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

([], Warning: some commands like subs might change arguments order , 0, 0, 0, 1, 0, 1e-10 , 10, [1 5

2 n:=5;A:=matrix(n,n,(i,j)->a[i,j]);M:=A;c:=1;k:=1;

// Warning: a declared as global variable(s)

(5, [[a[1,1], a[1,2], a[1,3], a[1,4], a[1,5]], [a[2,1], a[2,2], a[2,3], a[2,4], a[2,5]], [a[3,1], a[3,2], a[3,3], a[3,4], a[3,5]], [a[4,1], a[4,2], a[4,3], a[4,4], a[4,5]], [a[5,1], a[5,2], a[5,3], a[5,4], a[5,5]]], [[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] ...], [a[4,1], a[4,2], a[4,3] ...], [a[5,1], a[5,2], a[5,3] ...]]

3 Pour M, On peut ecraser car i,j>k

4
```
Lk:=[];
for i from k+1 to n do
for j from k+1 to n do
M[i,j]:=simplify(1/c*(M[k,k]*M[i,j]-M[i,k]*M[k,j]));
II:=[seq(l,l=1..k),i];JJ:=[seq(l,l=1..k),j];
AIIJJ:=matrix([seq([seq(A[u,v],v=JJ)],u=II)])
Lk:=[op(Lk),AIIJJ];
print(simplify(det(AIIJJ)-M[i,j]));
od:od:
c:=M[k,k]:k:=k+1;
Lk;
```

0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0

([], Done, Done, 2, [[a[1,1], a[1,2]], [a[2,1], a[2,2]]], [[a[1,1], a[1,3]], [a[2,1], a[2,3]]], [[a[1,1], a[1,4]], [a[2,1], a[2,4]]] ...

5
```
Lk:=[];
for i from k+1 to n do
for j from k+1 to n do
M[i,j]:=simplify(1/c*(M[k,k]*M[i,j]-M[i,k]*M[k,j]));
II:=[seq(l,l=1..k),i];JJ:=[seq(l,l=1..k),j];
AIIJJ:=matrix([seq([seq(A[u,v],v=JJ)],u=II)])
Lk:=[op(Lk),AIIJJ];
print(simplify(det(AIIJJ)-M[i,j]));
od:od:
c:=M[k,k]:k:=k+1;
Lk;
```

0
0
0
0
0
0
0
0

([], Done, Done, 3, [[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]]], [[a[1,1], a[1,2], a[1,4]], [a[2,1], a[2,2], a[2,4]], [a[3,1], a[3,2], a[3,4]]] ...

6
```
Lk:=[];
for i from k+1 to n do
for j from k+1 to n do
M[i,j]:=simplify(1/c*(M[k,k]*M[i,j]-M[i,k]*M[k,j]));
II:=[seq(l,l=1..k),i];JJ:=[seq(l,l=1..k),j];
AIIJJ:=matrix([seq([seq(A[u,v],v=JJ)],u=II)])
Lk:=[op(Lk),AIIJJ];
print(simplify(det(AIIJJ)-M[i,j]));
od:od:
c:=M[k,k]:k:=k+1;
Lk;
```

0
0
0
0
Evaluation time: 0.5

([], Done, Done, 4,

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]

a[1, 1]	a[1, 2]	a[
a[2, 1]	a[2, 2]	a[
a[3, 1]	a[3, 2]	a[
a[4, 1]	a[4, 2]	a[

Menu

7
```
Lk:=[];
for i from k+1 to n do
for j from k+1 to n do
M[i,j]:=simplify(1/c*(M[k,k]*M[i,j]-M[i,k]*M[k,j]));
II:=[seq(l,l=1..k),i];JJ:=[seq(l,l=1..k),j];
AIIJJ:=matrix([seq([seq(A[u,v],v=JJ)],u=II)])
print(simplify(det(AIIJJ)-M[i,j]));
od:od:
c:=M[k,k]:k:=k+1;
```

0
Evaluation time: 0.492188

([], Done, Done, 5)

Menu

8 On passe de Mk-1 a Mk par: L_i <-(c_k.L_i-a_{i,k}.L_k)/c_{k-1}. Chacune de ces operatons multiplie le determinant p
on ne touche pas a la premiere ligne de M_{k-1}. Donc le determinant de M_{k} fois le premier coefficient de M_{k-1}
(c_k/c_{k-1})^(dim Mk) fois det M_(k-1). d'ou la formule:det(M_k)=(c^{n-k-1}_k)/(c_{k-1}^{n-k}) *det(M_{k-1})
D'autre part, la matrice M_{n-1} est de taille 1 et vaut c_n. Ce qui donne bien det(M_0)=c_n. Donc c_n est entier. Si l
resultat a A_{I,J}, alors le dernier c sera a^(k)_{i,j}, il vaudra donc det A_{I,J} et sera entier. Ce qui prouve que ces di

9
```
n:=10;A:=matrix(n,n,(i,j)->rand(21)-10);M:=A;c:=1;k:=1;s:=1;
```

// Success

(10,

10	-5	6	1	7	-10	10	2	8	-3
-10	2	0	-7	-6	-6	-9	-2	-3	-8
1	8	-4	8	-6	-10	4	8	-10	8
0	4	3	7	-1	7	3	-7	0	2
4	4	0	-9	1	2	5	-4	10	-10
-7	-4	9	-7	6	0	10	9	-8	-9
5	8	7	7	8	8	-2	2	-8	7
6	-10	4	-7	-10	-9	1	-6	-5	-8
7	-6	10	3	-7	1	-9	-7	-9	-1
1	2	-4	-6	-8	-3	-5	-3	-8	-3

,

10	-5	6	1	7	-10	10	2	8
-10	2	0	-7	-6	-6	-9	-2	-
1	8	-4	8	-6	-10	4	8	-
0	4	3	7	-1	7	3	-7	0
4	4	0	-9	1	2	5	-4	1
-7	-4	9	-7	6	0	10	9	-
5	8	7	7	8	8	-2	2	-
6	-10	4	-7	-10	-9	1	-6	-
7	-6	10	3	-7	1	-9	-7	-
1	2	-4	-6	-8	-3	-5	-3	-

Menu

10
```
for k from 1 to n-1 do
for i from k+1 to n do
for j from k+1 to n do
M[i,j]:=simplify(1/c*(M[k,k]*M[i,j]-M[i,k]*M[k,j]));
od:od:
c:=M[k,k]:k:=k+1;s:=s*c;
od:s;
```

(Done, 850927891320)

11 --

12 A := [[0,-2,1,3],[0,0,0,1],[1,1,0,0],[-3,4,1,0]];

$$\begin{bmatrix} 0 & -2 & 1 & 3 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ -3 & 4 & 1 & 0 \end{bmatrix}$$

Menu

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

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

Menu

14 B*A;

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -2 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ -3 & 4 & 1 & 0 \end{bmatrix}$$

Menu

15
```
T:=proc(i,j,a)
local TT;
TT:=identity(4):TT[i,j]:=a:TT;
end proc;
```

// Success
// End defining T

```
proc(i,j,a)
  local TT;
  TT:=identity(4);
  TT[i,j]:=a;
  TT;

end;
```

Menu

16

undef

Menu

17 T(3,1,1/2)*B*A;

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -2 & 1 & 3 \\ 1 & 1 & 0 & \frac{1}{2} \\ -3 & 4 & 1 & 0 \end{bmatrix}$$

Menu

18 T(4,1,3/2)*T(3,1,1/2)*B*A;

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -2 & 1 & 3 \\ 1 & 1 & 0 & \frac{1}{2} \\ -3 & 4 & 1 & \frac{3}{2} \end{bmatrix}$$

19 F:=matrix(4,4):F[1,1]:=1:F[3,3]:=1:F[2,4]:=1:F[4,2]:=1:F;

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

Menu

20 F*T(4,1,3/2)*T(3,1,1/2)*B*A;#On compose les transpositions

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ -3 & 4 & 1 & \frac{3}{2} \\ 1 & 1 & 0 & \frac{1}{2} \\ 0 & -2 & 1 & 3 \end{bmatrix}$$

Menu

21 LU(A,L,U,P);inv(P)*L*U-A;

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

Menu

22 il faut deplacer les transpositions: On a S.T(i,j,s)=T(i',j',s).S pour i=sigma(i'),j=sigma(j')

23 U:=T(4,3,-2)*F*T(4,1,3/2)*T(3,1,1/2)*B*A;

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ -3 & 4 & 1 & \frac{3}{2} \\ 1 & 1 & 0 & \frac{1}{2} \\ -2 & -4 & 1 & 2 \end{bmatrix}$$

Menu

24 L:=(F*T(3,1,-1/2)*F^(-1))*(F*T(4,1,-3/2)*F^(-1))*T(4,3,2);

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

Menu

25 S:=B*F^(-1);

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

Menu

26 S*L*U-A;

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

Menu

27 --

28 A:=matrix(4,4,(i,j)->rand(21)-10);B:=A-x*identity(4);

// Success

$$\left(\begin{bmatrix}9&4&-3&6\\6&6&2&5\\2&8&10&10\\-5&6&-10&4\end{bmatrix},\begin{bmatrix}9-x&4&-3&6\\6&6-x&2&5\\2&8&10-x&10\\-5&6&-10&4-x\end{bmatrix}\right)$$

29 P:=normal(poly2symb(charpoly(A,x)));

$$x^4+(-29)\cdot x^3+370\cdot x^2+(-2020)\cdot x+4532$$

30 for i from 1 to 3 do B:=normal(pivot(B,i,i)); od;

$-x^7-(-58)\cdot x^6-1376\cdot x^5-(-16954)\cdot x^4-113187\cdot x^3-(-383484)\cdot x^2-525420\cdot x+155520$	0	0
0	$x^6+(-49)\cdot x^5+935\cdot x^4+(-8539)\cdot x^3+36336\cdot x^2+(-56460)\cdot x+17280$	0
0	0	$x^4+(-34)\cdot x^3+395\cdot x^2+(-$
0	0	0

31 simplify(B[4,4]/P);#le dernier terme diagonal est un multiple du poly carateristique

$$x^4-33\cdot x^3+381\cdot x^2-1755\cdot x+2430$$

32 B:=A-x*identity(4);

$$\begin{bmatrix}9-x&4&-3&6\\6&6-x&2&5\\2&8&10-x&10\\-5&6&-10&4-x\end{bmatrix}$$

33 for i from 1 to 3 do B:=normal(pivot(B,5-i,i)); od;

0	0	0	- 1 - 4
0	0	$-25\cdot x^2-(-480)\cdot x-4580$	- 1
0	$1300\cdot x^2+(-24960)\cdot x+238160$	0	156
$-6500\cdot x^2-(-124800)\cdot x-1190800$	0	0	- 1 - 2

34 simplify(B[1,4]/P);#le dernier terme diagonal est un multiple du poly carateristique

-1300

35 pari()#pour charger pari; sinon faire pari_matdet

All PARI functions are now defined with the pari_ prefix.
PARI functions are also defined without prefix except:
abs acos acosh arg asin asinh atan atanh binomial bitand bitor bitxor ceil charpoly concat conj content cos cosh di
Note that p-adic numbers must have O argument quoted e.g. 905/7+O('7^3')
Type ?pari for short help
Inside xcas, try Help->Manuals->PARI for HTML help

36 k:=30;A:=matrix(k,k,(i,j)->rand(21)-10):B:=A-x*identity(k):

// Success

(30, Done, Done)

37 time(matdet(B,1));#par pivot

Evaluation time: 1.46094

1.4609375

38 time(matdet(B,0));#par gauss-bareiss

Evaluation time: 0.492188

0.4921875

39 time(pari_charpoly(A,0));#par fadeev

0.1640625

40 k:=40;A:=matrix(k,k,(i,j)->rand(21)-10):B:=A-x*identity(k):

// Success

(40, Done, Done)

41 pour k>40 par pivot usuel on depasse la taille apr defaut de pari,
donc on arrete.

42 time(matdet(B,0));#par gauss-bareiss

Evaluation time: 2.03906

2.0390625

43 time(pari_charpoly(A,0));#par fadeev

Evaluation time: 0.53125

0.53125

44 k:=50;A:=matrix(k,k,(i,j)->rand(21)-10):B:=A-x*identity(k):

// Success

(50, Done, Done)

45 time(matdet(B,0));#par gauss-bareiss

Evaluation time: 6.375

6.375

46 time(pari_charpoly(A,0));#par fadeev

Evaluation time: 1.35938

1.359375