Practical Class 01

Linear Algebra

1. Consider the following matrix,

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 3 \\ 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 2 & 1 & 5 \\ 1 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 1 \\ 3 & 0 \\ 1 & 1 \end{bmatrix},$$

Is the product listed below well-defined?

AB, AC, AD, BC, CD, DC, DA, BD

If so, calculate the matrix product. Check your final answer with Python or any other software you prefer.

2. For the following matrix, find the inverse using elementary row operation if exists.

(a) A=$\begin{bmatrix} 3 & 1 & 10 \\ 2 & 5 & 4 \end{bmatrix}$

(b) B=$\begin{bmatrix} 1 & 3 & 1 \\ 2 & 0 & 1 \\ 5 & 3 & 3 \end{bmatrix}$

(c) C=$\begin{bmatrix} 1 & 1 & 1 \\ 6 & 5 & 4 \\ 13 & 10 & 8 \end{bmatrix}$

3. The upward velocity of a rocket is given at three different times on the following table. The velocity data is approximated by a polynomial as

$$v(t) = at^2 + bt + c, 5 \leq t \leq 12.$$

(a) Set up a system of linear equations in matrix form to find $a,\ b,\ c$ of the velocity profile.

(b) Find the determinant and the rank of the coefficient matrix you find above.

Time(t)	Velocity (v)
5	106.8
8	177.2
12	279.2

Table 1: Velocity vs. time data for a rocket

(c) Use Gaussian elimination to find the inverse of the coefficient matrix and thus the solution of this system of linear equations.

4. The monitor on your computer is a truly magical device. When you look at the color white on your screen, you're not actually looking at white, and the same thing for the color yellow. There is actually no white or yellow pigment in your screen. What you are looking at is a mixture of the colors red, green, and blue displayed by extremely small pixels on your screen. These pixels are displayed in a matrix like pattern, and the saturation of each pixel tricks your brain into thinking it's a different color entirely when looked at from a distance.

   Let's say that you have a grayscale image that is 100 100 pixels in dimension. Each of those pixels can be represented in a matrix that is also 100 100, where the values in the matrix range from 0 to 255. Now, if you wanted to store that image, you would have to keep track of exactly 100 100 numbers or 10,000 different pixel values. This number can be extremely larger when it is a colored image. Luckily we have SVD which enable us to arrange the data matrix in such a way that the most important data is stored on the top and therefore we are able to compress the image (reduce the dimension of the matrix) without losing too much information. Let's have a look at a simplified case: Assume the data of the image is stored in a 3×2 matrix

$$M = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}$$

   Find the SVD of M and the one-dimensional approximation of matrix M.