# LAB 02 – PROJECT 01

## 1. PHƯƠNG PHÁP SỬ DỤNG CÁC BIẾN ĐỔI SƠ CẤP TRÊN DÒNG ĐỂ

### a. TÍNH ĐỊNH THỨC CỦA MA TRẬN VUÔNG

Cho ma trận vuông A, ta tìm $\boldsymbol{det(A)}$ như sau:

- Dùng các phép biến đổi sơ cấp trên dòng, biến đổi ma trận đã cho thành ma trận bậc thang.
- Sau đó, nhân dấu định thức với các phần tử trên đường chéo chính với nhau ta được định thức của ma trận vuông.
- Khi dùng các phép biến đổi trên dòng, phải tuân theo các nguyên tắc:
  + Hoán vị 2 dòng (hoặc 2 cột): đổi dấu định thức
  + Nhân 1 dòng với 1 số thực λ: định thức tăng lên λ lần
  + Định thức sẽ không đổi nếu ta cộng vào 1 dòng (hoặc 1 cột) với λ lần dòng (hoặc cột) khác.

**Ví dụ:**

$$\boldsymbol{A} = \begin{vmatrix} \mathbf{0} & \mathbf{2} & \mathbf{4} \\ \mathbf{2} & \mathbf{3} & \mathbf{-1} \\ \mathbf{-2} & \mathbf{-3} & \mathbf{6} \end{vmatrix} \xrightarrow{d3 \,\rightarrow\, d3 + \,d2} \begin{vmatrix} \mathbf{0} & \mathbf{2} & \mathbf{4} \\ \mathbf{2} & \mathbf{3} & \mathbf{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{5} \end{vmatrix} \xrightarrow{d1 \,\leftrightarrow\, d2} - \begin{bmatrix} \mathbf{2} & \mathbf{3} & \mathbf{-1} \\ \mathbf{0} & \mathbf{2} & \mathbf{4} \\ \mathbf{0} & \mathbf{0} & \mathbf{5} \end{bmatrix}$$

$$= \mathbf{-2 \times 2 \times 5 = -20}$$

### b. TÌM NGHỊCH ĐẢO CỦA MA TRẬN VUÔNG

Cho $A \in M_n$ khả nghịch, ta tìm $A^{-1}$ như sau:

- Lập ma trận $A \mid I_n$ (ma trận chia khối) bằng cách ghép ma trận đơn vị $I_n$ vào bên phải ma trận A
- Dùng phép biến đổi sơ cấp trên dòng để đưa $A \mid I_n$ về dạng $I_n \mid B$.
- Khi đó $B = A^{-1}$.

**Ví dụ:** $\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{5} & \mathbf{3} \\ \mathbf{1} & \mathbf{0} & \mathbf{8} \end{bmatrix}$ $\qquad B = \boldsymbol{A^{-1}} = \begin{bmatrix} \mathbf{-40} & \mathbf{16} & \mathbf{9} \\ \mathbf{13} & \mathbf{-5} & \mathbf{-3} \\ \mathbf{5} & \mathbf{-2} & \mathbf{-1} \end{bmatrix}$

$$A \mid I_3 = \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{array}\right] \xrightarrow[d3 = d3 - d1]{d2 = d2 - 2d1} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 5 & -1 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{d3 = d3 + 2d2} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & -1 & -5 & 2 & 1 \end{array}\right] \xrightarrow{d3 = d3 * (-1)} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

$$\xrightarrow[d2 = d2 + 3d3]{d1 = d1 - 3d3} \left[\begin{array}{ccc|ccc} 1 & 2 & 0 & -14 & 6 & 3 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow{d1 = d1 - 2d2} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] = I_3 \mid B$$

## 2. Ý TƯỞNG CÀI ĐẶT HÀM

### a. `calc_determinant_row_operation(matrix)`

- Dùng 2 biến **`echelon_matrix, sign`** để nhận ma trận bậc thang và dấu của định thức từ hàm **`Gauss_elimination`**

- Hàm **`Gauss_elimination:`**

**+ Bước 1.** Xác định cột trái nhất không chứa toàn số 0.

**+ Bước 2.** Đổi chỗ hai dòng, nếu cần thiết, để đưa số hạng khác 0 nào đó ở dưới về đầu cột nhận được ở Bước 1. **`sign ← sign * (-1)`** *# đổi dấu định thức*

**+ Bước 3.** Với số hạng đầu cột nhận được từ Bước 2 là $a \neq 0$, nhân dòng chứa nó với $1a$ để có số dẫn đầu 1 (**`leading1`**). (*Bước này tùy chọn*)

**`sign ← sign * 1/R[row][col]`** *# Nhân 1 dòng với số thực λ: định thức tăng lên λ lần*

**+ Bước 4.** Cộng một bội số thích hợp của dòng đầu cho từng dòng dưới để biến các số hạng bên dưới số dẫn đầu thành 0.

**+ Bước 5.** Che dòng đầu đã làm xong. Lặp lại các bước cho đến khi được ma trận bậc thang.

- Dùng vòng lặp duyệt qua từng dòng của **`matrix`** để nhân dấu định thức **`sign`** với các phần tử trên đường chéo chính.

### b. `invert_matrix_row_operation(matrix)`

- Tạo biến **`res`** để trả về kết quả.
- Kiểm tra nếu định thức của matrix **`det`** = 0 => Ma trận không khả nghịch => Trả về None.
- Ngược lại, tạo ma trận **`temp`** để ghép **`matrix`** và ma trận đơn vị lại với nhau.
- Tạo ma trận **`R`** để lưu ma trận **`temp`** (lúc này đã thành ma trận bậc thang rút gọn).
- Dùng vòng lặp duyệt qua ma trận **`R`**, thực hiện các phép biến đổi trên dòng để chuyển **`matrix`** thành ma trận đơn vị.

*Ví dụ:*

```
A = [[1, a, b, ...]
     [0, 1, c, ...]
     [0, 0, 1, ...]]

Để A thành ma trận đơn vị, ta thực hiện:
    d1 --> d1 - d2 * a
    d1 --> d1 - d3 * b
    ...
    d2 --> d2 - d3 * c
    ...
```

- Dùng vòng lặp duyệt qua ma trận **`R`** để lưu ma trận nghịch đảo vào **`res`**.

**Tham khảo:** lab01.ipynb
lab02_project01_2021_updated.ipynb