# TD Maple n°7 : Matrices et algèbre linéaire
## Corrigé des exercices

Lundis 16 et 23 mars 2009

## 1 Exercice 1

Voici une procédure permettant d'effectuer un produit matriciel (maintenant, vous êtes assez grands pour la comprendre tout seuls) :

```
[>multmat :=proc(A,B)
local C,i,j,k ;
if coldim(A) <> rowdim(B) then print(''matrices incompatibles'')
else
    C :=matrix(rowdim(A),coldim(B)) ;
    for i from 1 to rowdim(C) do
        for j from 1 to coldim(C) do
            C[i,j] :=0 ;
            for k from 1 to coldim(A) do C[i,j] :=C[i,j]+A[i,k]*B[k,j] od ;
        od ;
    od ;
fi ;
C ;
end proc ;
```

## 2 Exercice 2 : Noyau, image d'une application linéaire

1. On commence par définir les polynômes $A$ et $B$ :
   ```
   [>a :=x^4-1 ;
   [>b :=x^4-x ;
   ```

   Puis on définit l'application $u$ :
   ```
   [>u :=p->rem(p*a,b,x) ;
   ```

   Je rappelle que l'instruction `rem(Polynome1,Polynom2,x)` renvoie le reste de la division euclidienne de Polynome1 par Polynom2. x est l'indéterminée des polynômes.
   Pour vérifier que $u$ est linéaire, on crée deux polynômes quelconques de $\mathbb{R}_3[X]$ :
   ```
   [>P1 :=a1*x^3+b1*x^2+c1*x+d1 ;
   [>P2 :=a2*x^3+b2*x^2+c2*x+d2 ;
   ```

   Puis on vérifie que $u(P1+P2)-(u(P1)+u(P2))=0$ :
   ```
   [>u(P1+P2)-(u(P1)+u(P2)) ;
   [>simplify(%) ;
   ```

   Et que $u(\lambda P1)-\lambda u(P1)=0$ :
   ```
   [>u(λ P1)-λ u(P1) ;
   [>simplify(%) ;
   ```

2. La matrice recherchée est la matrice dont les vecteurs colonnes sont $u(1), u(X), u(X^2), u(X^3)$ exprimés dans la base canonique de $\mathbb{R}_3[X]$. Or l'expression des $(u(X^j))_{0\leq j\leq 3}$ dans la base canonique de $\mathbb{R}_3[X]$ s'obtient directement grâce à la fonction `polyvect`. Donc, avec un peu d'astuce, on peut définir très facilement la matrice $A$ :

```
A :=matrix(4,4,(i,j)->polyvect(u(x^(j-1)),3)[i]) ;
```

On obtient la matrice :

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

3. Pour avoir une base du noyau de $u$ :
`[>kernel(A) ;`

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

Donc $ker(u) = vect\{[0, 1, 1, 1]\}$.
Pour avoir une base de l'image de $u$ :
`[>colspace(A) ;`

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

Donc $Im(u) = vect\{[1, 0, 0, -1], [0, 1, 0, -1], [0, 0, 1, -1]\}$.
Pour avoir les valeurs propres de $u$ ainsi que les vecteurs propres associés :
`[>eigenvects(A) ;`
J'ai la flemme de taper le résulat.

# 3 Exercice 3

1. On commence par définir la fonction a qui au couple $(i, j)$ associe $a_{ij}$ :

```
[>a :=proc(i,j)
if i=j then if i<5 then 1-(1/2)^(5-i)
else 1
fi ;
else if i=j+1 then (1/2)^(5-j)
else 0
fi ;
fi ;
end proc ;
```

Puis on définit la matrice A :
`A :=matrix(5,5,a) ;`

2. Pour imprimer les couples $(n, u_n)$ pour $n$ variant de 1 à 50, on fait une boucle for dans laquelle la matrice B prend successivement les valeurs $A^n$ pour $n$ allant de 1 à 50. Au k-ème passage dans la boucle, on imprime k et `multiply(B,V)[5,1]` qui est la 5-ème coordonnée du vecteur $Bv = A^k v$ :

```
[>V :=matrix(5,1,0) :V[1,1] :=1 :evalm(V) ;
```

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

```
[>B :=A :
[>for k from 1 to 50 do print([k,multiply(B,V)[5,1]]) ; B :=multiply(B,A) od :
```

3. Pour tracer le graphe $n \mapsto u_n$, on crée une liste L qui contient les couples $[n, u_n]$ en faisant une boucle comme ci-dessus :

```
[>L :=[] :B :=A :
[>for k from 1 to 50 do L :=[op(L),[k,evalf(multiply(B,V)[5,1])]] :
B :=multiply(B,A) ;
od ;

[>plot(L) ;
```

4. Pour trouver la plus petite valeur pour laquelle $u_n$ est supérieur à 0.99, on fait une boucle while :

```
[>uk :=multiply(A,V) ;
[>k :=0 ;
[>while(uk[5,1]<0.99) do
uk :=multiply(A,uk) ;
k :=k+1 ;
od ;
```

Remarquer que si à l'étape k, $uk = A^k v$ alors à l'étape k+1, on a bien $uk :=$ `multiply(A,uk)` $= A^{k+1}v$
Finalement, on regarde la valeur de k à la sortie de la boucle :

```
[>k ;
```

$$88$$

Donc la plus petite valeur de $n$ pour laquelle $u_n$ dépasse 0.99 est 89.