Correction du TP 4

B. Landreau

Systemes lineaires

Exercice 1 : resolution de systemes lineaires avec solve ou linsolve

Avec la commande solve

La commande solve permet de resoudre une (ou des) equation(s).

```
> equations1:={2*x+y+z=3,x+y+3*z=4,x-2*y-z=5};
```

$$equations1 := \{2\,x+y+z=3,\ x+y+3\,z=4,\ x-2\,y-z=5\}$$

```
> solve(equations1);
```

$$\{x=\frac{21}{11},\ z=\frac{16}{11},\ y=\frac{-25}{11}\}$$

Si on veut reutiliser l'un ou l'autre coordonnee de la solution, il faut affecter le resultat de solve sur une variable et utiliser ensuite subs

```
>
> S1:=solve(equations1);
```

$$S1 := \{x=\frac{21}{11},\ z=\frac{16}{11},\ y=\frac{-25}{11}\}$$

```
> a:=subs(S1,x);b:=subs(S1,y);c:=subs(S1,z);V:=vector([a,b,c]);
```

$$a := \frac{21}{11}$$

$$b := \frac{-25}{11}$$

$$c := \frac{16}{11}$$

$$V := \left[\frac{21}{11},\ \frac{-25}{11},\ \frac{16}{11}\right]$$

```
> equations2:={x+y+z=3,x+2*y-z=2,2*x+y+4*z=1};S2:=solve(equatio
  ns2);
```

$$equations2 := \{x+y+z=3,\ x+2\,y-z=2,\ 2\,x+y+4\,z=1\}$$

$$S2 :=$$

Dans ce cas, visiblement il n'y a pas de solution.

```
> equations3:={x+y+z=1,x+2*y-z=2,2*x+y+4*z=1};S3:=solve(equatio
  ns3);
```

$$equations3 := \{x+y+z=1,\ x+2\,y-z=2,\ 2\,x+y+4\,z=1\}$$

$$S3 := \{y = 1 + 2\,z,\ x = -3\,z,\ z = z\}$$

Dans ce cas, il y a une infinite de solutions, on a un parametrage de l'ensemble par y.

```
> equations4:={x+y+z=1,2*x+y-z=0};S4:=solve(equations4);
```

$$equations4 := \{x + y + z = 1,\ 2\,x + y - z = 0\}$$

$$S4 := \{y = 2 - 3\,z,\ z = z,\ x = -1 + 2\,z\}$$

Meme cas de figure.

```
>
```

Avec linsolve

On charge d'abord la librairie classique d'algebre lineaire.

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

```
> A1:=matrix(3,3,[2,1,1,1,1,3,1,-2,-1]);
  B1:=vector([3,4,5]);
```

$$A1 := \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 3 \\ 1 & -2 & -1 \end{bmatrix}$$

$$B1 := [3, 4, 5]$$

```
> V:=linsolve(A1,B1);
```

$$V := \left[\frac{21}{11}, \frac{-25}{11}, \frac{16}{11}\right]$$

OK c'est juste!

```
> A2:=matrix(3,3,[1,1,1,1,2,-1,2,1,4]);
  B2:=vector([3,2,1]);
  V2:=linsolve(A2,B2);
```

$$A2 := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & -1 \\ 2 & 1 & 4 \end{bmatrix}$$

$$B2 := [3, 2, 1]$$

$$V2 :=$$

Pas de solutions, il ne retourne rien.

```
> A3:=matrix(3,3,[1,1,1,1,2,-1,2,1,4]);
  B3:=vector([1,2,1]);
  V3:=linsolve(A3,B3);
```

$$A3 := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & -1 \\ 2 & 1 & 4 \end{bmatrix}$$

$$B3 := [1, 2, 1]$$

$$V3 := [-3\,\_t_1,\ 2\,\_t_1 + 1,\ \_t_1]$$

On a un vecteur parametre par le parametre _t1.

```
> A4:=matrix(2,3,[1,1,1,2,1,-1]);
  B4:=vector([1,0]);
  V4:=linsolve(A4,B4);
```

$$A4 := \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$$

$$B4 := [1, 0]$$

$$V4 := [-1 + 2\ \_t_1, 2 - 3\ \_t_1, \_t_1]$$

Encore un vecteur parametre.

Exercice 2 : le pivot de Gauss

On reprend l’algorithme du pivot simple de Gauss.
Attention il existe deja une procedure maple qui s’appelle pivot.

```
> PIVOT:=proc(A,B)
  local i,j,k,T,n;
  n:=coldim(A);
  T:=concat(A,B);
  print(T);
  for i from 1 to n-1 do
    for j from i+1 to n do
      T[j,i]:=-T[j,i]/T[i,i];
      for k from i+1 to n+1 do
        T[j,k]:=T[j,k]+T[j,i]*T[i,k];
      od;
    od;
  print(T);
  od;
  # remontee
  for i from n to 1 by -1 do

  T[i,n+1]:=(T[i,n+1]-sum(T[i,'j']*T['j',n+1],'j'=i+1..n))/T[i,i];
  od;
  print(T);
  RETURN(vector(n,i->T[i,n+1]));
  end:
```

TESTONS.

```
>
> PIVOT(A1,B1);
```

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

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

$$\begin{bmatrix} 2 & 1 & 1 & 3 \\ \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} & \frac{5}{2} \\ \frac{-1}{2} & 5 & 11 & 16 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & \frac{21}{11} \\ \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} & \frac{-25}{11} \\ \frac{-1}{2} & 5 & 11 & \frac{16}{11} \end{bmatrix}$$

$$\left[\frac{21}{11}, \frac{-25}{11}, \frac{16}{11}\right]$$

```
> PIVOT(A2,B2);
```

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

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

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

```
Error, (in PIVOT) numeric exception: division by zero
```

C’est normal, dans ce cas la matrice n’est pas inversible, on tombe sur un pivot nul.
Meme chose pour A3. POur A4, la matrice n’est pas carree, la procedure n’est pas adaptee.

Exercice 3 : un exemple

```
> A:=matrix(3,3,[2,1,1,1,1,3,1,-2,-1]);B:=vector([3.1,4.02,4.98]);
```

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

$$B := [3.1, 4.02, 4.98]$$

```
> W:=PIVOT(A,B);
```

$$\begin{bmatrix} 2 & 1 & 1 & 3.1 \\ 1 & 1 & 3 & 4.02 \\ 1 & -2 & -1 & 4.98 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 3.1 \\ \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} & 2.470000000 \\ \frac{-1}{2} & \frac{-5}{2} & \frac{-3}{2} & 3.430000000 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 3.1 \\ \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} & 2.470000000 \\ \frac{-1}{2} & 5 & 11 & 15.78000000 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 1.949090910 \\ \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} & -2.232727276 \\ \frac{-1}{2} & 5 & 11 & 1.434545455 \end{bmatrix}$$

$$W := [1.949090910, -2.232727276, 1.434545455]$$

A l’exercice 1, on avait obtenue comme solution

```
> V := vector([21/11, -25/11, 16/11]);evalm(1.*V);
```

$$V := \left[\frac{21}{11}, \frac{-25}{11}, \frac{16}{11}\right]$$

$$[1.909090909, -2.272727273, 1.454545454]$$

Evaluons l’erreur:

```
> norm(V-W);
```

$$.040000001$$

```
> cond(A,infinity);
```

$$\frac{60}{11}$$

Calculons les erreurs relatives

```
> EB:=norm(B-B1)/norm(B1);
```

$$EB := .02000000000$$

```
> EV:=norm(V-W)/norm(V);
```

$$EV := .01760000044$$

On sait que d'apres le cours EV<= cond(A) EB. C'est effectivement le cas.

```
> EB*cond(A,infinity);
```

$$.1090909091$$

```
>
```

Exercice 4 : influence du conditionnement

```
> A:=matrix(4,4,[10,7,8,7,7,5,6,5,8,6,10,9,7,5,9,10]);
```

$$A := \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}$$

```
> B:=vector([32,23,33,31]);
```

$$B := [32, 23, 33, 31]$$

```
>
```

Resolution du syteme non perturbe

On utilise la procedure pivot ou on a mis en commentaire (#) l'affichage des matrices intermediaires.

```
> X:=PIVOT(A,B);
```

$$X := [1, 1, 1, 1]$$

Resolution du syteme perturbe sur B

```
> B1:=vector([32.1,22.9,33.1,30.9]);
```

$$B1 := [32.1, 22.9, 33.1, 30.9]$$

```
> X1:=PIVOT(A,B1);
```

$$X1 := [9.200000000, -12.60000000, 4.500000000, -1.100000000]$$

On constate une tres grande variation sur le resultat X. Calculons les erreurs relatives sur B et sur X
pour la norme infinie.

```
> EB:=norm(B1-B)/norm(B);
```

$$EB := .003030303030$$

```
> EX:=norm(X1-X)/norm(X);
```

$$EX := 13.60000000$$

Comparons les erreurs relatives.

```
> EX/EB;
```

$$4488.000000$$

```
> cond(A);
```

$$4488$$

On sait que l'on a toujours EX <= cond(A) * EB. Ici, on est dans un cas ou il y a meme egalite.
Ce qui se passe ici c'est que la matrice est mal conditionnee (cond(A) est grand), donc une petite
erreur sur A ou B va occasionner une grande erreur sur la solution X.
Regardons ce qui se passe pour la norme 1 (par defaut **norm** est la norme infinie)

```
> EB1:=norm(B1-B,1)/norm(B,1);EX1:=norm(X1-X,1)/norm(X,1);EX1/E
  B1;
```

$$EB1 := .003361344538$$

$$EX1 := 6.850000000$$

$$2037.875000$$

```
> cond(A,1);
```

$$4488$$

On a cond(A,1)=cond(A,∞) puisque A est symetrique.
Cette fois l'inegalite est stricte.

Resolution du syteme perturbe sur A

```
> A2:=matrix(4,4,[10,7,8.1,7.2,7.08,5.04,6,5,8,5.98,9.89,9,6.99
  ,4.99,9,9.98]);
```

$$A2 := \begin{bmatrix} 10 & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 5.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{bmatrix}$$

```
> X2:=PIVOT(A2,B);
```

$$X2 := [-81.00000462, 137.0000077, -34.00000202, 22.00000121]$$

Encore une fois la variation sur X est enorme, cela est du au conditionnement. Calculons les erreurs relatives.

```
> EA:=norm(A2-A)/norm(A);
```

$$EA := .009090909091$$

```
> EX:=norm(X2-X)/norm(X);
```

$$EX := 136.0000077$$

```
> EX/EA;
```

$$14960.00085$$

```
> cond(A);
```

$$4488$$

Tiens! La on constate que EX > cond(A)* EA.
Ce n'est pas contradictoire avec le cours puisque l'on a vu en cours que en fait norm(X2-X)/norm(X2) <= cond(A) * norm(A2-A)/norm(A);
Ici, comme X2 est tres different de X, cela change tout.
Reprenons.

```
> EX:=norm(X2-X)/norm(X2);EX/EA;
```

$$EX := .9927007303$$

$$109.1970803$$

Cette fois on a bien EX <= cond(A) * EA

Exercice 5 : de l'influence du pivot

```
>
>
> Digits:=10;
```

$$Digits := 10$$

```
> A:=matrix(2,2,[10.^(-15),1.,1.,1]);B:=([1.,2.]);
```

$$A := \begin{bmatrix} .1000000000\ 10^{-14} & 1. \\ 1. & 1 \end{bmatrix}$$

$$B := [1., 2.]$$

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

$$[1., 1.]$$

La solution parait correcte.

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

$$[0., 1.000000000]$$

La c'est absurde!
Il y a un tout petit pivot au debut et cela va amplifier les erreurs d'arrondis.
Inversons les 2 lignes.

```
> A:=matrix(2,2,[1.,1.,10.^(-15),1.]);B:=([2.,1.]);
```

$$A := \begin{bmatrix} 1. & 1. \\ .1000000000\ 10^{-14} & 1. \end{bmatrix}$$

$$B := [2., 1.]$$

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

$$[1.000000000, 1.000000000]$$

Cette fois, c'est correct! D'ou l'importance du choix du pivot (pivot partiel ou total).

```
>
>
```

FIN