Réduction des matrices etc..

```
> restart: with(linalg):
Warning, new definition for norm
Warning, new definition for trace
> A := matrix(3,3,[4,4,6,-6,-7,-12,3,4,7]);
```

$$A := \begin{bmatrix} 4 & 4 & 6 \\ -6 & -7 & -12 \\ 3 & 4 & 7 \end{bmatrix}$$

```
> p := charpoly(A,x);# x est la variable du polynome.
```

$$p := x^3 + 5\,x - 4\,x^2 - 2$$

```
> q := polytools[minpoly](A,x); # meme syntaxe pour le polynome
  minimal.
```

$$q := 2 - 3\,x + x^2$$

```
> factor(p);factor(q);
```

$$(x-2)\,(x-1)^2$$

$$(x-1)\,(x-2)$$

Que peut-on en déduire pour les valeurs propres de A, pour sa diagonalisation ?

```
> eigenvals(A);eigenvalues(A): # instructions synonymes.
```

$$2,\ 1,\ 1$$

```
> eigenvects(A):lv := eigenvectors(A); # meme remarque
```

$$lv := [2,\ 1,\ \{[1,\ -2,\ 1]\}],\ \left[1,\ 2,\ \left\{\left[\frac{-4}{3},\ 1,\ 0\right],\ [-2,\ 0,\ 1]\right\}\right]$$

```
> op(lv[1]);
```

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

Expliquer ce que font les instructions suivantes :

```
> P := [seq(op(x[3]),x=lv)]:
> PP := transpose(matrix(P)):
> DD := diag(seq(seq(x[1],j=1..x[2]),x=lv)):
```

En fait Maple sait faire cela aussi bien que nous :

```
> jordan(A); # met la matrice sous forme de Jordan.
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> jordan(A,'P'): # donne la forme de Jordan et la matrice de
  passage...
> evalm(P);
```

$$\begin{bmatrix} -4 & 3 & -2 \\ 6 & -6 & 0 \\ -2 & 3 & 1 \end{bmatrix}$$

Il existe une forme inerte Eigenvals avec paramètre optionnel pour la matrice de passage, (à vérifier pour le cas complexe en Release 4)

```
> A1 := matrix(2,2,[3,4,-4,3]) : Eigenvals(A1,Q1);
```

$$\text{Eigenvals}\left(\begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix}, Q1\right)$$

```
> evalf(%);
```

```
> evalm(Q1);
```

$$\begin{bmatrix} 1.000000000 & 0. \\ 0. & 1. \end{bmatrix}$$

Essayons maintenant de travailler avec des matrices symétriques.

```
> A := matrix(3,3,[2,-2,-2,-2,2,2,-2,2,2]);
```

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

```
> T := eigenvects(A);
```

$$T := [0, 2, \{[1, 0, 1], [1, 1, 0]\}], [6, 1, \{[-1, 1, 1]\}]$$

```
> P := [seq(op(x[3]),x=T)];
```

$$P := [[1, 1, 0], [1, 0, 1], [-1, 1, 1]]$$

```
> U := GramSchmidt(P,normalized); # fournit une base orthonormée à
  partir de P.
```

$$U := \left[\left[\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2}, 0\right], \left[\frac{1}{6}\sqrt{3}\sqrt{2}, -\frac{1}{6}\sqrt{3}\sqrt{2}, \frac{1}{3}\sqrt{3}\sqrt{2}\right], \left[-\frac{1}{3}\sqrt{3}, \frac{1}{3}\sqrt{3}, \frac{1}{3}\sqrt{3}\right]\right]$$

```
> V := matrix(U):
> orthog(V) ; # teste si V est orthogonale.
```

true

```
> W :=  transpose(V): # plutot que d'inverser !
> evalm(V&*(A&*W));
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

```
> definite(A,'positive_def');
```

false

```
> definite(A,'positive_semidef');
```

true

```
> # teste si la matrice possède la propriété, idem avec negative.
```

Maple calcule directement les exponentielles de matrice...

```
> A := matrix(2,2,[1,2,2,4]);
```

$$A := \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

```
> exponential(A);
```

$$\begin{bmatrix} \frac{4}{5}+\frac{1}{5}\mathbf{e}^5 & \frac{2}{5}\mathbf{e}^5-\frac{2}{5} \\ \frac{2}{5}\mathbf{e}^5-\frac{2}{5} & \frac{1}{5}+\frac{4}{5}\mathbf{e}^5 \end{bmatrix}$$

```
> B := exponential(A,t); # calcule directement exp(tA) !
```

$$B := \begin{bmatrix} \frac{4}{5}+\frac{1}{5}\mathbf{e}^{(5\,t)} & \frac{2}{5}\mathbf{e}^{(5\,t)}-\frac{2}{5} \\ \frac{2}{5}\mathbf{e}^{(5\,t)}-\frac{2}{5} & \frac{1}{5}+\frac{4}{5}\mathbf{e}^{(5\,t)} \end{bmatrix}$$

Les décompositions classiques

```
> B := vector(3,[a,b,c]):
> M := matrix(3,3,[10,9,1,9,10,5,1,5,9]):
```

gausselim utilise réalise une mise sous forme triangulaire avec stratégie de pivot partiel, i.e. les lignes peuvent être permutées !

```
> gausselim(M);
```

$$\begin{bmatrix} 9 & 10 & 5 \\ 0 & \frac{-19}{9} & \frac{-41}{9} \\ 0 & 0 & \frac{1}{19} \end{bmatrix}$$

```
> N := concat(M,B): N[1,1]:=0: evalm(N), gausselim(N);
```

$$\begin{bmatrix} 0 & 9 & 1 & a \\ 9 & 10 & 5 & b \\ 1 & 5 & 9 & c \end{bmatrix}, \begin{bmatrix} 9 & 10 & 5 & b \\ 0 & 9 & 1 & a \\ 0 & 0 & \frac{649}{81} & c-\frac{1}{9}b-\frac{35}{81}a \end{bmatrix}$$

En cas de problème Maple échange aussi les lignes sans avertir ! On peut fournir des paramètres optionnels pour récupérer le rang, le déterminant de la matrice ou limiter le nombre de colonnes à traiter.

ffgausselim réalise la même chose sans dénominat eurs.

```
> ffgausselim(M);
```

$$\begin{bmatrix} 9 & 10 & 5 \\ 0 & -19 & -41 \\ 0 & 0 & -1 \end{bmatrix}$$

```
> NN := matrix(3,3,[1,1,m,1,m,1,m,1,1]);
```

$$NN := \begin{bmatrix} 1 & 1 & m \\ 1 & m & 1 \\ m & 1 & 1 \end{bmatrix}$$

```
> gausselim(NN),gaussjord(M);
```

$$\begin{bmatrix} 1 & 1 & m \\ 0 & m-1 & 1-m \\ 0 & 0 & 2-m^2-m \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pretty stupid, isn't it ?

```
> MM := concat(M,B):gaussjord(MM);
```

$$\begin{bmatrix} 1 & 0 & 0 & -76\,b+65\,a+35\,c \\ 0 & 1 & 0 & -76\,a+89\,b-41\,c \\ 0 & 0 & 1 & 19\,c-41\,b+35\,a \end{bmatrix}$$

A little bit clever !

On peut utiliser gaussjord pour calculer l'inverse d'une matrice selon la méthode bien connue...

```
> N := concat(M,diag(1,1,1));
```

$$N := \begin{bmatrix} 10 & 9 & 1 & 1 & 0 & 0 \\ 9 & 10 & 5 & 0 & 1 & 0 \\ 1 & 5 & 9 & 0 & 0 & 1 \end{bmatrix}$$

```
> gaussjord(N);
```

$$\begin{bmatrix} 1 & 0 & 0 & 65 & -76 & 35 \\ 0 & 1 & 0 & -76 & 89 & -41 \\ 0 & 0 & 1 & 35 & -41 & 19 \end{bmatrix}$$

Le bloc de droite fournit donc l'inverse de M.
Le pivot de Gauss et la méthode de Gauss-Jordan sont étroitement liées à la décomposition LU d'une matrice.
Maple nous fournit cette décomposition avec tous les raffinements.

```
> LUdecomp(M,L='lo',U='up',U1='u1',R='r',P='per',det='d',rank='ran
  ');
```

$$\begin{bmatrix} 10 & 9 & 1 \\ 0 & \frac{19}{10} & \frac{41}{10} \\ 0 & 0 & \frac{1}{19} \end{bmatrix}$$

La valeur retournée est U matrice triangulaire supérieure (upper), que l'on peut aussi récupérer parmi les paramètres, L est la matrice triangulaire inférieure (lower) avec des 1 sur la diagonale.
Il y a une forme améliorée avec U écrite sous la forme U1*R où R est la forme de Gauss-Jordan (row reduced) et U1 triangulaire supérieure.
Si Maple a été obligé de permuter des lignes, on en trouve trace dans la matrice P (permutations)

M = P*L*U = P*L*U1*R

ATTENTION : si une des variables lo,up,.. est déja définie ça bloque !

```
> eval(lo),eval(up);
```

$$\left[\begin{array}{ccc} 1 & 0 & 0 \\ \frac{9}{10} & 1 & 0 \\ \frac{1}{10} & \frac{41}{19} & 1 \end{array}\right], \left[\begin{array}{ccc} 10 & 9 & 1 \\ 0 & \frac{19}{10} & \frac{41}{10} \\ 0 & 0 & \frac{1}{19} \end{array}\right]$$

```
> eval(r),eval(u1),eval(per);
```

$$\left[\begin{array}{ccc} 1 & 0. & 0. \\ 0 & 1 & 0. \\ 0 & 0 & 1 \end{array}\right], \left[\begin{array}{ccc} 10 & 9 & 1 \\ 0 & \frac{19}{10} & \frac{41}{10} \\ 0 & 0 & \frac{1}{19} \end{array}\right], \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]$$

Exercice Ecrire, à partir de ce qui précède, une procédure qui fournisse la décomposition LDU où L est triangulaire inf, U triangulaire sup avec chacune des 1 sur la diagonale et D diagonale. Vérifier que si A est symétrique alors L et U sont transposées.

```
> A := matrix(3,3,[6,2,3,2,5,1,3,1,3]):
> A1 := cholesky(A);
```

$$A1 := \left[\begin{array}{ccc} \sqrt{6} & 0 & 0 \\ \frac{1}{3}\sqrt{6} & \frac{1}{3}\sqrt{39} & 0 \\ \frac{1}{2}\sqrt{6} & 0 & \frac{1}{2}\sqrt{6} \end{array}\right]$$

```
> evalm(A1&*transpose(A1));
```

$$\left[\begin{array}{ccc} 6 & 2 & 3 \\ 2 & 5 & 1 \\ 3 & 1 & 3 \end{array}\right]$$

La décomposition obtenue à l'exercice précédent s'appelle décomposition de Cholesky et est définie dans le package d'algèbre linéaire.

Maple pratique aussi la décomposition QR avec Q orthogonale et R echelonnée (i.e. triangulaire).

```
> R := QRdecomp(A, Q='q', rank='r');
```

$$R := \left[\begin{array}{ccc} 7 & \frac{25}{7} & \frac{29}{7} \\ 0 & \frac{13}{7}\sqrt{5} & -\frac{3}{35}\sqrt{5} \\ 0 & 0 & \frac{3}{5}\sqrt{5} \end{array}\right]$$

```
> evalm(q),orthog(q),evalm(q&*R);
```

$$\begin{bmatrix} \frac{6}{7} & -\frac{4}{35}\sqrt{5} & -\frac{1}{5}\sqrt{5} \\ \frac{2}{7} & \frac{3}{7}\sqrt{5} & 0 \\ \frac{3}{7} & -\frac{2}{35}\sqrt{5} & \frac{2}{5}\sqrt{5} \end{bmatrix}, true, \begin{bmatrix} 6 & 2 & 3 \\ 2 & 5 & 1 \\ 3 & 1 & 3 \end{bmatrix}$$

Exercice à comparer avec les nouveaux PC de la salle 0060 !

```
> N := 50 : A := randmatrix(N,N): M := evalm(transpose(A)&*A):
> B := vector(N,[seq(i,i=1..N)]):
> t := time():linsolve(M,B):time()-t;
```

37.605

```
> t := time(): R := cholesky(M): time()-t;
```

58.442

```
> t := time():Z:=linsolve(R,B):time()-t;
Warning, computation interrupted
> #On se perd en conjectures !!!
> RR := map(evalf,R):
> t := time():Z:=linsolve(RR,B):time()-t;
```

.100

```
> t := time():X:=linsolve(transpose(RR),B):time()-t;
```

.175

```
> # A méditer !
> MM := M*1.0:
> t := time():linsolve(MM,B):time()-t;
```

1.432

```
> t := time():S := cholesky(MM):time()-t;
```

2.839

```
> t := time():Z:=linsolve(S,B):time()-t;
```

.158

Quelle conclusion peut-on en tirer ?