Calvin Sea Phanh

Lùca Bruno

Emma Durand

Programme matrice

Objectifs :

L'objectif est de réaliser une librairie pour permettre le calcul d'une matrice.

Rappels mathématiques :

Matrices

Définition :

Une matrice n × m est un tableau de nombres à n lignes et à m colonnes :

Exemple avec n = 2, m = 3 : $A=\begin{bmatrix}1 & 2 & 0\\4 & 3 & -1\end{bmatrix}$

On note A_{ij} l'élément situé à l'intersection de la ligne i et de la colonne j (la ligne est toujours nommée en premier).

Exemple : $A=\begin{bmatrix}\boldsymbol{A}11 & \boldsymbol{A}12 & \dots & \boldsymbol{A}1\boldsymbol{m}\\\boldsymbol{A}21 & \boldsymbol{A}22 & \dots & \boldsymbol{A}2\boldsymbol{m}\\\dots & \dots & \dots & \dots\\\boldsymbol{An}1 & \boldsymbol{An}2 & \dots & \boldsymbol{Anm}\end{bmatrix}$

On note $[A_{ij}]$ la matrice d'élément général A_{ij}. On a donc : **A** = $[A_{ij}]$

Déterminant d'une matrice :

Pour une matrice 2 × 2, on montre que la matrice inverse est donnée par :

$$\mathbf{A}=\begin{bmatrix}\boldsymbol{a} & \boldsymbol{b}\\\boldsymbol{c} & \boldsymbol{d}\end{bmatrix} \Rightarrow \mathbf{A^{-1}}=\frac{1}{\boldsymbol{ad}-\boldsymbol{bc}}\begin{bmatrix}\boldsymbol{a} & -\boldsymbol{b}\\-\boldsymbol{c} & \boldsymbol{a}\end{bmatrix}$$

Le nombre *ad - bc* est appelé *déterminant* de la matrice **A**, noté :

$$\begin{vmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d} \end{vmatrix} = |A| = \det(A)$$

La matrice inverse $\mathbf{A}^{-1}$ n'existe donc que si det **A** est différent de zéro.

La matrice **A** est singulière si det **A** = 0, régulière dans le cas contraire. Ce résultat se généralise à une matrice de dimension quelconque.

Propriétés des déterminants :

- $\det(\mathbf{A}^T) = \det(\mathbf{A})$
- $\det(\mathbf{AB}) = \det(\mathbf{A}) \times \det(\mathbf{B})$
- Si **A** est régulière, $\det(\mathbf{A}^{-1}) = 1 / \det(\mathbf{A})$ puisque $\det(\mathbf{AA}^{-1}) = \det(\mathbf{A}) \times \det(\mathbf{A}^{-1}) = \det(\mathbf{I}) = 1$
- Si **A** est orthogonale, $\det(\mathbf{A}) = \pm 1$ puisque $\det(\mathbf{AA}^T) = [\det(\mathbf{A})]^2 = \det(\mathbf{I}) = 1$

<u>Transposition d'une matrice :</u>

La transposée $\mathbf{A}^T$ (aussi notée **A**') d'une matrice **A** est la matrice obtenue en échangeant les lignes et les colonnes de **A** :

$$\mathbf{A}=\begin{bmatrix} 1 & 2 & 0 \\ 4 & 3 & -1 \end{bmatrix} \Leftrightarrow \mathbf{A}^T=\begin{bmatrix} 1 & 4 \\ 2 & 3 \\ 0 & -1 \end{bmatrix}$$

<u>Inversion d'une matrice :</u>

Une matrice **A** est dite *inversible* ou *régulière* s'il existe une matrice $\mathbf{A}^{-1}$ (appelée *matrice inverse*) telle que :

$\mathbf{A} \times \mathbf{A}^{-1} = \mathbf{A}^{-1} \times \mathbf{A} = \mathbf{I}$

Si $\mathbf{A}^{-1}$ n'existe pas, la matrice **A** est dite *singulière*

Propriétés :

- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$
- $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$
- $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
- La matrice **A** est dite *orthogonale* si $\mathbf{A}^{-1} = \mathbf{A}^T$

<u>Comatrice :</u>

On appelle **comatrice** (ou **matrice adjointe**) de A, la matrice carrée d'ordre *n*, notée com(A) (ou adj(A)) définie par :

$$\text{com(A)}=\begin{bmatrix} \boldsymbol{\Delta}11 & \boldsymbol{\Delta}12 & \dots & \boldsymbol{\Delta ln} \\ \boldsymbol{\Delta}21 & \boldsymbol{\Delta}22 & \dots & \boldsymbol{\Delta}2\boldsymbol{n} \\ \dots & \dots & \dots & \dots \\ \boldsymbol{\Delta n}1 & \boldsymbol{\Delta n}2 & \dots & \boldsymbol{\Delta nm} \end{bmatrix}$$

où Δ_{ij} est le cofacteur de l'élément a_{ij} de A défini à partir du mineur $|M_{ij}|$ par la relation :

$\Delta_{ij} = (-1)^{i+j}|M_{ij}|$

Fonctions crées (algo + explication) :

Algo :

```
def determinant2(matrice):
    determinant_matrice = (matrice[0][0] * matrice[1][1]) - (matrice[0][1] *
matrice[1][0])
    return determinant_matrice

def determinant3(matrice):
    determinant_matrice = (matrice[0][0] * ((matrice[1][1]*matrice[2][2])-
(matrice[2][1]*matrice[1][2]))) - (matrice[0][1]*
((matrice[1][0]*matrice[2][2])-(matrice[2][0]*matrice[1][2]))) +
(matrice[0][2]* ((matrice[1][0]*matrice[2][1])-(matrice[2][0]*matrice[1][1])))
    return determinant_matrice

def transposer3(matrice):
    resultat = [[0,0,0],[0,0,0],[0,0,0]]
    for ligne in range(3):
        for colonne in range(3):
            resultat[ligne][colonne] = matrice[colonne][ligne]
    return resultat


def transposer2(matrice):
    resultat = [[0,0,0],[0,0,0]]
    for ligne in range(2):
        for colonne in range(2):
            resultat[ligne][colonne] = matrice[colonne][ligne]
    return resultat

def comatrice3(matrice):
    a1 = (matrice[1][1] * matrice[2][2]) - (matrice[1][2] * matrice[2][1])
    a2 = ((matrice[1][0] * matrice[2][2]) - (matrice[1][2] * matrice[2][0])) *
(-1)
    a3 = ((matrice[1][0] * matrice[2][1]) - (matrice[1][1] * matrice[2][0]))
```

```
    b1 = ((matrice[0][1] * matrice[2][2]) - (matrice[0][2] * matrice[2][1])) *
(-1)
    b2 = ((matrice[0][0] * matrice[2][2]) - (matrice[0][2] * matrice[2][0]))
    b3 = ((matrice[0][0] * matrice[2][1]) - (matrice[0][1] * matrice[2][0])) *
(-1)
    c1 = ((matrice[0][1] * matrice[1][2]) - (matrice[0][2] * matrice[1][1]))
    c2 = ((matrice[0][0] * matrice[1][2]) - (matrice[0][2] * matrice[1][0])) *
(-1)
    c3 = ((matrice[0][0] * matrice[1][1]) - (matrice[0][1] * matrice[1][0]))
    comatrice_matrice = [[a1,a2,a3],[b1,b2,b3],[c1,c2,c3]]
    return comatrice_matrice

def comatrice2(matrice):
    comatrice_matrice = [(matrice[1][1]),(matrice[0][1])*(-
1)],[(matrice[0][1])*(-1),(matrice[0][0])]
    return comatrice_matrice

def inverse3(transpo, deter):
    inverse_matrice = [[0,0,0],[0,0,0],[0,0,0]]
    for i in range(3):
        for j in range(3):
            a1 = transpo[i][j] / deter
            inverse_matrice[i][j] = a1
    return inverse_matrice

def inverse2(transpo, deter):
    inverse_matrice = [[0,0],[0,0]]
    for i in range(2):
        for j in range(2):
            a1 = transpo[i][j] / deter
            inverse_matrice[i][j] = a1
    return inverse_matrice
```

Explication :

- La fonction « determinant2 » permet de calculer le déterminant d'une matrice 2x2. Elle a comme paramètre « matrice » qui est la variable que rentrera l'utilisateur pour calculer le déterminant de sa matrice.
  La variable « determinant_matrice » va nous permettre de calculer le déterminant d'une matrice. Pour cela, on utilise la formule du déterminant : elle prend le premier caractère de la matrice et le multiplie par le deuxième de la deuxième ligne de la matrice et soustrait ensuite ce résultat avec la multiplication du deuxième nombre de la première ligne avec le premier nombre de la deuxième ligne. (Voir Rappel cours).
  La fonction « return » va nous permettre de renvoyer la valeur obtenue du calcul de « determinant_matrice » et la garder pour la suite.

- La fonction « determinant3 » permet de calculer le déterminant d'une matrice 3x3. Elle a comme paramètre « matrice » qui comme la fonction précédente, est la variable que rentrera l'utilisateur pour calculer le déterminant de sa matrice.
  La variable « determinant_matrice » va calculer le déterminant d'une matrice 3x3. Cette variable utilise la formule pour calculer le déterminant d'une matrice 3x3. (Voir Rappel cours).
  La fonction « return » va nous permettre de renvoyer la valeur obtenue du calcul de « determinant_matrice » et la garder pour la suite.

- La fonction « transposer2 » permet de calculer la transposée d'une matrice 2x2. Elle a comme paramètre « matrice » (variable que rentrera l'utilisateur).
  La variable « transposer_matrice » va calculer la transposée d'une matrice 2x2. Pour cela on utilise la formule de la transposée : on inverse chaque nombre, la première ligne de la matrice va se placer dans la première colonne et la deuxième ligne va se placer dans la deuxième colonne.
  La fonction « return » va renvoyer la valeur de la fonction « transposer_matrice » (renvoyer et garder pour la suite la disposition de la nouvelle matrice).

- La fonction « transposer3 » permet de calculer la transposée d'une matrice 3x3. Elle a comme paramètre « matrice » (variable que rentrera l'utilisateur).
  On crée une nouvelle matrice pour transposer « resultat ».
  On crée une boucle pour dans une boucle pour qui prennent chacune comme valeur 0, 1 et 2.
  Le résultat (ligne, colonne) va se mettre à l'endroit où on va transposée.
  La matrice va ensuite transposée et ainsi la ligne va devenir la colonne.
  La fonction « return » va renvoyer la valeur de la fonction « transposer_matrice » (renvoyer et garder pour la suite la disposition de la nouvelle matrice).

- La fonction « comatrice3 » permet de calculer la comatrice d'une matrice 3x3. Elle a comme paramètre « matrice » (variable que rentrera l'utilisateur).
  Dans la fonction, on calcul chaque valeur a l'aide de la formule mathématique (voir Rappel cours).
  Ensuite, on créera une nouvelle matrice « matricee » et y on ajoutera les valeurs calculées pour créer la comatrice.
  Et ensuite on renvoie la nouvelle comatrice avec la fonction « return ».

- La fonction « comatrice2 » permet de calculer la comatrice d'une matrice 2x2. Elle a comme paramètre « matrice » (variable que rentrera l'utilisateur).
  Dans la fonction, on calcul chaque valeur à l'aide de la formule mathématique (voir Rappel cours).
  Ensuite, on créera une nouvelle matrice « matricee » et y on ajoutera les valeurs calculées pour créer la comatrice.
  Et ensuite on renvoie la nouvelle comatrice avec la fonction « return ».

- La fonction « inverse2» permet de calculer l'inverse d'une matrice 2x2. Elle possède deux paramètres :
  - « transpo » : est une variable qui sera ensuite remplacée par la fonction qui permet de calculer la transposée.
  - « deter » : est une variable qui sera ensuite remplacée par la fonction qui permet de calculer le déterminant.

  On crée ensuite une variable inverse_matrice (qui sera la nouvelle matrice). Dans la fonction, on crée une boucle dans une boucle qui va permettre de prendre chaque valeur de la matrice. Et dans

la deuxième boucle, on crée une variable qui va calculer l'inverse de chaque valeur de la matrice. Pour calculer cela, on utilise la formule mathématique. Chaque valeur de la transposée est divisé par le déterminant. La variable inverse_matrice va ensuite prendre chaque nouvelle valeur grâce à la variable a1 qui les calcule.
On renvoie ensuite notre variable inverse_matrice pour garder le résultat.

- La fonction « inverse3» permet de calculer l'inverse d'une matrice 3x3. Elle possède deux paramètres :
    - « transpo » : est une variable qui sera ensuite remplacée par la fonction qui permet de calculer la transposée.
    - « deter » : est une variable qui sera ensuite remplacée par la fonction qui permet de calculer le déterminant.

  On crée ensuite une variable inverse_matrice (qui sera la nouvelle matrice). Dans la fonction, on crée une boucle dans une boucle qui va permettre de prendre chaque valeur de la matrice. Et dans la deuxième boucle, on crée une variable qui va calculer l'inverse de chaque valeur de la matrice. Pour calculer cela, on utilise la formule mathématique. Chaque valeur de la transposée est divisé par le déterminant. La variable inverse_matrice va ensuite prendre chaque nouvelle valeur grâce à la variable a1 qui les calcule.
  On renvoie ensuite notre variable inverse_matrice pour garder le résultat.
-

Test de la librairie :

Matrice 3x3 :

```
1  from matrice_inversee import *
2
3  matrice = [[1,3,3],[3,5,6],[7,8,9]]
4
5  print(determinant3(matrice))
6  print(comatrice(matrice))
7  print(transposer3(comatrice(matrice)))
8  print(inverse(transposer3(comatrice(matrice)),determinant3(matrice)))
9
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                1: Python
444444444444]]
PS C:\Users\emmad> & python c:/Users/emmad/OneDrive/Bureau/textttttxtxtxtxtxtx.py
9
[[-3, 15, -11], [-3, -12, 13], [3, 3, -4]]
[[-3, -3, 3], [15, -12, 3], [-11, 13, -4]]
[[-0.3333333333333333, -0.3333333333333333, 0.3333333333333333], [1.6666666666666667, -1.3333333333333333, 0.3333333333333333], [-1.2222222222222223, 1.4444444444444444, -0.4444
444444444444]]
PS C:\Users\emmad> & python c:/Users/emmad/OneDrive/Bureau/textttttxtxtxtxtxtx.py
9
```

Matrice 2x2 :

```python
from matrice_inversee import *

matrice = [[1,3],[3,5]]

print(determinant2(matrice))
print(comatrice2(matrice))
print(transposer2(comatrice2(matrice)))
print(inverse2(transposer2(comatrice2(a)),determinant2(a)))

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\emmad> & python c:/Users/emmad/OneDrive/Bureau/textttxtxtxtxtxtx.py
[[0.3076923076923077, 0.07692307692307693], [0.07692307692307693, 0.11538461538461539]]
-4
([5, -3], [-3, 1])
[[5, -3, 0], [-3, 1, 0]]
[[0.3076923076923077, 0.07692307692307693], [0.07692307692307693, 0.11538461538461539]]
PS C:\Users\emmad> & python c:/Users/emmad/OneDrive/Bureau/textttxtxtxtxtxtx.py
```