8 : Exemple mathématique complet : les matrices

On s'intéresse à une hiérarchie de classes représentant des matrices. Dans chaque classe, on utilisera un mode de stockage différent pour représenter des matrices ayant des propriétés différentes.

8.1 : Matrices particulières

On s'intéressera à des matrices de $\mathcal{M}_n(\mathbb{R})$. Ces matrices seront **denses**, **triangulaires** (inférieures et supérieures), **tridiagonales** ou **diagonales**.

Une matrice **dense** A sera représentée de la façon suivante :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$$

Les matrices **triangulaires** *inférieures* L et *supérieures* U auront les formes suivantes :

$$L = \begin{pmatrix} a_{1,1} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \qquad \text{et} \qquad U = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ 0 & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{n,n} \end{pmatrix}$$

Les matrices **tridiagonales** T vérifieront :

$$T = \begin{pmatrix} d_{1,1} & u_{1,2} & 0 & 0 & \dots & 0 \\ l_{2,1} & d_{2,2} & u_{2,3} & 0 & \dots & 0 \\ 0 & l_{3,2} & d_{3,3} & u_{3,4} & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & l_{n-1,n-2} & d_{n-1,n-1} & u_{n-1,n} \\ 0 & 0 & \dots & 0 & l_{n,n-1} & d_{n,n} \end{pmatrix}$$

On notera la diagonale par $(d_i)_{1\leq i\leq n}$, la sous-diagonale par $(l_i)_{1\leq i\leq n-1}$ et la sur-diagonale par $(u_i)_{1\leq i\leq n-1}$.

Les matrices **diagonales** D vérifieront :

$$D = \begin{pmatrix} d_{1,1} & 0 & 0 & 0 & \dots & 0 \\ 0 & d_{2,2} & 0 & 0 & \dots & 0 \\ 0 & 0 & d_{3,3} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & d_{n-1,n-1} & 0 \\ 0 & 0 & \dots & 0 & 0 & d_{n,n} \end{pmatrix}$$

8.2 : Méthodes des classes représentant ces matrices particulières

Dans la classe abstraite, on ne précise pas le stockage, mais certaines méthodes, notamment la trace, pourront être écrites indépendamment de la façon de stocker les coefficients.

Chaque classe stockera les coefficients de la matrice.

Cependant, dans chaque cas, les coefficients ne sont pas disposés de la même façon : on va donc proposer des méthodes propres à chaque classe pour gérer les accesseurs, mutateurs, et les fonctions mathématiques telles que l'addition, la transposition, le calcul du déterminant et l'inversion.

8.2.1 : Constructeur : comment stocker les coefficients ?

On considère des matrices de $\mathcal{M}_n(\mathbb{R})$. Les coefficients seront stockées dans une ou plusieurs listes, de la façon suivante :

Matrices denses : une liste de n listes de taille n

Matrices triangulaires : une liste de n listes, de tailles allant de n à 1

Matrices tridiagonales : trois listes, une de taille n (la diagonale) et deux de tailles $n-1$ (la sous-diagonale et la sur-diagonale)

Matrices diagonales : une liste de taille n

8.2.2 : Accesseurs et mutateurs

Matrices denses : `A[i, j] =` A_{ij}

Matrices triangulaires inférieures : `L[i, j] =` $\begin{cases} L_{ij} & \text{si } i \leq j, \\ 0 & \text{sinon.} \end{cases}$

Matrices triangulaires supérieures : `U[i, j] =` $\begin{cases} U_{ij} & \text{si } i \geq j, \\ 0 & \text{sinon.} \end{cases}$

Matrices tridiagonales : `T[i, j] =` $\begin{cases} l_i & \text{si } i = j - 1, \\ d_i & \text{si } i = j, \\ u_i & \text{si } i = j + 1, \\ 0 & \text{sinon.} \end{cases}$

Matrices diagonales : `D[i, j] =` $\begin{cases} d_i & \text{si } i = j, \\ 0 & \text{sinon.} \end{cases}$

8.2.3 : Addition

2 Matrices denses : `A1[i, j] + A2[i, j] =` $(A_1)_{ij} + (A_2)_{ij}$

2 Matrices triangulaires inférieures : `L1[i, j] + L2[i, j] =`
$\begin{cases} (L_1)_{ij} + (L_2)_{ij} & \text{si } i \leq j, \\ 0 & \text{sinon.} \end{cases}$

2 Matrices triangulaires supérieures : `U1[i, j] + U2[i, j] =`
$\begin{cases} (U_1)_{ij} + (U_2)_{ij} & \text{si } i \geq j, \\ 0 & \text{sinon.} \end{cases}$

2 Matrices tridiagonales : `T1[i, j] + T2[i, j] =` $\begin{cases} (l_1)_i + (l_2)_i & \text{si } i = j - 1, \\ (d_1)_i + (d_2)_i & \text{si } i = j, \\ (u_1)_i + (u_2)_i & \text{si } i = j + 1, \\ 0 & \text{sinon.} \end{cases}$

2 Matrices diagonales : `D1[i, j] + D2[i, j] =` $\begin{cases} (d_1)_i + (d_2)_i & \text{si } i = j, \\ 0 & \text{sinon.} \end{cases}$

1 Matrice dense et 1 Matrice diagonale : `A[i, j] + D[i, j] =`
$\begin{cases} a_{i,i} + d_i & \text{si } i = j, \\ a_{i,j} & \text{sinon.} \end{cases}$

8.2.4 : Transposée

Matrices denses : `tA[i, j] =` A_{ji}

Matrices triangulaires inférieures : `tL[i, j] =` $\begin{cases} L_{ji} & \text{si } i \geq j, \\ 0 & \text{sinon.} \end{cases}$ $\rightsquigarrow$ matrice triangulaire supérieure !

Matrices triangulaires supérieures : `tU[i, j] =` $\begin{cases} U_{ji} & \text{si } i \leq j, \\ 0 & \text{sinon.} \end{cases}$ $\rightsquigarrow$ matrice triangulaire inférieure !

Matrices tridiagonales : `tT[i, j] =` $\begin{cases} u_i & \text{si } i = j - 1, \\ d_i & \text{si } i = j, \\ l_i & \text{si } i = j + 1, \\ 0 & \text{sinon.} \end{cases}$

Matrices diagonales : `tD = D`

8.2.5 : Déterminant

On ne s'intéresse pas aux cas dense et tridiagonal ici : les formules sont trop compliquées... (voir TP4 pour une implémentation du déterminant dans le cas dense)

Matrices triangulaires inférieures : `det(L) =` $\prod_{i=1}^{n} l_{ii}$

Matrices triangulaires supérieures : `det(U) =` $\prod_{i=1}^{n} u_{ii}$

Matrices diagonales : `det(D) =` $\prod_{i=1}^{n} d_i$

8.2.6 : Inverse

On ne s'intéresse cette fois qu'au cas diagonal. (voir TP4 pour une implémentation de l'inverse dans le cas dense)

Matrices diagonales : `inv(D) =` $\begin{cases} \frac{1}{d_i} & \text{si } i = j, \\ 0 & \text{sinon.} \end{cases}$

Attention, l'inverse n'existe que si tous les d_i sont non nuls !