Guidelines

- Write your name and matriculation number on each sheet of paper.
- Only clearly readable exercise-elaborations are evaluated.
- Results have to be provided together with an evident way of calculation.
- Keep textual answers short and concise. Lengthy or vague statements won't gain points.

Exercise 7.1 (0.5 points)
Consider the matrix A:

$$A = \begin{bmatrix} 5 & 0 & 0 \\ 0 & -1 & -2 \\ 0 & 3 & 4 \end{bmatrix}$$

1. Calculate the eigenvalues $\lambda_{1,2,3}$ and the corresponding eigenvectors $\underline{v}_{1,2,3}$ of A.

2. Now construct a Hermitian matrix H (with the help of MATLAB) by the means of

$$H = \lambda_1 \underline{v}_1 \underline{v}_1^H + \lambda_2 \underline{v}_2 \underline{v}_2^H + \lambda_3 \underline{v}_3 \underline{v}_3^H$$

   and again use MATLAB to calculate the eigenvalues and eigenvectors of H. Why are not all eigenvectors and eigenvalues of the matrices A and H equal? Which one is equal and why?

3. Now let us construct a Hermitian matrix R that has the following eigenvalues and -vectors:

$$\begin{aligned} &\lambda_1 = 2 \Rightarrow X_1 = \{\underline{q}_1\} \\ &\lambda_2 = 3 \Rightarrow X_2 = \{\underline{q}_2, \underline{q}_3\} \\ &\underline{q}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \quad \underline{q}_2 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \quad \underline{q}_3 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \end{aligned}$$

Consider another vector $\underline{x}$:

$$\underline{x} = a\underline{q}_1 + b\underline{q}_2 + c\underline{q}_3.$$

4. Now let $a = 0, b = \frac{\sqrt{6}}{\lambda_2}, c = \frac{\sqrt{3}}{\lambda_2}$ and calculate

   $$\underline{y} = R\underline{x}$$

   for these factors.

5. Can you find a combination a,b,c so that

   $$\underline{y} = R\underline{x} = 0$$

   other than the trivial solution?

6. Reformulate the expression $R^n\underline{x}$ in terms of $\lambda_{1,2}$, $\underline{q}_{1,2,3}$, a,b,c.

Terms: Invariant subspaces of a matrix, Hermitian matrices.

Exercise 7.2 (0.5 points)
Consider the term

$$H^H H + \alpha I_n, \quad H \in \mathbb{C}^{m \times n}, \alpha \geq 0,$$

where the superscript ^{H} denotes the Hermitian adjoint (conjugate transpose) of a vector or matrix.

1. What are the properties of matrix (linear operator) $A = H^H H$? What can you tell about the eigenvalues and eigenvectors of such a matrix?

2. Find vectors $\underline{v}$ of unit norm that minimize / maximize the quadratic form

   $$\underline{v}^H \left(H^H H + \alpha I_n \right) \underline{v}$$

   as functions of the eigenvectors of $H^H H$ and α.

3. Find vectors $\underline{v}$ of unit norm that minimize / maximize the quadratic form

   $$\underline{v}^H \left(H^H H + \alpha I_n \right)^{-1} \underline{v}$$

   as functions of the eigenvectors of $H^H H$ and α.

4. Let $Q = \left[\begin{array}{ccc} \underline{h}_1 & \dots & \underline{h}_n \end{array} \right]$ denote the matrix whose columns $\underline{h}_i$ are the eigenvectors of $H^H H$. Now consider the matrix $G = Q + \beta I_n$; how do you have to choose $\beta \in \mathbb{C}$ in order for G to be a Hermitian matrix, how for an orthogonal matrix?

5. Given G from 4. with $\beta = 0$, find vectors $\underline{v}$ of unit norm that minimize / maximize the quadratic form

   $$\underline{v}^H G^H \left(H^H H + \alpha I_n \right) G \underline{v}.$$

Terms: Eigenvalues, spectral decomposition, quadratic form, Rayleigh quotient.

Exercise 7.3 (0.5 points)
Consider positive definite Hermitian matrices A and B.
Find vectors $\underline{f}$ that minimize the following terms:

1. $$\frac{\underline{f}^H A \underline{f}}{\underline{f}^H \underline{f}} + \frac{\underline{f}^H \underline{f}}{\underline{f}^H A \underline{f}}$$

2. $$\left(\frac{\underline{f}^H A \underline{f}}{\underline{f}^H \underline{f}} - 1 \right) \left(\frac{\underline{f}^H A \underline{f}}{\underline{f}^H \underline{f}} + 1 \right)$$

3. $$\frac{\underline{f}^H A \underline{f}}{\underline{f}^H B \underline{f}}$$

4. $$\frac{\underline{f}^H A \underline{f}}{\underline{f}^H \underline{f}} + (\underline{f}^H \underline{f} - 3)^2$$

Terms: Rayleigh quotient, quadratic form, Cholesky decomposition.

MATLAB-Exercise 7.1 (1 point)
In this exercise, you will learn about image compression utilizing a method called compressed sensing. Classical image compression techniques require to store the whole data before compression can be applied – compressed sensing allows to compress the data while sampling it. It is thus never required to store the whole data but just a bunch of samples, which is advantageous in the age of big data. You will learn how the sampling works and compare different methods to reconstruct the data from the samples.

Preliminaries: Consider a data vector $\underline{x} \in \mathbb{R}^N$. The data shall be sampled according to $\underline{y} = A\underline{x}$, where the columns of matrix $A \in \mathbb{R}^{M \times N}$ are the sampling basis functions, and $\underline{y} \in \mathbb{R}^M$ are the samples we actually store. Reconstructing $\underline{x}$ implies solving a system of M equations. If we choose $M < N$, the system is underdetermined – the solution for $\underline{x}$ is thus not unique and will in general differ from the original data. However, if the data vector $\underline{x}$ has only $K \ll N$ nonzero entries, it can be *reconstructed perfectly* from $M < N$ compressed sensing samples $\underline{y}$ if the following conditions are met:

- $\underline{x} \in \mathbb{R}^N$ is K-sparse, i.e., has $K \ll N$ nonzero entries,
- the number of samples in $\underline{y}$ obeys

$$M = \left\lceil cK \log\left(\frac{N}{K}\right) \right\rceil, \tag{1}$$

  (the proper choice of c is investigated in this exercise)
- at least $2K$ columns of A are linearly independent.

Compressed sensing reconstruction searches for the data vector $\underline{x}$ that satisfies

$$\underline{x}_{\mathrm{CS}} = \arg\min_{\underline{x}'} \|\underline{x}'\|_0 \quad \text{subject to} \quad A\underline{x} = \underline{y}. \tag{2}$$

The pseudo-norm $\|\underline{x}'\|_0$ counts the number of nonzero elements in $\underline{x}'$; we thus search for the sparsest solution that satisfies $A\underline{x}' = \underline{y}$. For comparison, using least squares reconstruction with the pseudo-inverse of A yields

$$\underline{x}_{\mathrm{LS}} = \arg\min_{\underline{x}'} \|\underline{x}'\|_2 \quad \text{subject to} \quad A\underline{x} = \underline{y}, \tag{3}$$

i.e., the solution with the minimal l_2-norm. *You will learn how this detail affects the reconstruction of an image.* Images are known to be compressible if represented in an appropriate basis as $\underline{x} = D\underline{z}$ ($\underline{x}$ is the compressed representation of $\underline{z}$ using basis D). This means that the energy of $\underline{z}$ is compressed into very few coefficients in $\underline{x}$, while the majority of coefficients in $\underline{x}$ stores almost no energy. We will use the Discrete Cosine Transform (DCT) to perform energy compression. Using compressible data instead of truly sparse data, the compressed sensing recovery yields an approximation of the true data. Let us investigate this now.

1. We first consider the case of *truly sparse* data.

   - Load `mesh16.mat` into Matlab, an array `z` should appear in the workspace. Visualize it using `imshow`.

   Handling monochrome images, we operate on 2-dimensional arrays. It is computationally efficient to perform processing on blocks rather than the whole image.

   - Consider blocks of size 16×16 pixels.
   - Perform a blockwise 2-dimensional DCT on all blocks of "image" `z`, use e.g. `dct2`.
   - Count the number of nonzero elements K per block in DCT domain. (Note that for the `mesh16` "image" `z`, K should be equal for all blocks.)
   - Compute sparsity

   $$\delta = \frac{K}{N}, \tag{4}$$

     where N is the total number of pixels per block.

   Let us now prepare compressed sensing. Note that the DCT could be incorporated into sampling directly as $\underline{y} = AD\underline{z} = A\underline{x}$, considering vectorized versions of the data (matrix D performs DCT.)

   - Compute M as a function of c, δ and N by utilizing Equations (1) and (4).
   - Determine M for the previously computed sparsity δ (should be equal for all block) and $c = 2.5$.
   - Generate a sensing matrix $A \in \mathbb{R}^{M \times N}$ with normal distributed entries $a_{i,j} \sim \mathcal{N}(0,1)$ (zero mean, unit variance). Normalize the columns of A so that they have unit l_2-norm. Omitting the details, this is one way of creating an appropriate sensing matrix for our problem.

   Now we are ready to perform compressed sensing and the reconstruction from the samples. Apply the following steps *block-wise* in DCT domain:

   - Perform compressed sensing using vectorized versions of the DCT blocks as $\underline{y} = A\underline{x}$, where $\underline{x} \in \mathbb{R}^N$ stores the DCT samples.
   - Perform compressed sensing reconstruction of $\underline{x}$ using function `CSrec` (details see Matlab code). This function approximately solves Equation (2) in an iterative manner. We obtain $\underline{x}_{\mathrm{CS}}$.
   - For comparison, perform least squares reconstruction of $\underline{x}$ from $\underline{y}$ using A (use Matlab function `pinv`). We obtain $\underline{x}_{\mathrm{LS}}$.

- After rearranging the reconstructed data vectors in blocks of size 16×16 (original order), apply the inverse 2D DCT (using e.g. `idct2`) to obtain the reconstructed image representations.

It is time to compare the results.

- Plot the original data, the compressed sensing reconstruction and the least squares reconstruction next to each other (using e.g. `subplot`). Also plot their DCT domain representations next to each other.
- Compute the Mean Squared Error (MSE) between original data and recovered data and display it in the respective pictures (e.g. as `xlabel`).
- Determine the value of c where $\underline{y} = A\underline{x}$ becomes overdetermined using the previously computed δ, Equation (1) where you can omit the ceiling operation $\lceil \cdot \rceil$, and Equation (4). Set c accordingly and run the program again – what changes do you observe? Why is that?

2. We are now ready to apply our Matlab script on real images that are *compressible* but in general not sparse.

   - Load `Lena256.jpg` into Matlab (use `z = double(imread('Lena256.jpg'));` to cast the data as double).
   - Since the image is not sparse but only compressible, we have to define our own δ this time. Try $\delta = 0.1$, i.e., assume that only 10% of the coefficients in DCT domain are "important". Use $c = 3$ to begin with. Run the script and compare the results.
   - As before, compute the value of c for which $M > N$ using $\delta = 0.1$. Use that value for c and compare the new results.

   You can now try different combinations of sparsity δ and the number of samples multiplier c. The achieved *compression factor* computes as $\beta = \frac{N}{M}$, the ratio between total samples and stored samples.

Matlab help: Use `imshow` to display images – note that you may have to typecast your image-matrix `X` with `uint8(X)` before plotting once you have performed numerical operations on it.
Terms: discrete cosine transformation, compressed sensing, sparse and compressible data, least squares, minimum norm solution.