Exercise 1
Consider the picture of the Duomo di Milano given in file duomo.jpg.

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.image import imread

image_path = 'duomo.jpg'
img = imread(image_path)

A = np.mean(np.array(img, dtype = np.float64)/255, axis = -1)
plt.imshow(A, cmap = 'gray')
plt.axis('off')
A.shape
```

1. Compute the SVD associated with the picture, by using the standard algorithm, that gives the exact (up to round-off errors) decomposition. Then, plot the trend w.r.t. i of the singular values σ_i . Then, repeat the computation by considering the randomized SVD algorithm of rank $k = 25, 50, 100, 200$ and 400 . Plot the approximate singular values together with the exact singular values. Comment what you see.

2. We now want to compress the image.

   - Use the exact SVD to perform image compression for rank $k = 25, 50, 100, 200$ and 400. Plot the compressed image and compute the reconstruction error as a function of k (use the matrix p-norm with $p = 2$).
   - Repeat the same exercise by using the randomized SVD algorithm.
   - Repeat the same exercise by using the randomized SVD algorithm with a +50% oversampling, that is by increasing by 50% the number of columns in the matrix random P (round the quantity $1.5k$ to the closest integer).
   - Finally, plot the trend of the reconstruction error of the dataset as a function of k in the three cases. Comment on the results.

3. Evaluate the time needed to compute the SVD with the three approaches (SVD, randomized SVD, randomized SVD with oversampling) in the case $k = 200$.

Exercise 2
Consider the following function (where $\mathbf{x} = (x, y)$)

$$f(\mathbf{x}) = 5x^2 - 6xy + 5y^2 + 4x + 4y \tag{1}$$

1. Plot the function.

2. Compute analitically the value $\mathbf{x}_{min}$ where the function attains its minimum value and $f(\mathbf{x}_{min})$.

3. Write (1) as

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T H \mathbf{x} + \mathbf{d}^T \mathbf{x} + c. \tag{2}$$

Write explicitly the expressions for the matrix H, the vector $\mathbf{d}$ and the constant c. What is H ?

4. Consider the gradient descent (GD) method

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k), \tag{3}$$

where α is the learning rate. Combining (2) and (3) find the expression of α_{max} (maximum value of α) such that the GD method is convergent. (*Hint:* remember that for an iterative method to be convergent the spectral radius ρ of the iteration matrix must satisfy $|\rho| < 1$).

5. Find the value of α_{max} for the GD method applied to (1).

6. Implement the GD method and verify the conclusion drawn at the previous point. Use the stopping criterium $E < \epsilon$ where E is the absolute value of the difference of the last two functional values *i.e.*

$$E = |f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)|, \tag{4}$$

and ϵ is the required tolerance (take $\epsilon = 10^{-3}$). Moreover set the maximum number of iterations to 200 and the initial guess equal to $\mathbf{x}^0 = (-5, 7)$.

Exercise 3

Consider the following computational graph:

Figure 1: Computational Graph

The upper node in each layer computes $\sin(x + y)$ and the lower computes $\cos(x + y)$ with respect to its 2 inputs. For the first hidden layer, there is only a single input x, and therefore the values $\sin(x)$ and $\cos(x)$ are computed. The final node computes the product of the two inputs. The single input is denoted by x (value in radiants). Compute the numerical value of the partial derivative of the output with respect to x for $x = 1$ using the backpropagation algorithm. Explain clearly each step you have performed.