```
    |\^/|     Maple 9 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2003
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
>  interface(screenwidth=120);
> with(linalg):with(LinearAlgebra):
>  n:=5:l:=seq(1.*cond(Matrix(n,n,rand(20)-10.),2),i=1..100):
bytes used=4000832, alloc=3407248, time=0.15
bytes used=8001608, alloc=4979824, time=0.26
bytes used=12001972, alloc=5176396, time=0.38
bytes used=16002324, alloc=5307444, time=0.48
bytes used=20002544, alloc=5307444, time=0.60
bytes used=24005192, alloc=5372968, time=0.71
>  with(stats[statplots]);
 [boxplot, histogram, scatterplot, xscale, xshift, xyexchange, xzexchange, yscale, yshift, yzexchange, zscale, zshift]

>  histogram([l],area=count);

            *GGGGGGGGG
            *         G
            *         G
         80 *         G
            *         G
            *         G
            *         G
         60 *         G
            *         G
            *         G
            *         G
         40 *         G
            *         G
            *         G
            *         G
            *         G
         20 *         G
            *         G
            *         G
            *         GGGGGGGGG
   +--+--+--**************************************************************************************************--+--+---
                       100           200           300           400           500           600
>  max(l);# le max de la suite
                                                635.5535941

>  histogram([l],area=count,numbars=floor(max(l)/100));

            *GGGGGGGGGGGGGG
            *              G
            *              G
         80 *              G
            *              G
            *              G
            *              G
         60 *              G
            *              G
            *              G
            *              G
            *              G
         40 *              G
            *              G
            *              G
            *              G
            *              G
         20 *              G
            *              G
            *              G
            *              G
   +--+--+--**************************************************************************************************--+--+---
                       100           200           300           400           500           600
>  for i  from 1 to 200 do A:= Matrix(n,n,(rand(20)-10.));
>  if cond(A,2)>500 then B:=A fi;od:B;
bytes used=28008436, alloc=5372968, time=0.83
bytes used=32012608, alloc=5635064, time=0.94
bytes used=36015532, alloc=5635064, time=1.06
bytes used=40017980, alloc=5635064, time=1.18
bytes used=44019468, alloc=5700588, time=1.29
bytes used=48020448, alloc=5700588, time=1.40
bytes used=52021916, alloc=5700588, time=1.51
bytes used=56022424, alloc=5700588, time=1.62
bytes used=60023272, alloc=5700588, time=1.73
bytes used=64023436, alloc=5700588, time=1.84
bytes used=68023640, alloc=5700588, time=1.96
bytes used=72023952, alloc=5700588, time=2.07
                                      [-6.    7.     -2.     6.     -8.]
                                      [                                 ]
                                      [-8.    -1.    -9.     1.     7. ]
                                      [                                 ]
                                      [-3.    8.     6.      5.     -3.]
                                      [                                 ]
                                      [-1.    3.     3.     -4.     8. ]
                                      [                                 ]
                                      [-4.    7.     -9.    -10.    -4.]

>  #Par exemple j'avais trouve le B suivant:
>  B:=Matrix([[-7., -8., 1., 0., -5., 6.], [-9., 0., 8., 2., 8., 0.], [7., 7., 1., 5., -4., 6.], [-4., -2., -6., 5., -8.
, -1.], [2., 2., -7., 2., -4., -10.], [6., 4., -2., -1., 1., -4.]]);
                                      [-7.    -8.    1.     0.     -5.     6. ]
                                      [                                       ]
                                      [-9.    0.     8.     2.     8.      0. ]
                                      [                                       ]
                                      [7.     7.     1.     5.     -4.     6. ]
                                B := [                                         ]
                                      [-4.    -2.    -6.    5.     -8.    -1. ]
                                      [                                       ]
                                      [2.     2.     -7.    2.     -4.    -10.]
                                      [                                       ]
                                      [6.     4.     -2.    -1.    1.     -4. ]

>  cond(B);eigenvectors(B);
                                                26823.91305

[0.02734089691, 1, {[9.857238578, -14.49906791, -0.6378662399, 14.21474185, 8.124022056, -0.910731683]}],

    [-18.20524649, 1, {[-0.4895009462, -0.7116777592, 0.3858122414, 0.1915483033, 0.6351457402, 0.4302455267]}],

    5.353510673 + 1.941791812 I, 1, {[0.1894918574 - 0.3427189460 I, -0.2540123768 + 0.7712375837 I,

    0.08420568662 + 0.5198831242 I, -0.1796078303 - 0.4012630680 I, -0.1833054883 - 0.3506774947 I,

    -0.004408933693 + 0.0051335989 I]}], [5.353510673 - 1.941791812 I, 1, {[0.1894918574 + 0.3427189460 I,

    -0.2540123768 - 0.7712375837 I, 0.08420568662 - 0.5198831242 I, -0.1796078303 + 0.4012630680 I,

    -0.1833054883 + 0.3506774947 I, -0.004408933693 - 0.0051335989 I]}], [-0.7645579109 + 1.920922581 I, 1, {[

    -1.619219882 - 1.550141631 I, 2.167726834 + 2.316641008 I, -0.2177600659 - 0.2604287061 I,

    -2.321782619 - 2.138362003 I, -1.786846744 - 0.6497864834 I, 0.2510818670 + 0.4614016401 I]}], [

    -0.7645579109 - 1.920922581 I, 1, {[-1.619219882 + 1.550141631 I, 2.167726834 - 2.316641008 I,

    -0.2177600659 + 0.2604287061 I, -2.321782619 + 2.138362003 I, -1.786846744 + 0.6497864834 I,

    0.2510818670 - 0.4614016401 I]}]

>  a:=eigenvectors(B)[6];b:=eigenvectors(B)[4];
bytes used=76024244, alloc=5700588, time=2.20
a := [-0.7645579109 - 1.920922581 I, 1, {[-1.619219882 + 1.550141631 I, 2.167726834 - 2.316641008 I,

    -0.2177600659 + 0.2604287061 I, -2.321782619 + 2.138362003 I, -1.786846744 + 0.6497864834 I,

    0.2510818670 - 0.4614016401 I]}]

b := [5.353510673 - 1.941791812 I, 1, {[0.1894918574 + 0.3427189460 I, -0.2540123768 - 0.7712375837 I,

    0.08420568662 - 0.5198831242 I, -0.1796078303 + 0.4012630680 I, -0.1833054883 + 0.3506774947 I,

    -0.004408933693 - 0.0051335989 I]}]

>  linsolve(B,0.1*a[3][1]);linsolve(B,100*b[3][1]);linsolve(B,100*b[3][1]+0.1*a[3][1]);
[-0.04069982137 - 0.1004931541 I, 0.06533476457 + 0.1388526563 I, -0.007808493319 - 0.01444412000 I,

    -0.05456753000 - 0.1425870083 I, 0.002759662200 - 0.09192193888 I, 0.01624403052 + 0.01953629348 I]

[1.076015071 + 6.792020734 I, 0.4246722164 - 14.25213038 I, 4.502844049 - 8.077824257 I, -5.367458844 + 5.5484380

    -5.125617093 + 4.691268282 I, -0.0420434251 - 0.111139605 I]

[1.035314649 + 6.691532398 I, 0.4900078637 - 14.11328480 I, 4.495035593 - 8.092268680 I, -5.422027240 + 5.4058579

    -5.122857928 + 4.599350330 I, -0.0257993382 - 0.091603755 I]

#NB: les [1] servent {\'a} retirer les {}
>  A := <<0,-2,1,3>|<0,0,0,1>|<1,1,0,0>|<-3,4,1,0>>;
                                                      [ 0    0    1    -3]
                                                      [                   ]
                                                      [-2    0    1     4]
                                                A := [                    ]
                                                      [ 1    0    0     1]
                                                      [                   ]
                                                      [ 3    1    0     0]

>  B:=Matrix(4):B[1,2]:=1:B[2,1]:=1:B[3,3]:=1:B[4,4]:=1:B;
                                                     [0    1    0    0]
                                                     [                 ]
                                                     [1    0    0    0]
                                                     [                 ]
                                                     [0    0    1    0]
                                                     [                 ]
                                                     [0    0    0    1]

>  B.A;
                                                    [-2    0    1     4]
                                                    [                   ]
                                                    [ 0    0    1    -3]
                                                    [                   ]
                                                    [ 1    0    0     1]
                                                    [                   ]
                                                    [ 3    1    0     0]

>  T:=proc(i,j,a)
>  TT:=Matrix(4)+1:TT[i,j]:=a:TT;
>  end proc;
                        T := proc(i, j, a) local TT; TT := Matrix(4) + 1; TT[i, j] := a; TT end proc

>  T(3,1,1/2).B.A;
                                                    [-2    0     1      4]
                                                    [                    ]
                                                    [ 0    0     1     -3]
                                                    [                    ]
                                                    [ 0    0    1/2     3]
                                                    [                    ]
                                                    [ 3    1     0      0]
```

```
>  T(4,1,3/2).T(3,1,1/2).B.A;
```

$$\begin{bmatrix} -2 & 0 & 1 & 4 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 1/2 & 3 \\ 0 & 1 & 3/2 & 6 \end{bmatrix}$$

```
>  F:=Matrix(4):F[1,1]:=1:F[3,3]:=1:F[2,4]:=1:F[4,2]:=1:F;
```

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

```
>  #On compose les transpositions
>  F.T(4,1,3/2).T(3,1,1/2).B.A;
```

$$\begin{bmatrix} -2 & 0 & 1 & 4 \\ 0 & 1 & 3/2 & 6 \\ 0 & 0 & 1/2 & 3 \\ 0 & 0 & 1 & -3 \end{bmatrix}$$

```
>  L:=LUDecomposition(A);L[1].L[2].L[3]-A;
```

$$L := \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3/2 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix}, \begin{bmatrix} -2 & 0 & 1 & 4 \\ 0 & 1 & 3/2 & 6 \\ 0 & 0 & 1/2 & 3 \\ 0 & 0 & 0 & -9 \end{bmatrix}$$

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

```
>  # il faut deplacer les transpositions:
#{\'e}galit{\'e} pour i=\sigma(i'),j=\sigma(j')
> U:=T(4,3,-2).F.T(4,1,3/2).T(3,1,1/2).B.A;
```

$$U := \begin{bmatrix} -2 & 0 & 1 & 4 \\ 0 & 1 & 3/2 & 6 \\ 0 & 0 & 1/2 & 3 \\ 0 & 0 & 0 & -9 \end{bmatrix}$$

```
> L:=(F.T(3,1,-1/2).F^(-1)).(F.T(4,1,-3/2).F^(-1)).T(4,3,2);
bytes used=80024408, alloc=5700588, time=2.34
```

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

```
> S:=B.F^(-1);
```

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

```
> S.L.U-A;
```

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

```
>  Digits:=2; #Pb avec MatrixInverse, il n'en tient pas compte! utiliser linalg(inverse)
```

$$Digits := 2$$

```
>  a:=1./3;a*3;
```

$$a := 0.33$$

$$0.99$$

```
>  r:=rand(-10..10);
r := proc()
local t;
global _seed;
    _seed := irem(a*_seed, p);
    t := _seed;
    to concats do _seed := irem(a*_seed, p); t := s*t + _seed end do;
    irem(t, divisor) + offset
end proc

>  A:=matrix(4,4,r*1.);
```

$$A := \begin{bmatrix} -7. & 5. & -5. & -2. \\ 2. & 7. & 2. & 10. \\ -4. & 4. & 0. & 10. \\ 10. & -8. & 7. & -5. \end{bmatrix}$$

```
>  #selon la doc de linalg, en taille <=4, il n'utilise pas LU.
>  inverse(A),evalm(inverse(A).A);
```

$$\begin{bmatrix} -0.33 & 0.11 & -0.30 & -0.27 \\ 0.26 & 0.15 & -0.020 & 0.14 \\ 0.60 & -0. & 0.40 & 0.59 \\ -0.24 & -0.020 & -0. & -0.16 \end{bmatrix}, \begin{bmatrix} 1.0 & 0.2 & -0.1 & 0.2 \\ 0. & 1.1 & -0.02 & 0.08 \\ 0.1 & -0.1 & 1.1 & -0.2 \\ 0.1 & 0. & 0.1 & 1.1 \end{bmatrix}$$

```
# Non, la mutiplication des flottants est commutative mais pas associative.
> quit
bytes used=80854616, alloc=5700588, time=2.38
```