1 restart;maple_mode(1);cas_setup(0,0,0,1,0,1e-10,25,[1,50,0,25],0,0,0);#radians,pas de cmplx, pas de Sqrt

Cas_setup Vector [0,0,0,1,0,1.0000000000e-10,25,[1,50,0,25],0,0,0]
^M

([], Warning: some commands like subs might change arguments order , 0, 0, 0, 1, 0, 1e-10, 25, [1 50 0 25], 0, 0,

2 diag(seq(1,4)); diag(seq(i,i=1..4));

$$\left(\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}, \begin{bmatrix}1&0&0&0\\0&2&0&0\\0&0&3&0\\0&0&0&4\end{bmatrix}\right)$$

3 A:=matrix(4,4)+1;v:=[seq(1,j=1..4)];

$$\left(\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}, \begin{bmatrix}1&1&1&1\end{bmatrix}\right)$$

4 A*v;# Attention il retourne une ligne

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

5 f:=(i,j)->if (i=j) then 0 else if (i<j) then a[i,j] else -a[j,i] fi;fi;

// Warning: a declared as global variable(s)
// End defining f

```
           if i=j then
            0 else
            if i<j then
             a[i,j] else
             -(a[j,i])
            fi
( i, j )-> fi
```

6 matrix(4,4,f); d:=det(matrix(4,4,f)):

$$\left(\begin{bmatrix}0 & a[1,2] & a[1,3] & a[1,4]\\ -(a[1,2]) & 0 & a[2,3] & a[2,4]\\ -(a[1,3]) & -(a[2,3]) & 0 & a[3,4]\\ -(a[1,4]) & -(a[2,4]) & -(a[3,4]) & 0\end{bmatrix}, \text{Done}\right)$$

7 factor(d);#c'est toujours un carre

$$((a[1,2])\cdot(a[3,4]) - (a[1,3])\cdot(a[2,4]) + (a[1,4])\cdot(a[2,3]))^2$$

8 M:=matrix(8,8,f);

$$\begin{bmatrix}
0 & a[1,2] & a[1,3] & a[1,4] & a[1,5] & a[1,6] & a[1,7] & a[1,8]\\
-(a[1,2]) & 0 & a[2,3] & a[2,4] & a[2,5] & a[2,6] & a[2,7] & a[2,8]\\
-(a[1,3]) & -(a[2,3]) & 0 & a[3,4] & a[3,5] & a[3,6] & a[3,7] & a[3,8]\\
-(a[1,4]) & -(a[2,4]) & -(a[3,4]) & 0 & a[4,5] & a[4,6] & a[4,7] & a[4,8]\\
-(a[1,5]) & -(a[2,5]) & -(a[3,5]) & -(a[4,5]) & 0 & a[5,6] & a[5,7] & a[5,8]\\
-(a[1,6]) & -(a[2,6]) & -(a[3,6]) & -(a[4,6]) & -(a[5,6]) & 0 & a[6,7] & a[6,8]\\
-(a[1,7]) & -(a[2,7]) & -(a[3,7]) & -(a[4,7]) & -(a[5,7]) & -(a[6,7]) & 0 & a[7,8]\\
-(a[1,8]) & -(a[2,8]) & -(a[3,8]) & -(a[4,8]) & -(a[5,8]) & -(a[6,8]) & -(a[7,8]) & 0
\end{bmatrix}$$

9 Methode type pivot, contre m'ethode de Laplace,... mais ici la matrice est antisymetrique.

10 d:=det(M):

Evaluation time: 22.01

Done

11 d:=det_minor(M):

Evaluation time: 1.93

Done

12 -------------------Illustration de la reduction de Jordan---
Construction de l'exemple: on veut une reponse de ce type:

13 f:=(i,j)->if(i=j-1) then 1 else 0 fi;

// Success
// End defining f

```
if i=(j-1) then
  1 else
  0
(i, j)-> fi
```

14 J:=matrix(8,8,f);#forme classique d'ordre 8.

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

15 J[3,4]:=0:J[6,7]:=0:J;#2 blocs d'ordre 3 et un d'ordre 2.

$$\left(\text{Done}, \text{Done}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

16 On veut faire un changement de base simple. Ex det=1 pour garder des coeffs entiers.
on cree une transvection:

17 f:=(i,j)->if(i=j) then 1 else 0 fi;

// Success
// End defining f

```
if i=j then
  1 else
  0
(i, j)-> fi
```

18
```
T:=proc(i,j,a)
local A;
A:=matrix(8,8,f):A[i,j]:=a;A;
end proc;
```

// Warning: f declared as global variable(s)
// End defining T

```
proc(i,j,a)
  local A;
  A:=matrix(8,8,f);
  A[i,j]:=a;
  A;

end;
```

19 B:=matrix(8,8,b);

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

20 faire Li<-Li+aLj c'est multiplier a gauche par T(i,j,a).
Par exemple L3<- L3+aL2 c'est multiplier a GAUCHE par: T(3,2,a);

21 T(3,2,a)*B;

$$\begin{bmatrix} b & b & b & b & b & b & b & b \\ b & b & b & b & b & b & b & b \\ b\cdot a+b & b\cdot a+b & b\cdot a+b & b\cdot a+b & b\cdot a+b & b\cdot a+b & b\cdot a+b & b\cdot a+b \\ b & b & b & b & b & b & b & b \\ b & b & b & b & b & b & b & b \\ b & b & b & b & b & b & b & b \\ b & b & b & b & b & b & b & b \\ b & b & b & b & b & b & b & b \end{bmatrix}$$

22 En revanche: C2<-C2+aC3 c'est multplier a DROITE par T(3,2,a)

23 B*T(3,2,a);

$$\begin{bmatrix} b & b\cdot a+b & b & b & b & b & b & b \\ b & b\cdot a+b & b & b & b & b & b & b \\ b & b\cdot a+b & b & b & b & b & b & b \\ b & b\cdot a+b & b & b & b & b & b & b \\ b & b\cdot a+b & b & b & b & b & b & b \\ b & b\cdot a+b & b & b & b & b & b & b \\ b & b\cdot a+b & b & b & b & b & b & b \\ b & b\cdot a+b & b & b & b & b & b & b \end{bmatrix}$$

24 Remarquer que l'inverse de T(i,j,a) est T(i,j,-a)

25 T(3,2,a)^(-1);

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

26 Donc conjuguer par T(i,j,a) c'est faire:
Li <- Li+aLj et Cj <- Cj-aCi

27 P:=T(6,7,2)*T(4,5,1)*T(3,2,2)*T(1,2,1):

Done

28 P,P^(-1);

$$\left(\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

29 Donc faire a l'ordinateur:

30 N:=P*J*P^(-1):

Done

31 est identique a faire a la main a partir de J:
L1 <- L1+L2 ; C2<-C2 - C1 puis
L3 <- L3+2L2 ; C2 <- C2 -2C3
L4 <- L4+L5 ; C5 <- C5 -C4
L6 <- L6 +2 L7; C7 <- C7 -2 C6
On a maintenant trouve un bel exercice: Trouver la forme de jordan de N et une matrice de passage pour l'obtenir.
On calcule N^2 et son noyau.

32
```
N,N^2;
```

$$\left(\begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

33
```
N2:=nullspace(N^2);
```

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

34 on choisit a et b independants modulo ker N^2 (qui est aussi im N). (attention a et b hors de ker N^2 est insuffisant).

35
```
a:=[0,0,1,0,0,0,0,0]:b:=[0,0,0,0,0,1,0,0]:
```

(Done , Done)

36
```
rank(matrix([op(N2),a,b]));# doit etre dim ker N^2 +2.
```

8

37
```
N1:=nullspace(N);
```

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

38 dim ker N^2 -dim ker N= 6-3=2+1 donc N.a,N.b doit etre complete par c tq N.a,N.b,c indep modulo ker N. Par exemple on prend celui la:

39
```
c:=[0,0,0,0,0,0,0,1]:
```

Done

40 On verifie qu'il convient:

41
```
rank(matrix([op(N1),N*a,N*b,c]));
```

6

42 dim ker N - dim ker N^0=3 c'est donc engendr'e par N^2.a,N^2.b,N.c. Il n'y a plus rien a faire, et l'on prend la base suivante: (Attention pour xcas N*a... sont des lignes, on prend donc la transposee)

43 Q:=transpose(matrix([(N^2)*a,N*a,a,(N^2)*b,N*b,b,N*c,c]));

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

Mer

44 On sait maintenant que Q^(-1)*N*Q doit donner J. verifification:

45 Q^(-1)*N*Q;

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

Mer

46 --