Corrigé du TP3

```
> #Exercice 1
> restart:with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
> #Question2
> ZRo:=diag(0$15);
```

$$ZRo := \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

```
> M:=matrix(10,10,(i,j)->if i=j then 0 else i+j fi);
```

$$M := \begin{bmatrix} 0&3&4&5&6&7&8&9&10&11\\ 3&0&5&6&7&8&9&10&11&12\\ 4&5&0&7&8&9&10&11&12&13\\ 5&6&7&0&9&10&11&12&13&14\\ 6&7&8&9&0&11&12&13&14&15\\ 7&8&9&10&11&0&13&14&15&16\\ 8&9&10&11&12&13&0&15&16&17\\ 9&10&11&12&13&14&15&0&17&18\\ 10&11&12&13&14&15&16&17&0&19\\ 11&12&13&14&15&16&17&18&19&0 \end{bmatrix}$$

```
> I20:=diag(1$20);
```

$$I20 := \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

```
> N:=matrix(3,3,(i,j)->x^(i-1)*y^(j-1));
```

$$N := \begin{bmatrix} 1 & y & y^2 \\ x & x\,y & x\,y^2 \\ x^2 & x^2\,y & x^2\,y^2 \end{bmatrix}$$

```
> x:=1;
```

$$x := 1$$

```
> eval(N);evalm(N);# ne donnent rien; il faut utiliser la commande
  "map"
```

$$\begin{bmatrix} 1 & y & y^2 \\ x & x\,y & x\,y^2 \\ x^2 & x^2\,y & x^2\,y^2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & y & y^2 \\ x & x\,y & x\,y^2 \\ x^2 & x^2\,y & x^2\,y^2 \end{bmatrix}$$

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

$$\begin{bmatrix} 1 & y & y^2 \\ 1 & y & y^2 \\ 1 & y & y^2 \end{bmatrix}$$

```
> x:='x';# on libère x
```

$$x := x$$

```
> T:=band([3,2,-1],5);
```

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

```
> #Question3
> A:=matrix(3,3,[1,2,-3,5,0,2,1,-1,1]);
```

$$A := \begin{bmatrix} 1 & 2 & -3 \\ 5 & 0 & 2 \\ 1 & -1 & 1 \end{bmatrix}$$

```
> evalm(scalarmul(A,sqrt(2))+scalarmul(N,3));
```

$$\begin{bmatrix} \sqrt{2}+3 & 2\sqrt{2}+3y & -3\sqrt{2}+3y^2 \\ 5\sqrt{2}+3x & 3xy & 2\sqrt{2}+3xy^2 \\ \sqrt{2}+3x^2 & -\sqrt{2}+3x^2y & \sqrt{2}+3x^2y^2 \end{bmatrix}$$

```
>
> #Exercice2
  restart:with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

```
> #Question 1
> u:=vector([1,3,5]);v:=vector([-2,3,0]);w:=vector([0,-3,6]);n:=Vector([a,b,c]);
```

$$u := [1, 3, 5]$$

$$v := [-2, 3, 0]$$

$$w := [0, -3, 6]$$

$$n := \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

```
> #Question 2
> uv:=crossprod(u,v);dotprod(uv,w);
```

$$uv := [-15, -10, 9]$$

$$84$$

```
> vw:=crossprod(v,w);dotprod(u,vw);
```

$$vw := [18, 12, 6]$$

$$84$$

```
> #Question 3
> uv:=crossprod(u,v);norm(uv,2);evalf(norm(uv,2));
```

$$uv := [-15, -10, 9]$$

$$\sqrt{406}$$

$$20.14944168$$

```
> #Question 4
>
> V:=vector(3,i->b[i]);U:=vector(3,i->a[i]);W:=vector(3,i->c[i]);
```

$$V := [b_1, b_2, b_3]$$

$$U := [a_1, a_2, a_3]$$

$$W := [c_1, c_2, c_3]$$

```
> for i from 1 to 3 do
assume(a[i],real);assume(b[i],real);assume(c[i],real) od;
> UV:=crossprod(U,V);
```

$$UV := [a_2\, b_3 - a_3\, b_2, a_3\, b_1 - a_1\, b_3, a_1\, b_2 - a_2\, b_1]$$

```
> k1:=dotprod(UV,W);
```

$$k1 := (a_2\, b_3 - a_3\, b_2)\, c_1 + (a_3\, b_1 - a_1\, b_3)\, c_2 + (a_1\, b_2 - a_2\, b_1)\, c_3$$

```
> VW:=crossprod(V,W);
```

$$VW := [b_2\, c_3 - b_3\, c_2, b_3\, c_1 - b_1\, c_3, b_1\, c_2 - b_2\, c_1]$$

```
> k2:=dotprod(U,VW);# Le produit mixte et qui re presente à un
signe prés le volume du......
```

$$k2 := a_1\,(b_2\, c_3 - b_3\, c_2) + a_2\,(b_3\, c_1 - b_1\, c_3) + a_3\,(b_1\, c_2 - b_2\, c_1)$$

```
> evalb(k1=k2);#attention
```

false

```
> evalb(expand(k1)=expand(k2));#Ouf!!!
```

true

```
> #Question 5
> A:=stackmatrix(U,V,W);
```

$$A := \begin{bmatrix} a\sim_1 & a\sim_2 & a\sim_3 \\ b\sim_1 & b\sim_2 & b\sim_3 \\ c\sim_1 & c\sim_2 & c\sim_3 \end{bmatrix}$$

```
> det(A);
```

$$c\sim_1\, a\sim_2\, b\sim_3 - c\sim_1\, a\sim_3\, b\sim_2 + c\sim_2\, a\sim_3\, b\sim_1 - c\sim_2\, a\sim_1\, b\sim_3 + c\sim_3\, a\sim_1\, b\sim_2 - c\sim_3\, a\sim_2\, b\sim_1$$

```
> evalb(det(A)=expand(k1));# On retrouve le résultat  bien connu :
déterminant=produit mixte
```

true

```
> #Exercice 3
> restart:with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
> A:=vandermonde([x,y,z]);
```

$$A := \begin{bmatrix} 1 & x & x^2 \\ 1 & y & y^2 \\ 1 & z & z^2 \end{bmatrix}$$

```
> #1°)
> A[2,3];
```

$$y^2$$

```
> #2°)
> swapcol(delrows(delcols(A,2..2),1..1),1,2);
```

$$\begin{bmatrix} y^2 & 1 \\ z^2 & 1 \end{bmatrix}$$

```
> submatrix(swapcol(swapcol(A,1,3),2,3),2..3,1..2);
```

$$\begin{bmatrix} y^2 & 1 \\ z^2 & 1 \end{bmatrix}$$

```
> #3°)
> row(A,2);col(A,3);
```

$$[1, y, y^2]$$

$$[x^2, y^2, z^2]$$

```
> restart:with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
```

```
> #Exercice 4
> A:=vandermonde([x,y,z]);
```

$$A := \begin{bmatrix} 1 & x & x^2 \\ 1 & y & y^2 \\ 1 & z & z^2 \end{bmatrix}$$

```
>
>
> B:=matrix(3,2,[1,2,2,3,3,4]);
```

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

```
> stackmatrix(A,transpose(B));
```

$$\begin{bmatrix} 1 & x & x^2 \\ 1 & y & y^2 \\ 1 & z & z^2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

```
> concat(A,B);
```

$$\begin{bmatrix} 1 & x & x^2 & 1 & 2 \\ 1 & y & y^2 & 2 & 3 \\ 1 & z & z^2 & 3 & 4 \end{bmatrix}$$

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

$$\begin{bmatrix} 1 & x & x^2 & 1 & 2 \\ 1 & y & y^2 & 2 & 3 \\ 1 & z & z^2 & 3 & 4 \end{bmatrix}$$

```
> #exercice5
> restart:with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
```

```
> M:=matrix(5,5,[0, 1, 1, 1, 1,1 ,0 ,1 ,1, 1,1 ,1, 0, 1, 1,1 ,1,
```

```
1, 0, 1,1 ,1 ,1 ,1 ,0]);
```

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

```
> A:=evalm(delcols(M,5..5));
```

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

```
> rank(A);
```

$$4$$

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

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

```
> det(%);
```

$$0$$

```
> AA:=evalm(transpose(A)&*A);
```

$$AA := \begin{bmatrix} 4 & 3 & 3 & 3 \\ 3 & 4 & 3 & 3 \\ 3 & 3 & 4 & 3 \\ 3 & 3 & 3 & 4 \end{bmatrix}$$

```
> det(AA);
```

$$13$$

```
> #Donc AA  est inversible et son inverse est
> inverse(AA);
```

$$\begin{bmatrix} \frac{10}{13} & \frac{-3}{13} & \frac{-3}{13} & \frac{-3}{13} \\ \frac{-3}{13} & \frac{10}{13} & \frac{-3}{13} & \frac{-3}{13} \\ \frac{-3}{13} & \frac{-3}{13} & \frac{10}{13} & \frac{-3}{13} \\ \frac{-3}{13} & \frac{-3}{13} & \frac{-3}{13} & \frac{10}{13} \end{bmatrix}$$

```
> #Exercice 6
> Restart:with(linalg):
> A:=matrix(3,3,(i,j)->a[i,j]);
```

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

```
> AT:=transpose(A);
```

$$AT := \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} \\ a_{1,2} & a_{2,2} & a_{3,2} \\ a_{1,3} & a_{2,3} & a_{3,3} \end{bmatrix}$$

```
> AAT:=evalm(A&*AT);
```

$$AAT := \begin{bmatrix} a_{1,1}^2 + a_{1,2}^2 + a_{1,3}^2, a_{1,1} a_{2,1} + a_{1,2} a_{2,2} + a_{1,3} a_{2,3}, a_{1,1} a_{3,1} + a_{1,2} a_{3,2} + a_{1,3} a_{3,3} \\ a_{1,1} a_{2,1} + a_{1,2} a_{2,2} + a_{1,3} a_{2,3}, a_{2,1}^2 + a_{2,2}^2 + a_{2,3}^2, a_{2,1} a_{3,1} + a_{2,2} a_{3,2} + a_{2,3} a_{3,3} \\ a_{1,1} a_{3,1} + a_{1,2} a_{3,2} + a_{1,3} a_{3,3}, a_{2,1} a_{3,1} + a_{2,2} a_{3,2} + a_{2,3} a_{3,3}, a_{3,1}^2 + a_{3,2}^2 + a_{3,3}^2 \end{bmatrix}$$

```
> TAA:=transpose(AAT);
```

$$TAA := \begin{bmatrix} a_{1,1}^2 + a_{1,2}^2 + a_{1,3}^2, a_{1,1} a_{2,1} + a_{1,2} a_{2,2} + a_{1,3} a_{2,3}, a_{1,1} a_{3,1} + a_{1,2} a_{3,2} + a_{1,3} a_{3,3} \\ a_{1,1} a_{2,1} + a_{1,2} a_{2,2} + a_{1,3} a_{2,3}, a_{2,1}^2 + a_{2,2}^2 + a_{2,3}^2, a_{2,1} a_{3,1} + a_{2,2} a_{3,2} + a_{2,3} a_{3,3} \\ a_{1,1} a_{3,1} + a_{1,2} a_{3,2} + a_{1,3} a_{3,3}, a_{2,1} a_{3,1} + a_{2,2} a_{3,2} + a_{2,3} a_{3,3}, a_{3,1}^2 + a_{3,2}^2 + a_{3,3}^2 \end{bmatrix}$$

```
> equal(AAT,TAA);
```

true

```
> B:=matrix(4,4,[0,1,1,1,0,0,1,1,0,0,0,1,0,0,0,0]);
```

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

```
> evalm(B^5);
```

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

```
> puissance:=proc(N,n::integer)
  local H , i;
  H:=evalm(copy(N)):
  evalm(H^n);
  end;
```

puissance := **proc**(N, n::*integer*) **local** H, i; H := evalm(copy(N)); evalm(H^n) **end proc**

```
> for i from 1 to 4 do puissance(B,i) od;
```

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

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

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

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

```
> #Exercice 7
> Restart:with(linalg):
> S:=matrix(5,5,(i,j)->s[i,j]);
```

$$S := \begin{bmatrix} s_{1,1} & s_{1,2} & s_{1,3} & s_{1,4} & s_{1,5} \\ s_{2,1} & s_{2,2} & s_{2,3} & s_{2,4} & s_{2,5} \\ s_{3,1} & s_{3,2} & s_{3,3} & s_{3,4} & s_{3,5} \\ s_{4,1} & s_{4,2} & s_{4,3} & s_{4,4} & s_{4,5} \\ s_{5,1} & s_{5,2} & s_{5,3} & s_{5,4} & s_{5,5} \end{bmatrix}$$

```
> seq(row(S,i),i=1..5);
```

$$[s_{1,1}, s_{1,2}, s_{1,3}, s_{1,4}, s_{1,5}], [s_{2,1}, s_{2,2}, s_{2,3}, s_{2,4}, s_{2,5}], [s_{3,1}, s_{3,2}, s_{3,3}, s_{3,4}, s_{3,5}],$$
$$[s_{4,1}, s_{4,2}, s_{4,3}, s_{4,4}, s_{4,5}], [s_{5,1}, s_{5,2}, s_{5,3}, s_{5,4}, s_{5,5}]$$

```
> T:=augment(%);
```

$$T := \begin{bmatrix} s_{1,1} & s_{2,1} & s_{3,1} & s_{4,1} & s_{5,1} \\ s_{1,2} & s_{2,2} & s_{3,2} & s_{4,2} & s_{5,2} \\ s_{1,3} & s_{2,3} & s_{3,3} & s_{4,3} & s_{5,3} \\ s_{1,4} & s_{2,4} & s_{3,4} & s_{4,4} & s_{5,4} \\ s_{1,5} & s_{2,5} & s_{3,5} & s_{4,5} & s_{5,5} \end{bmatrix}$$

```
> #row(S,i) retourne sous forme de  (vector) vecteur (donc array
  uniligne) les lignes de la matrice S. col(S,i) retourne sous
  forme de  (vector) vecteur (donc array uniligne) les colonnes de
  la matrice S. Pour les explications : voir le premier tableau et
  l'action des fonctions augment (concat) et stackmatrix sur les
  vecteurs qui est rappelée dans le sujet du TD.
> seq(col(S,i),i=1..5);
```

$$[s_{1,1}, s_{2,1}, s_{3,1}, s_{4,1}, s_{5,1}], [s_{1,2}, s_{2,2}, s_{3,2}, s_{4,2}, s_{5,2}], [s_{1,3}, s_{2,3}, s_{3,3}, s_{4,3}, s_{5,3}],$$
$$[s_{1,4}, s_{2,4}, s_{3,4}, s_{4,4}, s_{5,4}], [s_{1,5}, s_{2,5}, s_{3,5}, s_{4,5}, s_{5,5}]$$

```
> augment(%);
```

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

exercice 8

```
> restart;
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
```

1°)

```
> f:=proc(i,n,x)
  if type(x,numeric) then
   if x<= (i-1)/n then 0;
   elif x<=i/n then n*(x-(i-1)/n)
   elif x<=(i+1)/n then n*((i+1)/n-x);
   else 0;
   fi;
  elif type(x,realcons) then f(i,n,evalf(x));
  else 'f'(i,n,x);
  fi;
   end;
```

$f :=$ **proc**(i, n, x)

if type$(x, numeric)$ **then**

if $x \leq (i-1)/n$ **then** 0

elif $x \leq i/n$ **then** $n*(x-(i-1)/n)$

elif $x \leq (i+1)/n$ **then** $n*((i+1)/n-x)$

else 0

end if

elif type$(x, realcons)$ **then** f$(i, n, \text{evalf}(x))$

else 'f'(i, n, x)

end if

end proc

```
> f(2,3,1/3);
```

$$0$$

```
> f(2,3,sqrt(2)/2);
```

$$0.8786796570$$

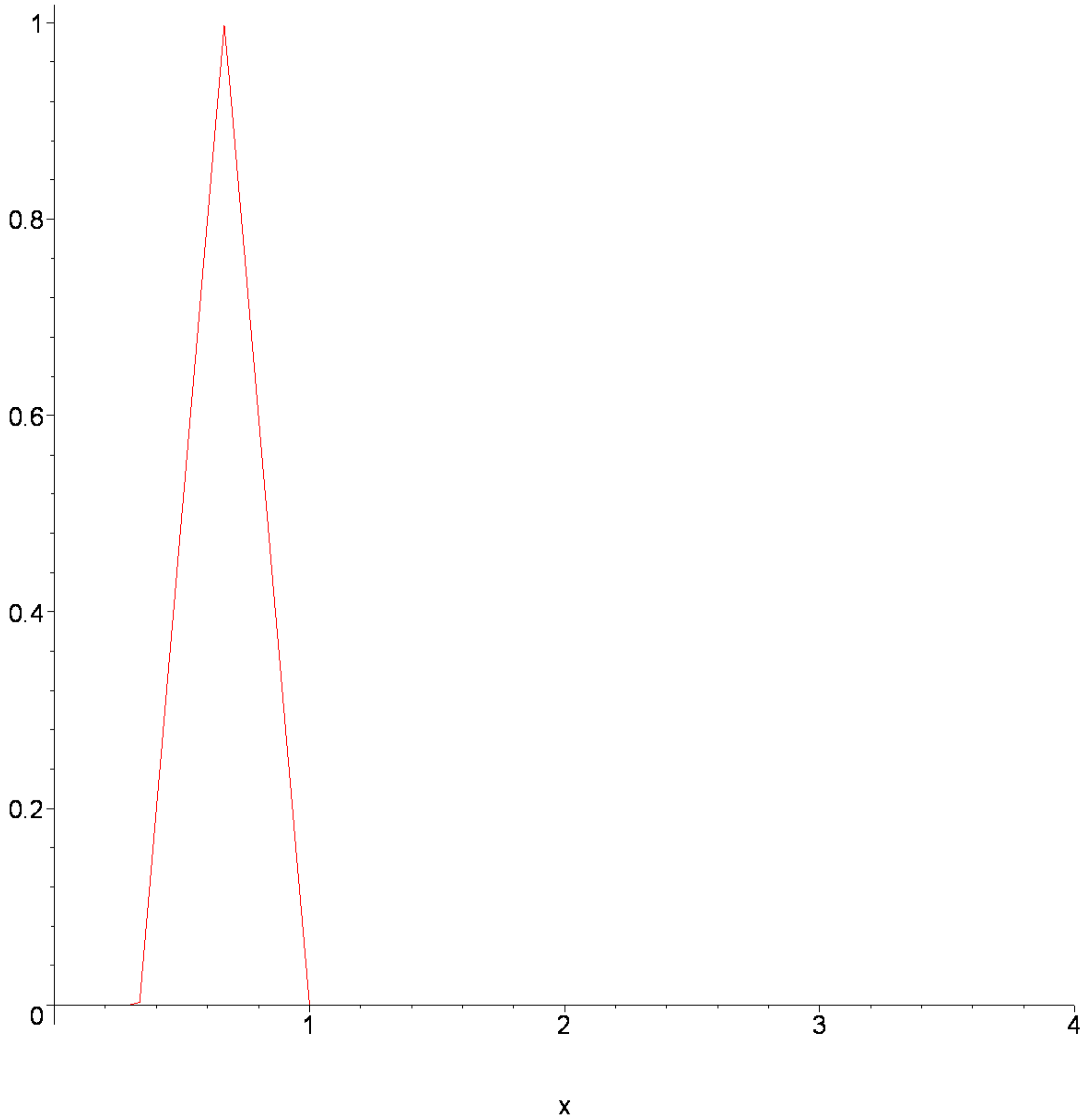
```
> f(2,3,a);
```

$$f(2, 3, a)$$

```
> G:=x->f(2,3,x);
```

$$G := x \to \mathrm{f}(2, 3, x)$$

```
> plot(G(x),x=0..4);
```



```
> G(1/3);G(a);
```

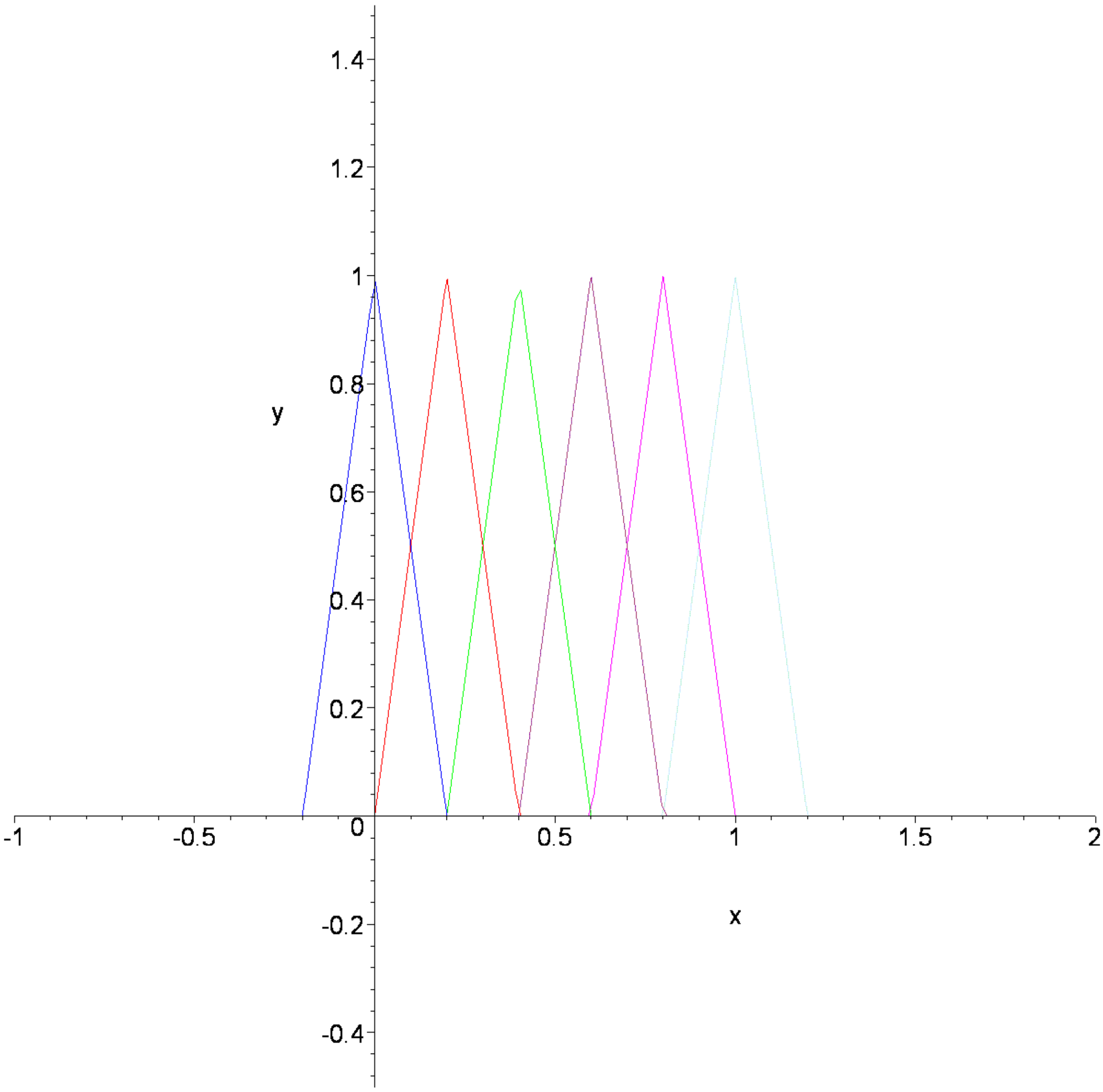
$$0$$

$$\mathrm{f}(2, 3, a)$$

```
> L:=[seq(f(i,5,x),i=0..5)];
```

$$L := [\mathrm{f}(0, 5, x), \mathrm{f}(1, 5, x), \mathrm{f}(2, 5, x), \mathrm{f}(3, 5, x), \mathrm{f}(4, 5, x), \mathrm{f}(5, 5, x)]$$

```
> plot(L,x=-1..2,y=-0.5..1.5,color=[blue,red,green,maroon,ma
  genta,turquoise]);
```



2°)

La matrice identité

```
> In:=proc(n)
  diag(1$n);

> end;
```

$In := \textbf{proc}(n)\ \text{diag}(1\ \$\ n)\ \textbf{end proc}$

```
> In(5);
```

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

```
> #Et pour fair compliquer
> IN:=proc(n::integer)
```

```
local M, i,j;
M:=matrix(n,n);
for i from 1 to n do;
    for j from 1 to n do;
 if i<>j then M[i,j]:=0;
else M[i,j]:=1 fi;
od;
od;
print(evalm(M));
end;
```

$IN := \textbf{proc}(n::integer)$

$\textbf{local}\ M, i, j;$

$M := \text{matrix}(n, n);$

$\textbf{for}\ i\ \textbf{to}\ n\ \textbf{do}$

$\textbf{for}\ j\ \textbf{to}\ n\ \textbf{do if}\ i \neq j\ \textbf{then}\ M[i, j] := 0\ \textbf{else}\ M[i, j] := 1\ \textbf{end if end do}$

$\textbf{end do};$

$\text{print}(\text{evalm}(M))$

$\textbf{end proc}$

```
> IN(1);IN(2);IN(3);
```

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

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

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

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

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

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

La matrice A

```
> A:=proc(n::integer)
local M,i,j;
M:=matrix(n,n);
for i from 1 to n do;
    for j from 1 to n do;
 if i<>j then M[i,j]:=i+j;
else M[i,j]:=0 fi;
od;
od;
```

```
print(evalm(M));
end;
```

$A := \mathbf{proc}(n::integer)$

local $M, i, j;$

$M := \text{matrix}(n, n);$

for i **to** n **do**

for j **to** n **do if** $i \neq j$ **then** $M[i, j] := i + j$ **else** $M[i, j] := 0$ **end if end do**

end do;

$\text{print}(\text{evalm}(M))$

end proc

```
> A(4);
```

$$\begin{bmatrix} 0 & 3 & 4 & 5 \\ 3 & 0 & 5 & 6 \\ 4 & 5 & 0 & 7 \\ 5 & 6 & 7 & 0 \end{bmatrix}$$

```
> A(2);
```

$$\begin{bmatrix} 0 & 3 \\ 3 & 0 \end{bmatrix}$$

```
> A(10);
```

$$\begin{bmatrix} 0 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 3 & 0 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 4 & 5 & 0 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 5 & 6 & 7 & 0 & 9 & 10 & 11 & 12 & 13 & 14 \\ 6 & 7 & 8 & 9 & 0 & 11 & 12 & 13 & 14 & 15 \\ 7 & 8 & 9 & 10 & 11 & 0 & 13 & 14 & 15 & 16 \\ 8 & 9 & 10 & 11 & 12 & 13 & 0 & 15 & 16 & 17 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 0 & 17 & 18 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 0 & 19 \\ 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 0 \end{bmatrix}$$

```
> A(1);
```

$$[\ 0]$$

La matrice B

```
> B:=proc(n::integer)
  local M, i,j;
  M:=matrix(n,n);
  for i from 1 to n do;
      for j from 1 to n do;
   M[i,j]:=(i-j)/(i+j);
  od;
  od;
  print(evalm(M));
```

```
end;
```

$B := \mathbf{proc}(n::integer)$
local $M, i, j;$
$M := \text{matrix}(n, n);$
for i **to** n **do for** j **to** n **do** $M[i, j] := (i - j) / (i + j)$ **end do end do**;
$\text{print}(\text{evalm}(M))$
end proc

```
> B(4);B(2);
```

$$\begin{bmatrix} 0 & \frac{-1}{3} & \frac{-1}{2} & \frac{-3}{5} \\ \frac{1}{3} & 0 & \frac{-1}{5} & \frac{-1}{3} \\ \frac{1}{2} & \frac{1}{5} & 0 & \frac{-1}{7} \\ \frac{3}{5} & \frac{1}{3} & \frac{1}{7} & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & \frac{-1}{3} \\ \frac{1}{3} & 0 \end{bmatrix}$$

La matrice C

```
> C:=proc(n::integer)
  local M,i,j;
  M:=matrix(n,n);
  for i from 1 to n do;
      for j from 1 to n do;
          if j=i then if i=n then M[i,j]:=1;
                      else M[i,j]:=1-(1/2)^(n-i) fi;
          elif j=i-1 then M[i,j]:=(1/2)^(n-i)
          else M[i,j]:=0 fi;
  od;
  od;
  print(evalm(M));
  end;
```

$C := \mathbf{proc}(n::integer)$
local $M, i, j;$
$M := \text{matrix}(n, n);$
for i **to** n **do for** j **to** n **do**
if $j = i$ **then**
if $i = n$ **then** $M[i, j] := 1$
else $M[i, j] := 1 - (1 / 2)^\wedge(n - i)$

```
                end if
            elif j = i - 1 then M[i, j] := (1/2)^(n - i)
            else M[i, j] := 0
            end if
        end do
    end do;
    print(evalm(M))
end proc
```

```
> C(5);
```

$$\begin{bmatrix} \frac{15}{16} & 0 & 0 & 0 & 0 \\ \frac{1}{8} & \frac{7}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

```
> C(2);
```

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

```
> #Exercice 9
> A:=matrix(4,4,(i,j)->a[i,j]);
```

$$A := \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix}$$

```
> B:=swapcol(A,2,4);
```

$$B := \begin{bmatrix} a_{1,1} & a_{1,4} & a_{1,3} & a_{1,2} \\ a_{2,1} & a_{2,4} & a_{2,3} & a_{2,2} \\ a_{3,1} & a_{3,4} & a_{3,3} & a_{3,2} \\ a_{4,1} & a_{4,4} & a_{4,3} & a_{4,2} \end{bmatrix}$$

```
> evalb(expand(det(A))=expand(-det(B)));
```

$$true$$

```
> C:=swaprow(A,1,4);
```

$$C := \begin{bmatrix} a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \end{bmatrix}$$

```
> evalb(expand(det(A))=expand(-det(B)));
```

true

>