

# Programming Tasks - session 1

## Linear Equations

### Task 1 - Solving Equations

Write the `solve_equations` function that takes a matrix `A` (ndarray) and a vector `b` as inputs and returns the exact solution `x` for the well-determined (i.e., full-rank) linear matrix equation  $Ax = b$ . If the matrix  $A$  is singular or not square, the function should raise an error.

**Note:** You are not allowed to use predefined functions for solving linear equations (such as `numpy.linalg.solve`). You must implement the algorithm yourself.

### Task 2 - Inverse Matrix

Write a function named `inverse` that takes a matrix `A` as input and returns its inverse, `B`, such that the matrix multiplication  $A \cdot B = B \cdot A = I$ , where  $I$  is the identity matrix of the same size as  $A$ . The function should raise an error if the matrix  $A$  is not square or if the inversion process fails.

**Note:** You are not allowed to use predefined functions for matrix inversion (such as `numpy.linalg.inv`), but you may use functions from previous tasks.

### Task 3 (Optional) - Numba

Optimize the performance of the `solve_equations` and `inverse` functions by utilizing the Numba library. The more effectively you leverage the features of Numba, the higher your score will be.