# TD 4 : Algèbre linéaire

Jean-Sébastien ROY, 1997-98

# Exercice 1

En prévision de ce qui va suivre, chargeons linalg.

```
> restart:with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

Définissons A, B et la fonction u :

```
> A:=X^4-1;B:=X^4-X;
```

$$A := X^4 - 1$$

$$B := X^4 - X$$

```
> u:=y->rem(A*y,B,X);
```

$$u := y \to \mathrm{rem}(A\,y, B, X)$$

On prend deux polynomes génériques P1 et P2 :

```
> P1:=add(a.i*X^i,i=0..3);P2:=add(b.i*X^i,i=0..3);
```

$$P1 := a0 + a1\,X + a2\,X^2 + a3\,X^3$$

$$P2 := b0 + b1\,X + b2\,X^2 + b3\,X^3$$

Et on vérifie que u est linéaire :

```
> expand(u(l1*P1+l2*P2)-l1*u(P1)-l2*u(P2));
```

$$0$$

Calcul direct des coefficients de la matrice de u :
A la ligne i, colonne j, on trouve le coefficient de X^(i-1) dans u(X^(j-1))

```
> U:=matrix(4,4,(i,j) -> coeff(u(X^(j-1)),X,i-1));
```

$$U := \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Son noyau :

```
> kernel(U);
```

$$\{[0, 1, 1, 1]\}$$

Que l'on peut visualiser de maniere plus agréable avec une fonction auxiliaire :

```
> vect2poly:= proc(v,x) local i;
  add(v[i]*x^(i-1),i=1..op(2,op(2,eval(v)))) end:
> map(vect2poly,kernel(U),X);
```

$$\{X + X^2 + X^3\}$$

Une base de son image (avec colspace) :

```
> map(vect2poly,colspace(U),X);
```

$$\{X^2 - X^3, X - X^3, 1 - X^3\}$$

Ses valeur propre et vecteurs propres :

```
> eigenvects(U);
```

$$\left[-\frac{3}{2}+\frac{1}{2}I\sqrt{3}, 1, \left\{\left[0, 1, -\frac{1}{2}-\frac{1}{2}I\sqrt{3}, -\frac{1}{2}+\frac{1}{2}I\sqrt{3}\right]\right\}\right],$$

$$\left[-\frac{3}{2}-\frac{1}{2}I\sqrt{3}, 1, \left\{\left[0, 1, -\frac{1}{2}+\frac{1}{2}I\sqrt{3}, -\frac{1}{2}-\frac{1}{2}I\sqrt{3}\right]\right\}\right], [-1, 1, \{[-1, 0, 0, 1]\}],$$



$$[0, 1, \{[0, 1, 1, 1]\}]$$

Ou, en plus lisible :

```
> map(vp ->
  print(valeur_propre=vp[1],multiplicité=vp[2],base=map(vect2po
  ly,vp[3],X)),[eigenvects(U)]):
```

$$valeur\_propre = -1,\ multiplicité = 1,\ base = \{-1 + X^3\}$$

$$valeur\_propre = 0,\ multiplicité = 1,\ base = \{X + X^2 + X^3\}$$

$$valeur\_propre = -\frac{3}{2}+\frac{1}{2}I\sqrt{3},\ multiplicité = 1,$$

$$base = \left\{\left(-\frac{1}{2}-\frac{1}{2}I\sqrt{3}\right)X + \left(-\frac{1}{2}+\frac{1}{2}I\sqrt{3}\right)X^2 + X^3\right\}$$

$$valeur\_propre = -\frac{3}{2}-\frac{1}{2}I\sqrt{3},\ multiplicité = 1,$$

$$base = \left\{\left(-\frac{1}{2}+\frac{1}{2}I\sqrt{3}\right)X + \left(-\frac{1}{2}-\frac{1}{2}I\sqrt{3}\right)X^2 + X^3\right\}$$

# Exercice 2

```
> restart:with(linalg):
Warning, new definition for norm
Warning, new definition for trace
> M:=evalm(matrix([[0,1,3],[3,0,1],[1,3,0]])/4);
```

$$M := \begin{bmatrix} 0 & \frac{1}{4} & \frac{3}{4} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} & 0 \end{bmatrix}$$

Apres avoir rentré la matrice M, on regarde numériquement la valeur aprochée de M^60

```
> evalf(evalm(M^60));
```

$$\begin{bmatrix} .3333333333 & .3333333333 & .3333333333 \\ .3333333333 & .3333333333 & .3333333333 \\ .3333333333 & .3333333333 & .3333333333 \end{bmatrix}$$

On en déduit donc que M^n doit converger, quand n tends vers l'infini, vers une matrice dont tous les éléments valent 1/3.
On va la diagonaliser et calculer une matrice de passage.
Calculons les valeurs propres/vecteurs propres de M :

```
> E:=[eigenvects(M)];
```

$$E := \left[[1, 1, \{[1, 1, 1]\}], \left[-\frac{1}{2}+\frac{1}{4}I\sqrt{3}, 1, \left\{\left[1, -\frac{1}{2}-\frac{1}{2}I\sqrt{3}, -\frac{1}{2}+\frac{1}{2}I\sqrt{3}\right]\right\}\right],\right.$$

$$\left.\left[-\frac{1}{2}-\frac{1}{4}I\sqrt{3}, 1, \left\{\left[1, -\frac{1}{2}+\frac{1}{2}I\sqrt{3}, -\frac{1}{2}-\frac{1}{2}I\sqrt{3}\right]\right\}\right]\right]$$

On extrait de ce résultat les valeur propres avec leur multiplicité DANS LE MEME ORDRE

que les vecteurs propres calculés précédement, (c'est très important).
Dans l'expression ci-dessous, e[1] est la valeur propre, et e[2] sa multiplicité. a$b = a,a,...,a (b fois)

```
> V:=map(e->e[1]$e[2],E);
```

$$V := \left[1, -\frac{1}{2}+\frac{1}{4}I\sqrt{3}, -\frac{1}{2}-\frac{1}{4}I\sqrt{3}\right]$$

D'ou une matrice de passage :
(op(e[3]) est la sequence des vecteurs propres formant une base du sous espace propre associé à la valeur propre e[1].
e est un élément de E, c'est a dire un triplet valeur propre/multiplicité/liste de vecteurs propres)
'op' permet d'enlever les accolades/crochets : op([1,2,3])=1,2,3

```
> P:=transpose(matrix(map(e -> op(e[3]),E)));
```

$$P := \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2}-\frac{1}{2}I\sqrt{3} & -\frac{1}{2}+\frac{1}{2}I\sqrt{3} \\ 1 & -\frac{1}{2}+\frac{1}{2}I\sqrt{3} & -\frac{1}{2}-\frac{1}{2}I\sqrt{3} \end{bmatrix}$$

La matrice diagonalisée (avec les valeurs propres dans le BON ORDRE), mise à la puissance n.

```
> Nn:=diag(op(map(e->e^n,V)));
```

$$Nn := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \left(-\frac{1}{2}+\frac{1}{4}I\sqrt{3}\right)^n & 0 \\ 0 & 0 & \left(-\frac{1}{2}-\frac{1}{4}I\sqrt{3}\right)^n \end{bmatrix}$$

On calcule la limite de cette matrice quand n-> infini.
Noter la présence de l'evalc qui est indispensable pour que limit fonctionne.
evalc mets (c+I*d)^n sous la forme a(n)+I*b(n) avec a(n) et b(n) réels.
Noter aussi qu'il faut appliquer limit sur chaque élément avec map.
(limit ne fonctionne pas directement sur une matrice)

```
> L:=map(e->limit(evalc(e),n=infinity),Nn);
```

$$L := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

D'ou a limite de M^n:

```
> evalm(P &* L &* P^(-1));
```

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

La fonction jordan permet d'obtenir directement la matrice diagonalisée (J) et la matrice de passage (Q) :



(le reste est identique). L'expression est :

```
> J:=jordan(M,'Q'):
```

# Exercice 3

```
> restart:with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

On entre la matrice A :

```
> A:=matrix([[2,1,1],[1,2,1],[0,0,3]]);
```

$$A := \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

Une matrice C quelconque :

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

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

La matrice suivante doit etre nulle si C commute avec A :

```
> N:=evalm(C &* A - A &* C);
```

$$N := \begin{bmatrix} C_{1,2}-C_{2,1}-C_{3,1} & C_{1,1}-C_{2,2}-C_{3,2} & C_{1,1}+C_{1,2}+C_{1,3}-C_{2,3}-C_{3,3} \\ C_{2,2}-C_{1,1}-C_{3,1} & C_{2,1}-C_{1,2}-C_{3,2} & C_{2,1}+C_{2,2}+C_{2,3}-C_{1,3}-C_{3,3} \\ -C_{3,1}+C_{3,2} & C_{3,1}-C_{3,2} & C_{3,1}+C_{3,2} \end{bmatrix}$$

On resoud donc apres avoir converti la matrice en ensemble (set) d'equation :

```
> solve(convert(N,set));
```

$$\{C_{3,2}=0, C_{2,3}=C_{1,3}, C_{3,3}=C_{1,1}+C_{1,2}, C_{2,2}=C_{1,1}, C_{2,1}=C_{1,2}, C_{3,1}=0, C_{1,2}=C_{1,2}, C_{1,1}=C_{1,1}, C_{1,3}=C_{1,3}\}$$

On affecte à la matrice C les valeurs trouvées :

```
> assign(");
> eval(C);
```

$$\begin{bmatrix} ?_{1,1} & ?_{1,2} & ?_{1,3} \\ C_{1,2} & C_{1,1} & C_{1,3} \\ 0 & 0 & C_{1,1}+C_{1,2} \end{bmatrix}$$

La ou il y a des ' ? ' la valeur est quelconque. On verifie qu'une telle matrice convient :

```
> map(eval,N);
```

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

Soit P un polynome quelconque en A (degré 2 suffit car A est de dimension 3) :

```
> P:=add(a.i*A^i,i=0..2);
```

$$P := a0 + a1\,A + a2\,A^2$$

On va essayer de trouver des valeurs pour a0, a1, a2 telles que P=C, c'est à dire, telles que la matrice suivante est nulle :

```
> evalm(P-C);
```

$$\begin{bmatrix} 2\,a1+5\,a2-C_{1,1}+a0 & a1+4\,a2-C_{1,2} & a1+6\,a2-C_{1,3} \\ a1+4\,a2-C_{1,2} & 2\,a1+5\,a2-C_{1,1}+a0 & a1+6\,a2-C_{1,3} \\ 0 & 0 & 3\,a1+9\,a2-C_{1,1}-C_{1,2}+a0 \end{bmatrix}$$

Meme methode que précédement pour résoudre :

```
> solve(convert(",set),{a0,a1,a2});
```

$$\{a2=-\frac{1}{2}C_{1,2}+\frac{1}{2}C_{1,3},\ a1=3\,C_{1,2}-2\,C_{1,3},\ a0=-\frac{7}{2}C_{1,2}+\frac{3}{2}C_{1,3}+C_{1,1}\}$$

```
> assign(");
> evalm(P);
```

$$\begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} \\ C_{1,2} & C_{1,1} & C_{1,3} \\ 0 & 0 & C_{1,1}+C_{1,2} \end{bmatrix}$$

```
> evalm(P-C);
```

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

Tout va bien.

# Exercice 4

```
> restart:with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

Les fonctions intermédiaires :

```
> poly2vect:= proc(p,d) local i:
  [seq(coeff(p,indets(p)[1],i),i=0..d)] end:
> vect2poly:= proc(v,x) local i:
  add(v[i]*x^(i-1),i=1..op(2,op(2,eval(v)))) end:
```

La fonction principale :

```
> sompoly:= p ->
  vect2poly(linsolve(vandermonde([seq(i,i=0..degree(p)+1)]),[se
  q(add(subs(indets(p)[1]=j,p),j=0..i),i=0..degree(p)+1)]),inde
  ts(p)[1]);
```

$$sompoly := p \to \mathrm{vect2poly}(\mathrm{linsolve}(\mathrm{vandermonde}([\mathrm{seq}(i, i=0\,..\,\mathrm{degree}(p)+1)]),$$
$$[\mathrm{seq}(\mathrm{add}(\mathrm{subs}(\mathrm{indets}(p)_1=j, p), j=0\,..\,i), i=0\,..\,\mathrm{degree}(p)+1)]), \mathrm{indets}(p)_1)$$

Prennons un exemple avant de détailler :

```
> sompoly(X^2);
```

$$\frac{1}{6}X+\frac{1}{2}X^2+\frac{1}{3}X^3$$

Soit sous la forme plus classique :

```
> factor(");
```

$$\frac{1}{6}X\,(X+1)\,(2\,X+1)$$

Détaillons :
On cherche à résoudre le système suivant (en a(0)..a(n+1), ou n est le degré de P) :
a(0)+a(1)*i+...+a(n+1)*i^(n+1)=sum(P(j),j=0..i) pour i de 0 à n+1
La matrice de ce système est donc une matrice de vandermonde :

```
> p:=X^3:
> vandermonde([seq(i,i=0..degree(p)+1)]);
```



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix}$$

L'indeterminée de P est indets(P)[1] (ici X) et l'on doit calculer la valeur de sum(P(j),j=0..i) pour i de 0 à n+1:

```
> subs(indets(p)[1]=j,p);
```

$$j^3$$

```
> add(",j=0..3);
```

$$36$$

```
> [seq(add(subs(indets(p)[1]=j,p),j=0..i),i=0..degree(p)+1)];
```

$$[0, 1, 9, 36, 100]$$

On aurait pu écrire, avec unapply :

```
> [seq(add(unapply(p,indets(p)[1])(j),j=0..i),i=0..degree(p)+1)
  ];
```

$$[0, 1, 9, 36, 100]$$

On résoud alors le système avec linsolve, la solution etant les coefficients a(0)..a(n+1).

```
> linsolve(vandermonde([seq(i,i=0..degree(p)+1)]),[seq(add(subs
  (indets(p)[1]=j,p),j=0..i),i=0..degree(p)+1)]);
```

$$\left[0, 0, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right]$$

Que l'on convertit en polynome :

```
> vect2poly(",indets(p)[1]);
```

$$\frac{1}{4}X^2+\frac{1}{2}X^3+\frac{1}{4}X^4$$

```
> factor(");
```

$$\frac{1}{4}X^2\,(X+1)^2$$