Assignment 3

1. (**5**) Consider the matrices:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{1}$$

   - Derive (on a piece of paper) orthogonal projectors on range(A) and range(B).
   - Derive (on a piece of paper) QR decomposition of matrices A and B.

2. (**5**) Consider a particle of unit mass, which is prepared at $t = 0$ at $x = 0$ at rest $v = 0$. The particle is exposed to piece–wise constant external force f_i at $i - 1 < t \leq i$, with $i = 1, 2, ..., 10$. Let $a = (x(t = 10), v(t = 10))$ be a vector composed of coordinate and velocity of a particle at $t = 10$. Derive the matrix A such that $a = Af$ (note that A is of a shape 2×10). Using (a numerical) SVD decomposition, evaluate f of minimal Euclidean norm such that $a = (1, 0)$.

3. (**10**) Using SVD decomposition to perform the change of variables, solve the following linear systems $Ax = b$ for x in the least squares sense:

$$A = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/2 & 1/2 \\ -1/2 & -1/2 \end{bmatrix}, \quad b = (1, 0, 1)^T \tag{2}$$

$$A = \begin{bmatrix} 1/\sqrt{2} & 1/2 & -1/2 \\ -1/\sqrt{2} & 1/2 & -1/2 \end{bmatrix}, \quad b = (1, -1)^T. \tag{3}$$

4. (**5**) Consider the function $f(x) = 10\sin(x)$. Generate a dataset D that will consist of $n = 7$ points drawn as follows. For each point randomly draw x_i uniformly in $[0, 6]$ and define $y_i = f(x_i) + \epsilon_i$, where ϵ_i are iid standard gaussian random numbers. Plot D together with the true function $f(x)$. Fit a linear $l(x) = w_0 + w_1 x$ and a cubic $c(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$ models to D. Plot those models together with the dataset D.

5. (**10**) In the file `image_data.hdf` you will find the image A and the filter C. The image, stored in the matrix A is obtained from certain original image A_0 via convoluting it with the filter C and adding some noise. The filter C blurs an image, simultaneously increasing its size from 16×51 to 25×60. The filter is defined as a matrix, acting on an image A flattened to a vector a:

```
def mat2vec(A):
    A = np.flipud(A)
    a = np.reshape(A, np.prod(A.shape))
    return a
```

   with inverse transform, from vector a to matrix A given by

```
def vec2mat(a, shape):
    A = np.reshape(a, shape)
    A = np.flipud(A)
    return A
```

   In terms of the associated vectors a and a_0, one may write

$$a_0 \to a = Ca_0 + \epsilon, \tag{4}$$

   where ϵ is a vector of iid Gaussian random numbers. Your task will be to recover an original image A_0, being supplied by the image A and the filter C.

   - Plot the image A.
   - Explore how the filter C acts on images.

- A naive way to recover A_0 from A would be to solve $a = Ca_0$ for a_0. Is this system under– or over–determined? Using SVD of the filter matrix C, evaluate a_0 and plot the corresponding A_0.
- In order to improve the result, consider keeping certain fraction of singular values of C. Choose a value delivering the best empirical recovery quality.

6. (**10**) Consider the problem

$$\text{minimize} \quad \|Ax - b\|_2 \quad \text{subject to} \quad Cx = 0 \quad \text{with respect to} \quad x, \tag{5}$$

where A and C are matrices and x and b are vectors. Using the method of Lagrange multipliers, and assuming both $A^T A$ and $C(A^T A)^{-1} C^T$ to be invertible, derive explicit expression for optimal x.

7. (**20***) Here we will consider the problem of localization of points in a 2D plane. Consider n points, for which we have the *approximate* locations $r_i = (x_i, y_i)$. We measure k angles between certain points: $\theta_{ijk} = \angle(r_k - r_i, r_j - r_i)$. Our goal is to use the results of the measurements to improve the estimation of the locations r_i.

To be specific, consider $n = 3$ points, for which we have approximate locations $r_1 = (-1, 0)$, $r_2 = (0, 1)$, $r_3 = (1, 0)$ and $k = 1$ measurement $\theta_{123} = 9\pi/40$. Clearly, our estimates $r_{1,2,3}$ are not consistent with the measured angle and we have to adjust the estimate: $r_i \to \bar{r}_i = r_i + dr_i$ where dr_i should be found from the condition

$$\theta_{123} = \arccos \frac{(\bar{r}_3 - \bar{r}_1) \cdot (\bar{r}_2 - \bar{r}_1)}{|\bar{r}_3 - \bar{r}_1||\bar{r}_2 - \bar{r}_1|}, \tag{6}$$

which can be linearized in dr_i, assuming this correction will end up small. In this approach, one constructs a single (in general, k) equation for six (in general, $2n$) variables, so the system will typically by underdetermined. We can consider this system in the least squares sense, which amounts to determining the smallest correction to all r_i which makes the updated locations consistent with observations. In the particular numerical example above, one may find $dr_1 = (-h, 0)$, $dr_2 = (h, -h)$, $dr_3 = (0, h)$ where $h = \pi/80 \approx 0.04$.

Your task is to write the code, which will accept the current estimate of the positions r_i ($n \times 2$, float) and measurement results θ_{ijk}, which are specified by i) indices of points ($k \times 3$, int) and ii) angles (k, float); and will output the derived correction to the point positions dr_i ($n \times 2$, float). You can use the numerical example in this exercise to test your code. This dataset is provided in `localization_data_1.hdf` (it also includes the correct answer to the problem instance, dr_i which you can compare your result to). You can find additional dataset with the same structure in `localization_data_2.hdf`.