

# TP d'analyse numérique n° 2 : inversion de matrices

## Introduction

Ce TP est à réaliser en binôme. Les fichiers Jupyter (en .ipynb) à compléter sont à votre disposition à l'adresse suivante :

<https://github.com/LucasMorlet/AnalyseNumerique/>

Pour éditer ce genre de fichier, vous pouvez utiliser Google Colab (pas besoin de vous authentifier, mais vous pouvez le faire pour synchroniser automatiquement votre fichier avec votre Google Drive). A la fin du TP, vous rendrez directement le fichier.

Le barème est donné à titre indicatif ci-dessous :

### Partie 1 : calcul de l'antécédent (4 points)

- Matrices diagonales
- Matrices triangulaires
- Pivot de Gauss
- Résolution Lower-Upper

### Partie 2 : inversion de matrices (3 points)

- Matrices diagonales
- Matrices triangulaires
- Pivot de Gauss

### Partie 3 : méthodes itératives (3 points)

- Résolution générique
- Méthode de Jacobi
- Méthode de Gauss-Siedel

## 1 Calcul de l'antécédent

### Objectif

Soit une matrice carrée  $A$  et deux vecteurs-colonnes  $x$  et  $b$  tels que  $Ax = b$ . L'objectif de cet exercice est de trouver  $x$  pour  $A$  et  $b$  donnés. Vous allez programmer différentes méthodes pour trouver la solution de cette équation selon les propriétés de la matrice  $A$  (diagonale, triangulaire..)

#### 1.1 Matrices diagonales

Pour commencer, programmez la fonction *resolution\_diagonale* qui renvoie l'antécédent pour une matrice diagonale  $A$  et une image  $b$ .

#### 1.2 Matrices triangulaires

Ensuite, programmez la fonction *resolution\_inferieure* qui renvoie l'antécédent pour une matrice triangulaire supérieure  $A$  et une image  $b$ . Pour résoudre cette équation, on "descend" le triangle en faisant les opérations suivantes sur chaque ligne  $i$  :

- On calcule la "partie-gauche" de la ligne :  $G_i = \sum_{j=0}^{i-1} A_{i,j}x_j$  (par définition  $G_0 = 0$ )
- On calcule  $x_i = (b_i - G_i)/A_{i,i}$

Exemple :

$$\begin{pmatrix} 6.6 & 0 & 0 \\ 0.1 & 1.2 & 0 \\ 5.6 & 5.9 & 9.0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.2 \\ 1.4 \\ 2.3 \end{pmatrix}$$

$$\text{Etape 1 : } 6.6x_0 = 4.2 \implies x_0 = \frac{4.2}{6.6}$$

$$\text{Etape 2 : } 0.1x_0 + 1.2x_1 = 1.4 \implies x_1 = \frac{1.4 - 0.1x_0}{1.2}$$

$$\text{Etape 3 : } 5.6x_0 + 5.9x_1 + 9.0x_2 = 2.3 \implies x_2 = \frac{2.3 - 5.6x_0 - 5.9x_1}{9.0}$$

Copiez-coller le code de *resolution\_inferieure* vers *resolution\_superieure*. Vous devriez pouvoir adapter la fonction en changeant uniquement les bornes des boucles de sorte à "remonter" le triangle.

### 1.3 Pivot de Gauss

Programmez la fonction *resolution\_gauss* qui résout l'équation pour n'importe qu'elle matrice carrée inversible par la méthode du pivot de Gauss. Le principe du pivot de Gauss est que l'on va rajouter des zéros dans la matrice jusqu'à obtenir une matrice diagonale. Comme vous avez déjà programmé la fonction *resolution\_superieure*, vous vous contenterez de rajouter des zéros sous la diagonale. Pour cela vous programmerez les étapes suivantes pour chaque ligne  $i$  de la matrice  $A$  :

- choix du pivot :  $p = A_{i,i}$  (par convention le pivot est choisi sur la diagonale)
- pour chaque ligne  $k$  en dessous de la ligne  $i$  :
  - calcul du coefficient de la ligne :  $c = A_{k,i}/p$
  - modifier la ligne  $k$  telle que  $L_k \leftarrow L_k - cL_i$  (ainsi  $A_{k,i} = 0$ )
  - diminuer  $b_k$  de  $cb_i$

Une fois ces étapes finies, il ne vous reste plus qu'une matrice triangulaire supérieure, dont vous avez déjà programmé la résolution.

NB : chaque modification effectuée sur une ligne  $k$  de la matrice doit être également effectuée sur le coefficient  $b_k$  de l'image pour trouver la solution.

Exemple :

$$\begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 1.7 & 1.3 & 2.4 \\ 7.1 & 6.8 & 2.3 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.2 \\ 1.4 \\ 2.3 \end{pmatrix} \quad \text{on pose } p = 3.6$$

$$\begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 1.7 - 3.6c & 1.3 - 2.1c & 2.4 - 5.5c \\ 7.1 & 6.8 & 2.3 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.2 \\ 1.4 - 4.2c \\ 2.3 \end{pmatrix} \quad \text{avec } c = 1.7/3.6$$

$$\begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 7.1 - 3.6c & 6.8 - 2.1c & 2.3 - 5.5c \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.2 \\ -0.6 \\ 2.3 - 4.2c \end{pmatrix} \quad \text{avec } c = 7.1/3.6$$

$$\begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 0 & 2.7 & -8.5 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.2 \\ -0.6 \\ -6 \end{pmatrix} \quad \text{on pose } p = 0.3$$

$$\begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 0 & 2.7 - 0.3c & -8.5 + 0.2c \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.2 \\ -0.6 \\ -6 + 0.6c \end{pmatrix} \quad \text{avec } c = 2.7/0.3$$

$$\begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 0 & 0 & -6.7 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.2 \\ -0.6 \\ -0.6 \end{pmatrix} \quad \text{puis résolution d'une matrice triangulaire supérieure}$$

## 1.4 Méthode Lower-Upper

Programmez la fonction *resolution\_LU* qui résout l'équation pour n'importe qu'elle matrice carrée inversible par la méthode Lower-Upper (abrégée LU). Cette méthode propose de décomposer la matrice  $A$  en deux matrices  $L$  et  $U$ , respectivement triangulaire inférieure et supérieure telles que  $A = LU$ . Ainsi résoudre  $Ax = b$  revient à résoudre  $(LU)x = b \Leftrightarrow L(Ux) = b$ . En posant  $y = Ux$  on obtient deux résolutions consécutives qui nécessitent uniquement de savoir résoudre pour une matrice triangulaire inférieure et supérieure : d'abord trouver  $y$  tel que  $Ly = b$  puis résoudre  $Ux = y$ .

Pour décomposer une matrice quelconque en une matrice triangulaire inférieure  $L$  et une matrice triangulaire supérieure  $U$ , vous effectuerez les étapes suivantes :

- initialisation :  $U = A$  et  $L = Id$
- appliquer le pivot de Gauss sur la matrice  $U$  jusqu'à avoir une matrice triangulaire supérieure.
- chaque fois que vous calculez un coefficient  $c = U_{k,i}/p$ , vous le placerez sur la matrice  $L$  à la position  $L_{k,i}$  (chaque fois que vous ajoutez un 0 à  $u$ , vous en enlevez un de  $L$ )

Une fois ces étapes réalisées vous pouvez utiliser les résolutions que vous avez déjà programmées pour conclure.

Exemple :

$$U = \begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 1.7 & 1.3 & 2.4 \\ 7.1 & 6.8 & 2.3 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{on pose } p = 3.6$$

$$U = \begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 1.7 - 3.6c & 1.3 - 2.1c & 2.4 - 5.5c \\ 7.1 & 6.8 & 2.3 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{avec } c = 1.7/3.6$$

$$U = \begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 7.1 - 3.6c & 6.8 - 2.1c & 2.3 - 5.5c \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0.47 & 1 & 0 \\ c & 0 & 1 \end{pmatrix} \quad \text{avec } c = 7.1/3.6$$

$$U = \begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 0 & 2.7 & -8.5 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0.47 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix} \quad \text{on pose } p = 0.3$$

$$U = \begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 0 & 2.7 - 0.3c & -8.5 + 0.2c \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0.47 & 1 & 0 \\ 2 & c & 1 \end{pmatrix} \quad \text{avec } c = 2.7/0.3$$

$$U = \begin{pmatrix} 3.6 & 2.1 & 5.5 \\ 0 & 0.3 & -0.2 \\ 0 & 0 & -6.7 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0.47 & 1 & 0 \\ 2 & 9 & 1 \end{pmatrix} \quad \text{il ne reste qu'à conclure}$$

## 2 Inversion de matrices

### Objectif

Soit une matrice carrée  $A$  à inverser. Vous allez programmer différentes méthodes pour inverser cette matrice selon ces propriétés (diagonale, triangulaire..).

### 2.1 Matrices diagonales

Pour commencer, programmez la fonction *inversion\_diagonale* qui renvoie l'inverse de la matrice diagonale  $A$ . Pour rappel, l'inverse d'une matrice diagonale est la matrice diagonale dont les coefficients sont l'inverse de ceux de la première.

### 2.2 Matrices triangulaires

Programmez la fonction *inversion\_inférieure* qui renvoie l'inverse de la matrice triangulaire inférieure  $A$  passée en paramètre. Pour ce faire, la méthode consiste à transformer progressivement la matrice en la matrice identité. En appliquant exactement les mêmes transformations sur la matrice identité, on obtient l'inverse de la matrice de départ.

Dans le cas d'une matrice triangulaire inférieure, les étapes à appliquer sur chaque ligne  $i$  (de haut en bas) sont les suivantes :

- Diviser la ligne par l'élément sur la diagonale  $L_i \leftarrow L_i/A_{i,i}$
- Faire de même pour la matrice inverse
- Pour chaque ligne  $k$  en dessous de la ligne  $i$  :
  - calcul du coefficient de la ligne :  $c = A_{k,i}$
  - modifier la ligne  $k$  telle que  $L_k \leftarrow L_k - cL_i$  (ainsi  $A_{k,i} = 0$ )
  - Faire la même modification sur la matrice inverse

Copiez-coller le code de *inversion\_inferieure* vers *inversion\_superieure*. Vous devriez pouvoir adapter la fonction en changeant uniquement les bornes des boucles de sorte à "remonter" le triangle.

## 2.3 Matrices quelconques

Pour une matrice inversible n'ayant aucune propriété particulière, il suffit de faire les opérations pour une matrice triangulaire inférieure et pour une triangulaire supérieure l'une à la suite de l'autre. Il s'agit de la méthode du pivot de Gauss pour inverser les matrices.

Programmez la fonction *inversion\_gauss* qui renvoie l'inverse d'une matrice quelconque  $A$  passée en paramètre.

## 3 Méthodes itératives

### Objectif

Soit une matrice carrée  $A$  et deux vecteurs-colonnes  $x$  et  $b$  tels que  $Ax = b$ . L'objectif de cet exercice est de trouver  $x$  pour  $A$  et  $b$  donnés. Vous allez programmer deux méthodes itératives pour trouver la solution de cette équation.

Les méthodes itératives consistent à proposer une solution (fausse) et à corriger cette solution de proche en proche. Si la matrice d'itération possède certaines propriétés, on est assurés que la solution converge vers un point-fixe qui est la solution de l'équation. Dans le cas présent, on s'assure de la convergence en étudiant une matrice dont la diagonale est strictement dominante.

### 3.1 Résolution générique

Résoudre  $Ax = b$  revient à résoudre  $(M - N)x = b$  avec  $A = M - N$ .

$$\begin{aligned}(M - N)x &= b \\ Mx - Nx &= b \\ Mx &= Nx + b \\ x &= M^{-1}(Nx + b)\end{aligned}$$

$x$  est donc le point fixe de la transformation  $\Phi(u) = M^{-1}(Nu + b)$ . Si la transformation est contractante, on peut construire une suite  $u_{n+1} = \Phi(u_n)$  qui converge vers  $x$ . L'avantage de cette méthode est qu'elle est applicable sur n'importe quelle matrice, il suffit de choisir  $M$  et  $N$  (pour rappel  $A = M - N$ ) telles que  $M$  est facilement inversible (*e.g.* diagonale ou triangulaire).

Programmez la fonction *resolution\_iterative* qui résout par itération l'équation et renvoie le point-fixe. Cette fonction prend en paramètre  $M^{-1}$ ,  $N$  et  $b$ . Pour l'instant, vous ne pouvez pas tester cette fonction, car il faut encore définir  $M$  et  $N$ . Pour cela, les deux méthodes suivantes sont à implémenter.

### 3.2 Méthode de Jacobi

La méthode de Jacobi est une méthode itérative. Elle décompose la matrice  $A$  en deux matrices :

- $M$  qui est la diagonale de  $A$  (donc facilement inversible)
- $N$  qui est tous les autres coefficients passés en négatif

Programmez la fonction *resolution\_jacobi* qui décompose  $A$  en  $M$  et  $N$ , inverse  $M$ , puis appelle la fonction *resolution\_iterative*. Pour inverser  $M$ , vous utiliserez la fonction *inversion\_diagonale* que vous avez implémentée dans l'exercice précédent.

### 3.3 Méthode de Gauss-Siedel

La méthode de Gauss-Siedel est aussi une méthode itérative. Elle décompose la matrice  $A$  en deux matrices :

- $M$  qui est la triangulaire inférieure de  $A$  (donc facilement inversible)
- $N$  qui est la triangulaire strictement supérieure de  $A$  passée en négatif

Programmez la fonction *resolution\_gauss\_siedel* qui décompose  $A$  en  $M$  et  $N$ , inverse  $M$ , puis appelle la fonction *resolution\_iterative*. Pour inverser  $M$ , vous utiliserez la fonction *inversion\_inferieure* que vous avez implémentée dans l'exercice précédent.