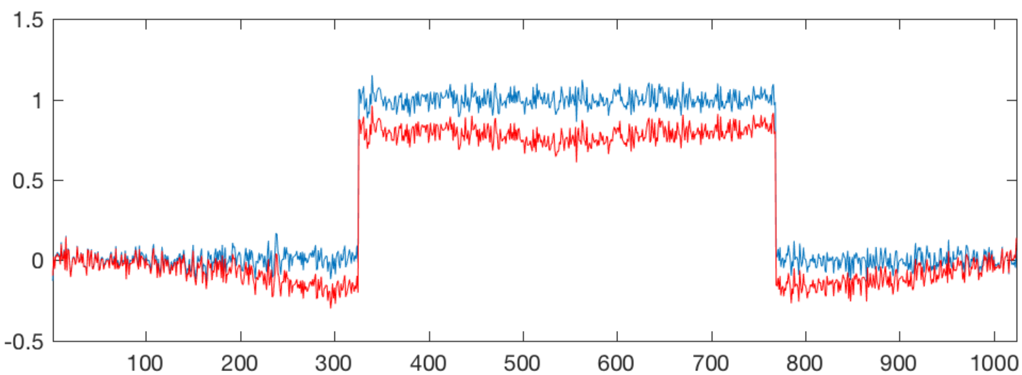
   (b) Compute Haar wavelet transform with maximal number of scales and plot as in Problem 1. How many of the coefficients are nonzero?

   (c) Compute the Daubechies 2 wavelet transform with maximal number of scales and plot as in Problem 1. Are there fewer nonzero coefficients than in (b)? If so, it is due to vanishing moments. Find out what that means.

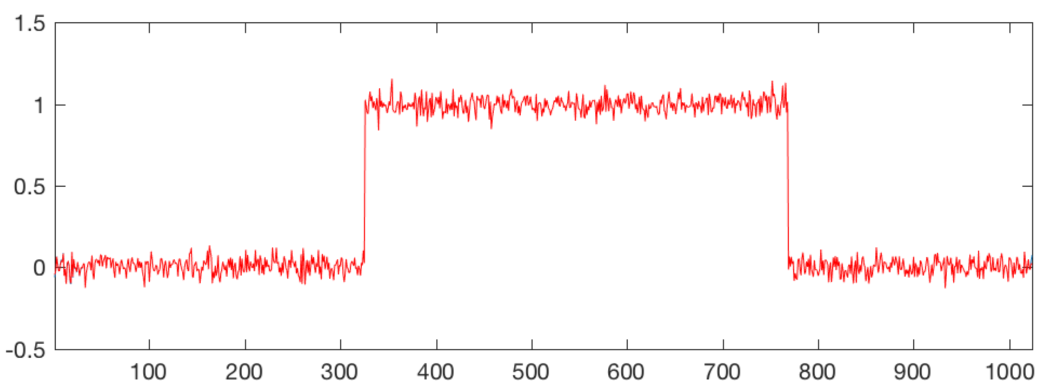
3. **Boundary effects in wavelet transform.** Running the routine *Wavelet_tr_test.m* for the step-and-noise signal using Haar wavelets, $J = 10$, and *no filtering of wavelet coefficients* results in the following image. The blue line shows the original signal and the red line shows the recovered signal, completely overlapping. This is expected since we first apply a transform and then the inverse transform: the result should be the original signal.



Here the routine *Wavelet_tr_test.m* for the same signal using Daubechies 2 wavelets, $J = 10$, and again no filtering. The only difference is using the Daubechies 2 filters of length 4. This is the result:



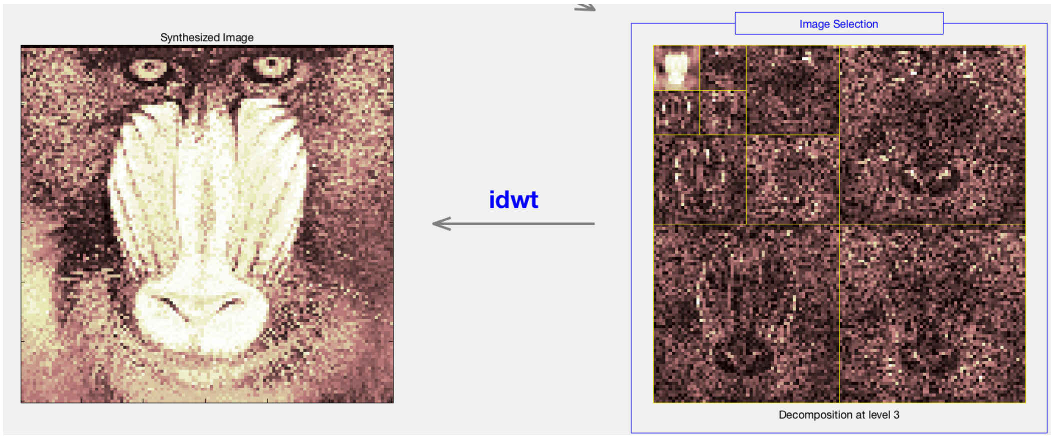
Clearly something is wrong. Now if we repeat the computation with only one change: $J = 5$. We get this:



Find out why this happens and explain it. Hint: It may help to compute a similar plot than in Problem 1 with Daubechies 2 wavelets and both choices $J = 5, 10$.

4. **Two-dimensional Haar wavelet transform.**

   (a) Take your own square (grayscale) image of size $2^M \times 2^M$. Download the files *wavetrans2Donce.m, wavetrans2Donce_inv.m, wavetrans2D.m* and *wavetrans2D_inv.m*. Produce a picture like this showing the image and its wavelet transform:

   (b) The transform image on the right above consists of 10 square areas: three big, three medium and four small sub-squares. Choose one of the four small sub-squares and replace the numbers in all the other sub-squares by zero. Apply inverse wavelet transform and show the resulting image next to the original. Explain what features of the original image are included in the filtered image.

   Repeat for one of the big sub-squares.

   (c) Now put everything in the transform to zero except one element. Replace the value of the remaining element by 1. Apply inverse transform and see what happens.

   Repeat for another pixel in the same sub-square. What changes? Why?

   Repeat for one nonzero pixel in each of the sub-squares. Explain what you see.

5. **Wavelet compression.**

   (a) Wavelet transform a grayscale image I_0, set $p\%$ of the smallest wavelet coefficients (smallest in absolute value, of course) to zero, and inverse transform. Call the result I_p. Calculate the square norm error in the result using formula $E(p) = \frac{\|I_0 - I_p\|}{\|I_0\|} \cdot \%$, where $\|M\|$ for a matrix M can be computed in Matlab as `norm(M(:))`; Plot $E(p)$ as a function of p. How large do you need to take p to get E to be higher than 10%?

   (b) Repeat (a), but instead of the square norm use the Structural Similarity Index (SSIM). They offer Matlab code at their website. Explain briefly the idea of SSIM.

   (c) Return to Problem 3 of Exercise 5. Compare the SSIM curves of wavelet compression and JPEG-type compression. Which one can compress more while retaining good SSIM value?