```
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
2 n:=5;M:=matrix(n,n,(i,j)->rand(21)-10.0);#NB:rand(21) repond entre 0 et 20
// Success
```

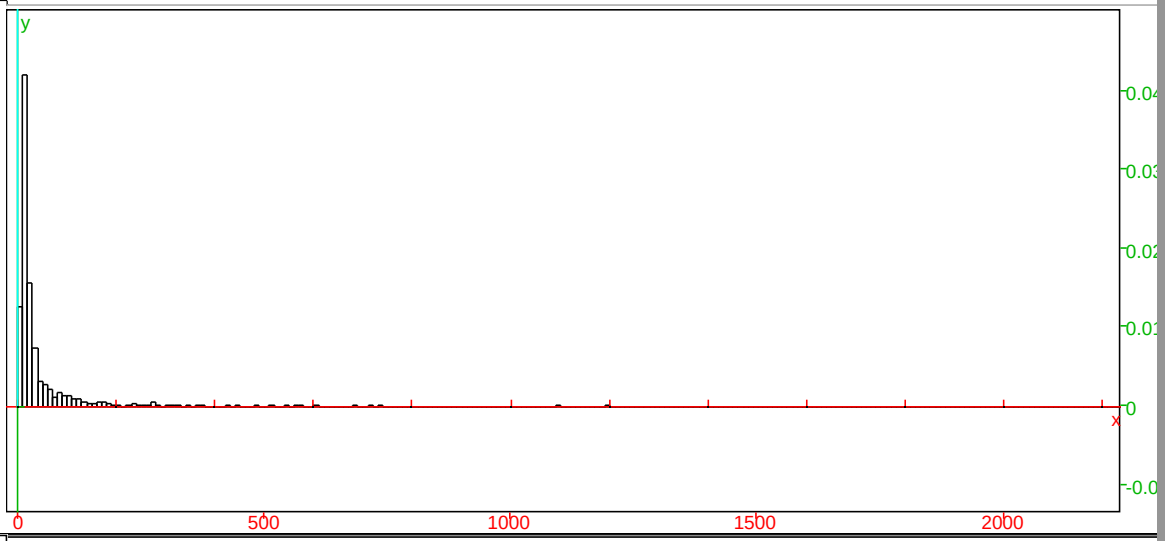
$$\left(5,\ \begin{bmatrix} 10.0 & 5.0 & -3.0 & -6.0 & 0.0 \\ 7.0 & 4.0 & 8.0 & 3.0 & 7.0 \\ -8.0 & -4.0 & 7.0 & 4.0 & 2.0 \\ -8.0 & -2.0 & 1.0 & -5.0 & 1.0 \\ -2.0 & -9.0 & -4.0 & 2.0 & 6.0 \end{bmatrix}\right)$$

```
3 maxnorm(M);colnorm(M);norm(M);
                    ( 10.0,  35.0,  27.23967694375 2
4 cond:=M->[norm(evalf(M))*norm(1/evalf(M)),M];
// Success
// End defining cond
                          begin
                            'nop';
                            [norm(evalf(M))*norm(1/(evalf(M))),M];

                    M  ->  end
5 cond(M);
```

$$\left[23.961725779448,\ \begin{bmatrix} 10.0 & 5.0 & -3.0 & -6.0 & 0.0 \\ 7.0 & 4.0 & 8.0 & 3.0 & 7.0 \\ -8.0 & -4.0 & 7.0 & 4.0 & 2.0 \\ -8.0 & -2.0 & 1.0 & -5.0 & 1.0 \\ -2.0 & -9.0 & -4.0 & 2.0 & 6.0 \end{bmatrix}\right]$$

```
6 n:=5:l:=[seq(cond(matrix(n,n,(i,j)->rand(21)-10.0)),k=1..1000)]:
// Success
Evaluation time: 1.69
                          ( Done ,  Done  )
7 premiere methode: On prend la premiere ligne de la transpos'ee. Seconde methode avec une suite indexee par la liste
8 listecondi:=transpose(l)[1];
[24.710033422221 16.168287663799 24449652949 20.235628355718 10.029293344592 12.164692679241 21.591237577351 14.164168568437 2446
9 listecondi:=[seq(k[1],k=l)];
[24.710033422221 16.168287663799 24449652949 20.235628355718 10.029293344592 12.164692679241 21.591237577351 14.164168568437 2446
10 histogram(classes(listecondi,0,10));
```




```
11 ml:=max(listecondi);# le max de la suite
                          4414.0944344859
12 ecart_type(listecondi);#ou bien: stddev donne l'ecart type
```

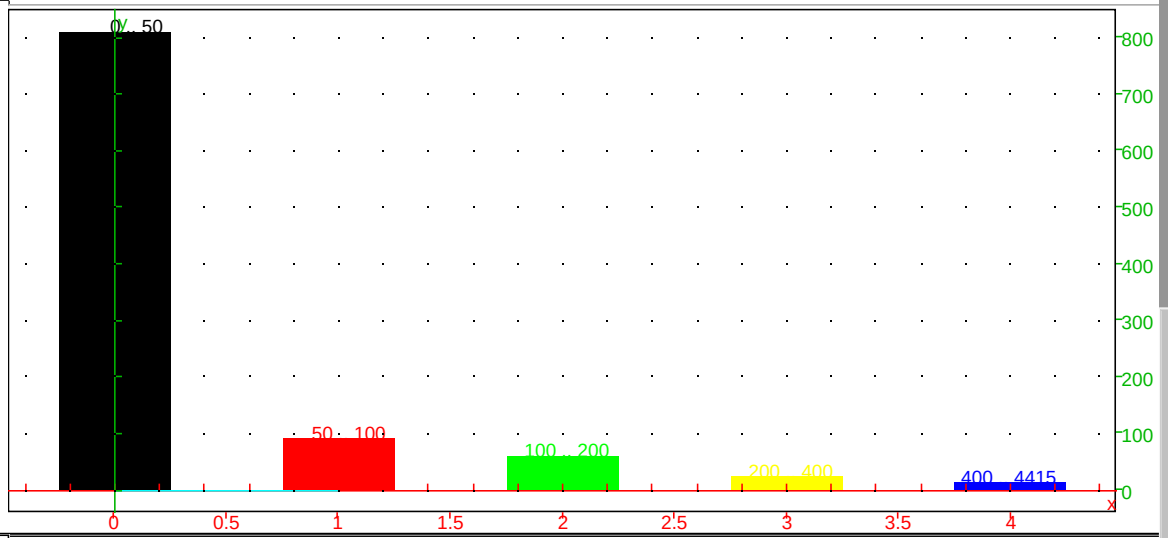
13 moyenne(listecondi);

49.625869937614

14 classes(listecondi,[0..50,50..100,100..200,200..400,400..floor(ml+1)]);#l'infini ne marche pas, on prend donc une borne strictement supe

0 .. 50	809
50 .. 100	92
100 .. 200	60
200 .. 400	25
400 .. 4415	14

15 diagramme_batons(classes(listecondi,[0..50,50..100,100..200,200..400,400..floor(ml+1)]));



16 L'ecart type est enorme par rapport a la moyenne. On constate
souvent 80 pourcent des condtionnements en dessous de 50, alors que
parfois ca peut depasser 1000

17 listetriee:=sort(l,(x,y)->x[1]>=y[1]);

// Success

Done

18 listetriee[1];M:=listetriee[1][2];

$$\left(4414.0944344859,\ \begin{bmatrix} 6.0 & -7.0 & 7.0 & 10.0 & -6.0 \\ -7.0 & -2.0 & 3.0 & 9.0 & -9.0 \\ -9.0 & -7.0 & 0.0 & -3.0 & 2.0 \\ 5.0 & -5.0 & -4.0 & -9.0 & 4.0 \\ 9.0 & -10.0 & -10.0 & -9.0 & -10.0 \end{bmatrix},\ \begin{bmatrix} 6.0 & -7.0 & 7.0 & 10.0 & -6.0 \\ -7.0 & -2.0 & 3.0 & 9.0 & -9.0 \\ -9.0 & -7.0 & 0.0 & -3.0 & 2.0 \\ 5.0 & -5.0 & -4.0 & -9.0 & 4.0 \\ 9.0 & -10.0 & -10.0 & -9.0 & -10.0 \end{bmatrix}\right)$$

19 complex_mode:=1;det(M);eigenvalues(M);

$$\left(1,\ -484.0,\ \begin{bmatrix} 10.279620340638 & 0 & 0 & 0 \\ 0 & 0.017223534135993 & 0 & 0 \\ 0 & 0 & -17.085154802655 & 0 \\ 0 & 0 & 0 & -4.1058445360596-11.964307848269*I \\ 0 & 0 & 0 & -4.1058445360596+11.964307 \end{bmatrix}\right)$$

20 jordan(M);#bon c'etait pas la peine!

$$\begin{bmatrix} 10.279620340638 & 0 & 0 & 0 \\ 0 & 0.017223534135993 & 0 & 0 \\ 0 & 0 & -17.085154802655 & 0 \\ 0 & 0 & 0 & -4.1058445360596-11.964307848269*I \\ 0 & 0 & 0 & -4.1058445360596+11.964307843269*I \end{bmatrix}$$

21 Les cas anormaux on une valeur du determinant tout a fait normale,
la matrice est donc bien inversible, en revanche, on constate une
valeur propre (eventuellement complexe) proche de 0, elle n'est donc
pas loin d'une matrice non inversible.

```
22
23 Digits:=4;
                    [0  0  1  1  0  [1e-10   1e-15 ]  4  [1  50  0  25 ]  0  0  0 ]
24 a:=1-(0.1)^10;
                                  0.9999999999
25 a-1;#il a travaille en fait avec plus de chiffres.
                                -1.0000000827404e-10
26 Digits:=14;
                    [0  0  1  1  0  [1e-10   1e-15 ]  14  [1  50  0  25 ]  0  0  0 ]
27 a:=1-(0.1)^15;
                                          1
28 a-1;#il n'a toujours pas perdu le 0.1
                                -9.9920072216264e-16
29 Digits:=15;
                    [0  0  1  1  0  [1e-10   1e-15 ]  15  [1  50  0  25 ]  0  0  0 ]
30 a:=1-(0.1)^16;#la il a vraiment travaille en 15 chiffres.
                                 1.000000000000000
31 a-1;
                                 0.0000000000000000
32 Digits:=15;
                    [0  0  1  1  0  [1e-10   1e-15 ]  15  [1  50  0  25 ]  0  0  0 ]
33 on s'assure d'avoir une precision exacte, car sous xcas  en 32
   bits, moins de 14 chiffres fait la meme chose que 14 chiffres.
34 v:=[seq(rand(-10.0,10.0),i=1..n)];
   [-0.6846539943875882   8.641035146089905   -8.866944375834677   2.198911985251669   -2.965035573387198 ]
35 b:=M*v;
          [-86.884447326958   7.3852025687791   -66.85216717567   -42.811018359863   5.9573542145664 ]
36 purge(x);
                              No such variable x
37 X:=[seq(x[i],i=1..n)];
                              [x[1]  x[2]  x[3]  x[4]  x[5] ]
38 linsolve(M*X=b,X)-v;
[-1.5543122344752e-14   1.2612133559742e-13   5.2580162446247e-13   -4.3209880118411e-13   -2.7355895326764e-13 ]
39 P:=eigenvectors(M);eigenvalues(M);
   63433.581799643    137.00298260688    1016.946216378    17819.734500686-2964.0116148644*I    17819.734500686+296
   -44564.490493047   -1081.5145626437    1893.3634789561   17906.820788367-9326.0170064984*I    17906.820788367+932
   -29068.381216441   -4476.6090636783    981.96245182837   -2284.9860858545-23032.949174338*I   -2284.9860858545+23(
   43415.52178355     3701.9689149527     -209.31424052684  -15601.588228988+2550.8621650559*I   -15601.588228988-255
   45192.722637079    2345.6143103963     2500.568539316    -13635.452743396+18801.174190721*I   -13635.452743396-188
40 v:=(transpose(P)[2]);
          [137.00298260688   -1081.5145626437   -4476.6090636783   3701.9689149527   2345.6143103963 ]
41 b:=M*v;
            [2.359675547661   -18.627502988267   -77.103029021759   63.76098797707   40.399768144984 ]
42 NB une erreur de 10^-11 alors que l'on travaille avec 15 chiffres donne tout de
   meme un rapport 1000.
43 linsolve(M*X=b,X)-v;
[-4.433786671143e-12   3.6152414395474e-11   1.4824763638899e-10   -1.2323653209023e-10   -7.8216544352472e-11 ]
44
```