CHAPITRE 8: UTILISATION DE MAPLE EN ALGEBRE LINEAIRE

FONCTIONS DE BASE:

```
> restart;
```

Les vecteurs et les matrices peuvent être définis comme des tableaux rectangulaires, par la fonction ***array***, rencontrée dans le chapitre sur les tableaux :

Vecteur-ligne d'ordre 3 sans initialisation:

```
> A:=array(1..3);
```

$$A := \text{array}(1 .. 3, [\ ])$$

Vecteur-ligne d'ordre 3 avec initialisation:

```
> B:=array([x,y,z]);
```

$$B := [x, y, z]$$

Vecteur-colonne d'ordre 3 sans initialisation:

```
> U:=array(1..3,1..1);print(U);
```

$$U := \text{array}(1 .. 3, 1 .. 1, [\ ])$$

$$\begin{bmatrix} U_{1,1} \\ U_{2,1} \\ U_{3,1} \end{bmatrix}$$

Vecteur-colonne d'ordre 3 avec initialisation:

```
> V:=array([[x],[y],[z]]);
```

$$V := \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Matrice carrée d'ordre 2 sans initialisation:

```
> M:=array(1..2,1..2);print(M);
```

$$M := \text{array}(1 .. 2, 1 .. 2, [\ ])$$

$$\begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{bmatrix}$$

Matrice carrée d'ordre 2 avec initialisation:

```
> N:=array([[a,b],[c,d]]);
```

$$N := \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Opérations sur les vecteurs et les matrices , avec la fonction ***evalm*** :
Somme de vecteurs ou de matrices de même dimension avec +

```
> evalm(A+U);
```

$$\begin{bmatrix} A_1 + U_{1,1} \\ A_2 + U_{2,1} \\ A_3 + U_{3,1} \end{bmatrix}$$

Multiplication de vecteurs ou de matrices de dimensions compatibles avec &*

```
> evalm(M &* N);
```

$$\begin{bmatrix} M_{1,1}\,a + M_{1,2}\,c & M_{1,1}\,b + M_{1,2}\,d \\ M_{2,1}\,a + M_{2,2}\,c & M_{2,1}\,b + M_{2,2}\,d \end{bmatrix}$$

Calcul de la matrice inverse, lorsque N est inversible, avec ^(-1)

```
> evalm(N^(-1));
```

$$\begin{bmatrix} \frac{d}{a\,d - b\,c} & -\frac{b}{a\,d - b\,c} \\ -\frac{c}{a\,d - b\,c} & \frac{a}{a\,d - b\,c} \end{bmatrix}$$

Puissance n-ième avec ^n

```
> evalm(N^3);
```

$$\begin{bmatrix} (a^2 + b\,c)\,a + (a\,b + b\,d)\,c & (a^2 + b\,c)\,b + (a\,b + b\,d)\,d \\ (c\,a + d\,c)\,a + (b\,c + d^2)\,c & (c\,a + d\,c)\,b + (b\,c + d^2)\,d \end{bmatrix}$$

Multiplication par un scalaire avec *

```
> evalm(2*M-5*N);
```

$$\begin{bmatrix} 2\,M_{1,1} - 5\,a & 2\,M_{1,2} - 5\,b \\ 2\,M_{2,1} - 5\,c & 2\,M_{2,2} - 5\,d \end{bmatrix}$$

LES OUTILS DU PACKAGE LINALG:

Le package ***linalg*** contient plus d'une centaine d'outils en algèbre linéaire :

```
> with(linalg);
```

[*BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian*]

Les fonctions ***matrix*** et ***vector*** permettent de définir respectivement une matrice (tableau rectangulaire d'objets) ou un vecteur (tableau d'objets à une seule ligne) :

Définition d'une matrice 2x4 :

```
> P:=matrix([[s,t,u,v],[w,x,y,z]]);
```

$$P := \begin{bmatrix} s & t & u & v \\ w & x & y & z \end{bmatrix}$$

Définition d'un vecteur-ligne d'ordre 4 et d'une matrice-colonne d'ordre 3:

```
> V:=vector([a,b,c,d]);
```

$$V := [a, b, c, d]$$

```
> C:=matrix([[1],[2],[3]]);
```

$$C := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Une matrice a le type *matrix* , un vecteur a le type *vector*.

Opérations sur les matrices:

A la place de la fonction ***evalm***, on peut utiliser les outils spécifiques du package ***linalg***:

Multiplication de vecteurs ou de matrices de dimensions compatibles avec ***multiply***

```
> multiply(M,N);
```

$$\begin{bmatrix} M_{1,1}\,a + M_{1,2}\,c & M_{1,1}\,b + M_{1,2}\,d \\ M_{2,1}\,a + M_{2,2}\,c & M_{2,1}\,b + M_{2,2}\,d \end{bmatrix}$$

Calcul de la matrice inverse,lorsque N est inversible, avec ***inverse***

```
> inverse(N);
```

$$\begin{bmatrix} \frac{d}{a\,d - b\,c} & -\frac{b}{a\,d - b\,c} \\ -\frac{c}{a\,d - b\,c} & \frac{a}{a\,d - b\,c} \end{bmatrix}$$

Multiplication par un scalaire λ avec ***scalarmul***

```
> scalarmul(P,lambda);
```

$$\begin{bmatrix} \lambda\,s & \lambda\,t & \lambda\,u & \lambda\,v \\ \lambda\,w & \lambda\,x & \lambda\,y & \lambda\,z \end{bmatrix}$$

Test d'égalité de deux matrices avec ***equal***

```
> equal(M,N);
```

$$false$$

Etude d'une matrice:

Déterminant et trace d'une matrice carrée (somme des éléments diagonaux):

```
> N:=matrix([[1,2],[4,-3]]);det(N);trace(N);
```

$$N := \begin{bmatrix} 1 & 2 \\ 4 & -3 \end{bmatrix}$$

$$-11$$

$$-2$$

Rang d'une matrice:

```
> rank(N);
```

$$2$$

kernel(A); ou nullspace(A); détermine une base du noyau de A
colspace(A); détermine une base du sous-espace engendré par les vecteurs-colonnes de A
rowspace(A); détermine une base du sous-espace engendré par les vecteurs-lignes de A

```
> A:=matrix([[2,-2],[-1,1]]);kernel(A);
```

$$A := \begin{bmatrix} 2 & -2 \\ -1 & 1 \end{bmatrix}$$

$$\{[1, 1]\}$$

```
> colspace(A);
```

$$\left\{\left[1, \frac{-1}{2}\right]\right\}$$

```
> rowspace(A);
```

$$\{[1, -1]\}$$

Matrice caractéristique: $\lambda I - A$, polynôme caractéristique de A: det($\lambda I - A$).

```
> charmat(A,lambda);
```

$$\begin{bmatrix} \lambda - 2 & 2 \\ 1 & \lambda - 1 \end{bmatrix}$$

```
> charpoly(A,lambda);
```

$$\lambda^2 - 3\,\lambda$$

Valeurs propres de A (ce sont les racines du polynôme caractéristique):

```
> eigenvals(A);
```

$$0, 3$$

Un vecteur propre associé à la valeur propre λ est un vecteur X non nul tel que $A\,X = \lambda\,X$.

```
> eigenvects(A);
```

$$[0, 1, \{[1, 1]\}], [3, 1, \{[-2, 1]\}]$$

Le résultat affiché signifie que $\lambda = 0$ est une valeur propre d'ordre 1 de vecteur propre associé $X = (1, 1)$ et que $\lambda = 3$ est une valeur propre d'ordre 1 de vecteur propre associé $X = (-2, 1)$.

Modification d'une matrice:

```
> print(P);
```

$$\begin{bmatrix} s & t & u & v \\ w & x & y & z \end{bmatrix}$$

row(P,k) permet d'extraire la k-ième ligne de la matrice P. Le résultat est un vecteur.
col(P,k) permet d'extraire la k-ième colonne de la matrice P. Le résultat est un vecteur.

```
> row(P,2),col(P,3);
```

$$[w, x, y, z], [u, y]$$

rowdim(P) et ***coldim(P)*** retournent le nombre de lignes et de colonnes de la matrice *P*.

```
> rowdim(P),coldim(P);
```

$$2, 4$$

augment(M1,M2,...,Mn) permet de concaténer horizontalement les *n* matrices *M1,M2,...,Mn*:

```
> augment(A,P);
```

$$\begin{bmatrix} 2 & -2 & s & t & u & v \\ -1 & 1 & w & x & y & z \end{bmatrix}$$

stackmatrix(M1,M2,...,Mn) permet de concaténer verticalement les *n* matrices *M1,M2,...,Mn*:

```
> stackmatrix(A,N);
```

$$\begin{bmatrix} 2 & -2 \\ -1 & 1 \\ 1 & 2 \\ 4 & -3 \end{bmatrix}$$

copyinto(M,N,m,n) : copie *M* dans *N* , l'élément M[1,1] étant copié en N[m,n] :

```
> copyinto(N,P,1,2);
```

$$\begin{bmatrix} s & 1 & 2 & v \\ w & 4 & -3 & z \end{bmatrix}$$

swapcol(M,m,n) et ***swaprow(M,m,n)*** permettent d'échanger les colonnes (ou les lignes) *m* et *n* de la matrice *M*:

```
> swaprow(A,1,2);
```

$$\begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix}$$

transpose(M) permet de transposer la matrice *M*:

```
> print(M);transpose(M);
```

$$\begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{bmatrix}$$

$$\begin{bmatrix} M_{1,1} & M_{2,1} \\ M_{1,2} & M_{2,2} \end{bmatrix}$$

addcol(M,m,n,k) ou ***addrow(M,m,n,k)*** rendent une matrice obtenue à partir de *M* par l'opération *col n := col n + k * col m* (respectivement *lig n := lig n + k * lig m*) :

```
> print(N);addcol(N,1,2,mu);
```

$$\begin{bmatrix} 1 & 2 \\ 4 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \mu + 2 \\ 4 & 4\,\mu - 3 \end{bmatrix}$$

Matrices particulières:

```
> vandermonde([a,b,c,d]);
```

$$\begin{bmatrix} 1 & a & a^2 & a^3 \\ 1 & b & b^2 & b^3 \\ 1 & c & c^2 & c^3 \\ 1 & d & d^2 & d^3 \end{bmatrix}$$

```
> JordanBlock(expr,4);
```

$$\begin{bmatrix} expr & 1 & 0 & 0 \\ 0 & expr & 1 & 0 \\ 0 & 0 & expr & 1 \\ 0 & 0 & 0 & expr \end{bmatrix}$$

LES OUTILS DU PACKAGE LinearAlgebra:

A partir de la version 7, existe dans Maple un nouveau package ***LinearAlgebra*** avec des outils similaires à ceux du package ***linalg*** et destiné à supplanter l'ancien. En voici les fonctions:

```
> with(LinearAlgebra);
```

[*Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix, ConstantVector, CreatePermutation, CrossProduct, DeleteColumn, DeleteRow, Determinant, DiagonalMatrix, Dimension, Dimensions, DotProduct, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, LA_Main, LUDecomposition, LeastSquares, LinearSolve, Map, Map2, MatrixAdd, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, QRDecomposition, RandomMatrix, RandomVector, Rank, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, SubMatrix, SubVector, SumBasis, SylvesterMatrix, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip*]

On définit une matrice A dont on calcule le déterminant, la matrice et le polynôme caractéristique, ainsi que les valeurs propres et vecteurs propres associés:

```
> A:=Matrix([[1,-1,-1],[-2,2,3],[2,-2,-3]]);
  Determinant(A);
  CharacteristicMatrix(A,lambda);
  factor(CharacteristicPolynomial(A,lambda));
```

```
Eigenvalues(A);
Eigenvectors(A);
```

$$A := \begin{bmatrix} 1 & -1 & -1 \\ -2 & 2 & 3 \\ 2 & -2 & -3 \end{bmatrix}$$

$$0$$

$$\begin{bmatrix} -\lambda + 1 & -1 & -1 \\ -2 & -\lambda + 2 & 3 \\ 2 & -2 & -\lambda - 3 \end{bmatrix}$$

$$(\lambda - 1)\,\lambda\,(\lambda + 1)$$

$$\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

Exercice corrigé 8:

Rechercher les matrices carrées X d'ordre 2 telles que $X^2 + X = A$, où $A := \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

```
> with(linalg):A:=matrix(2,2,1);X:=matrix(2,2):
```

$$A := \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

```
> Y:=evalm(X^2+X-A);
```

$$Y := \begin{bmatrix} X_{1,1}^{\,2} + X_{1,2} X_{2,1} + X_{1,1} - 1 & X_{1,1} X_{1,2} + X_{1,2} X_{2,2} + X_{1,2} - 1 \\ X_{2,1} X_{1,1} + X_{2,2} X_{2,1} + X_{2,1} - 1 & X_{1,2} X_{2,1} + X_{2,2}^{\,2} + X_{2,2} - 1 \end{bmatrix}$$

```
> Y:={seq(seq(Y[i,j],j=1..2),i=1..2)};
```

$$Y := \{X_{1,1}^{\,2} + X_{1,2} X_{2,1} + X_{1,1} - 1,\ X_{1,2} X_{2,1} + X_{2,2}^{\,2} + X_{2,2} - 1,\ X_{1,1} X_{1,2} + X_{1,2} X_{2,2} + X_{1,2} - 1,$$
$$X_{2,1} X_{1,1} + X_{2,2} X_{2,1} + X_{2,1} - 1\}$$

```
> solutions:={solve(Y)}:
  for k to nops(solutions) do
    X:=matrix(2,2):assign(solutions[k]):print(X);
  end do:
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{-3}{2} & \frac{-1}{2} \\ \frac{-1}{2} & \frac{-3}{2} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

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

Donc 4 matrices solutions.

Travail dirigé 8:

TD 8.1:
Ecrire une procédure ***pcd(L::list)*** retournant à partir d'une liste donnée *L* la liste obtenue par permutation circulaire vers la droite des éléments de *L*.
Exemple:

```
> pcd([a,b,c,d,e,f,g]);
```

$$[g, a, b, c, d, e, f]$$

TD 8.2:
Ecrire une procédure ***circulante_droite(n::posint)*** retournant à partir de l'entier $n>0$ une matrice carrée d'ordre *n* telle que

- sa première ligne est constituée par les entiers de 1 à *n*.
- pour $k>1$, sa k-ième ligne est déduite de la ligne précédente par permutation circulaire vers la droite de ses éléments.

Exemple:

```
> circulante_droite(5);
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 4 \\ 4 & 5 & 1 & 2 & 3 \\ 3 & 4 & 5 & 1 & 2 \\ 2 & 3 & 4 & 5 & 1 \end{bmatrix}$$

TD 8.3: Test d'égalité de 2 matrices:

2 matrices *M* et *N* sont égales si et seulement si elles ont même nombre de lignes, même nombre de colonnes, et les mêmes coefficients .
Sans avoir recours à la fonction préexistante equal de Maple, écrire une procédure ***egal(M: : matrix , N: : matrix)*** retournant la valeur booléenne *true* si $M=N$ et *false* sinon.