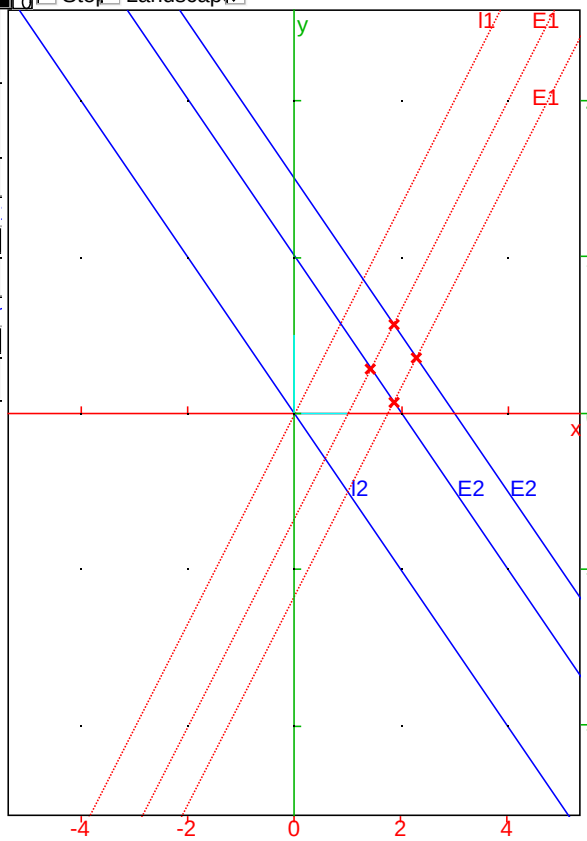
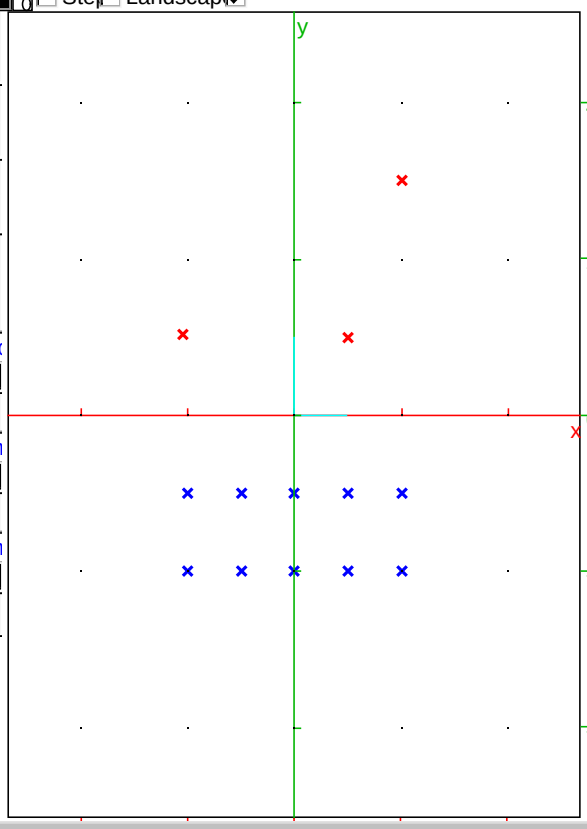
```
1 art;maple_mode(0);cas_setup(0,0,0,1,0,1e-10,10,[1,50,0,25],0,0,0);//radians,pas de cmplx, pas de  Sqrt
2
3 Fig  Edit  Graph  Frame  Mode  Step  Landscape
1 uleur'=(rouge+point_width_5+dot_line));
  droite(y=((4*x)/3))
2 l2 := droite(0,1-i,'couleur'=bleu);
  droite(y=(-x))
3 E1:=[l1+1,l1+1-i]; E2:=[l2+2,l2+3];
  [pnt(pnt[line[1,4+4*i],10485761,"E1"]),pnt(pnt[line[
4 inter(E1,E2,couleur=(rouge+point_width_2));
  [point(10/7,4/7),point(13/7,8/7),point(13/7,1/7),poir
5
```




```
4 Fig  Edit  Graph  Frame  Mode  Step  Landscape
1 _cache); //on sauve les options dans af
  'couleur'=-2146959359
2 l:=[point(1+i,af),point([2,3],af)];
  [point(1,1),point(2,3)]
3 A:=point(-2.098,1.039,'couleur'=(rouge+point_wic
  point(-2.098,1.039)
4 af:=objet->couleur(objet,bleu+point_width_2+non
  // Success
  // End defining af
  (objet)->couleur(objet,bleu+epaisseur_point_2+nc
5 seq(af(point(ii,-1)),ii=-2..2);
  [point(-2,-1),point(-1,-1),point(0,-1),point(1,-1),poin
6 af(seq(point(ii,-2),ii=-2..2));
  [point(-2,-2),point(-1,-2),point(0,-2),point(1,-2),poin
7
```



5 ----------Exercice----------
Famille generatrice de I comme Z module: les generateurs et leurs multiples par Isqrt(5)

6 M:=transpose(matrix([[2,0],[3,3],[0,2],[-15,3]]));

$$\begin{bmatrix} 2, & 3, & 0, & -15 \\ 0, & 3, & 2, & 3 \end{bmatrix}$$

7 ismith(M);//Donc N(I)=2

$$\left(\begin{bmatrix} 1, & 0 \\ 3, & 1 \end{bmatrix}, \begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & 2, & 0, & 0 \end{bmatrix}, \begin{bmatrix} -7, & 0, & 21, & 15 \\ 0, & 0, & 1, & 0 \\ 0, & 1, & -6, & -3 \\ -1, & 0, & 3, & 2 \end{bmatrix}\right)$$

8 Zl1:=[seq(couleur(droite(-10,10)+i*j*sqrt(5),red+dash_line),j=-10..10)]:;//les lignes du reseau.

Done

9 couleur(hidden_name); //On le met en global.

-2147483648

10 Zc1:=[seq(couleur(droite(-10*i,10*i)+j,red+dash_line),j=-10..10)]:; //les colonnes du reseau.

Done

11 Pour le reseau associe a I on cherche d'abord une base de I comme Z module, soit avec ismith soit de tete par ope

12 M:=matrix([[2,1],[0,1]]); //est une base de I

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

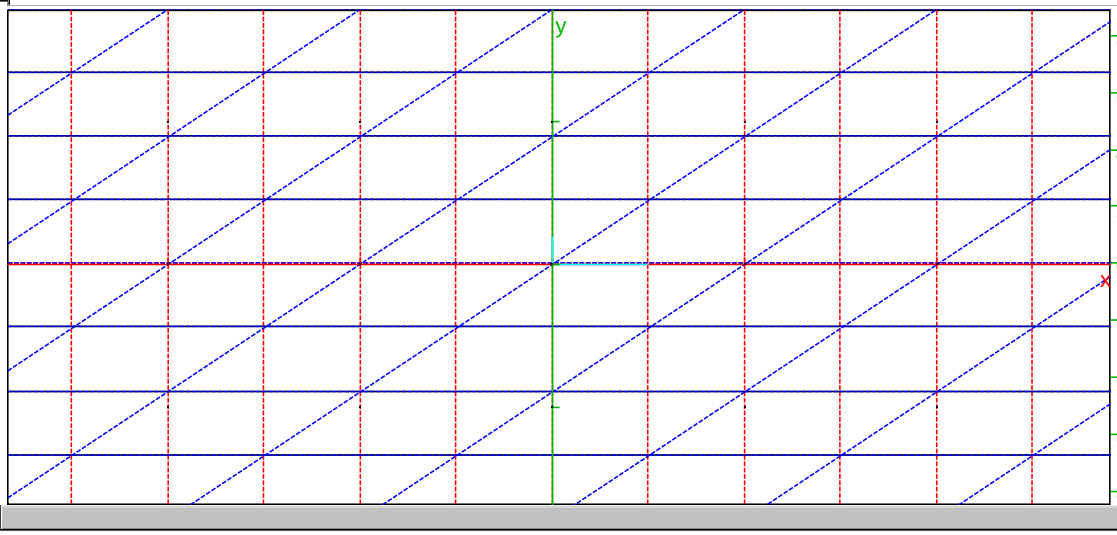
13 Zl2:=[seq(couleur(droite(2*j-10-10*sqrt(5)*i,2*j+10+10*sqrt(5)*i),blue+dash_line),j=-10..10)]:;

Done

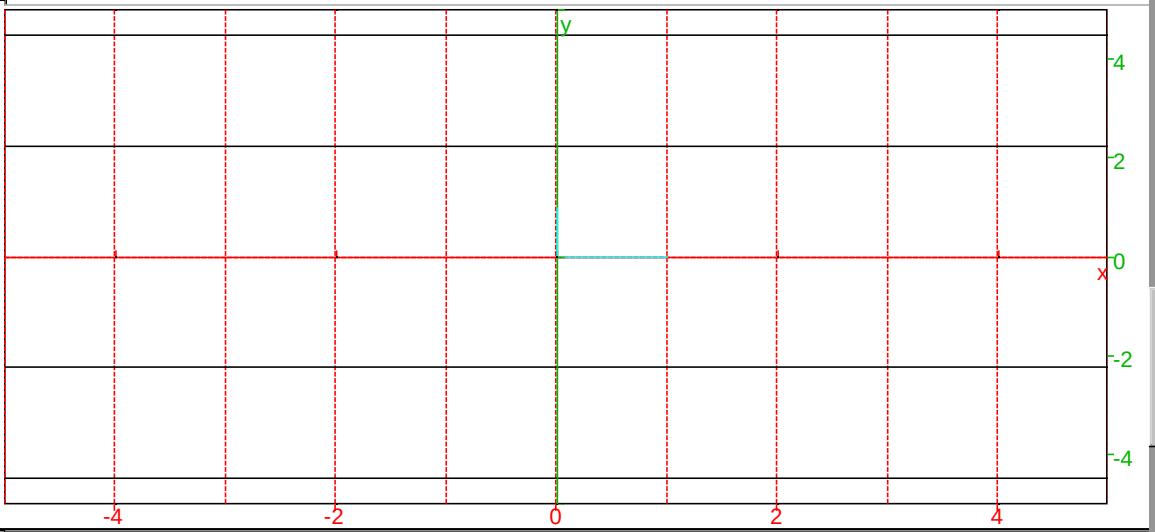
14 Zc2:=[seq(couleur(droite(-10+j+j*sqrt(5)*i,10+j+j*sqrt(5)*i),blue+dash_line),j=-10..10)]:;

Done

15 evalf(Zc1,Zl1,Zc2,Zl2);

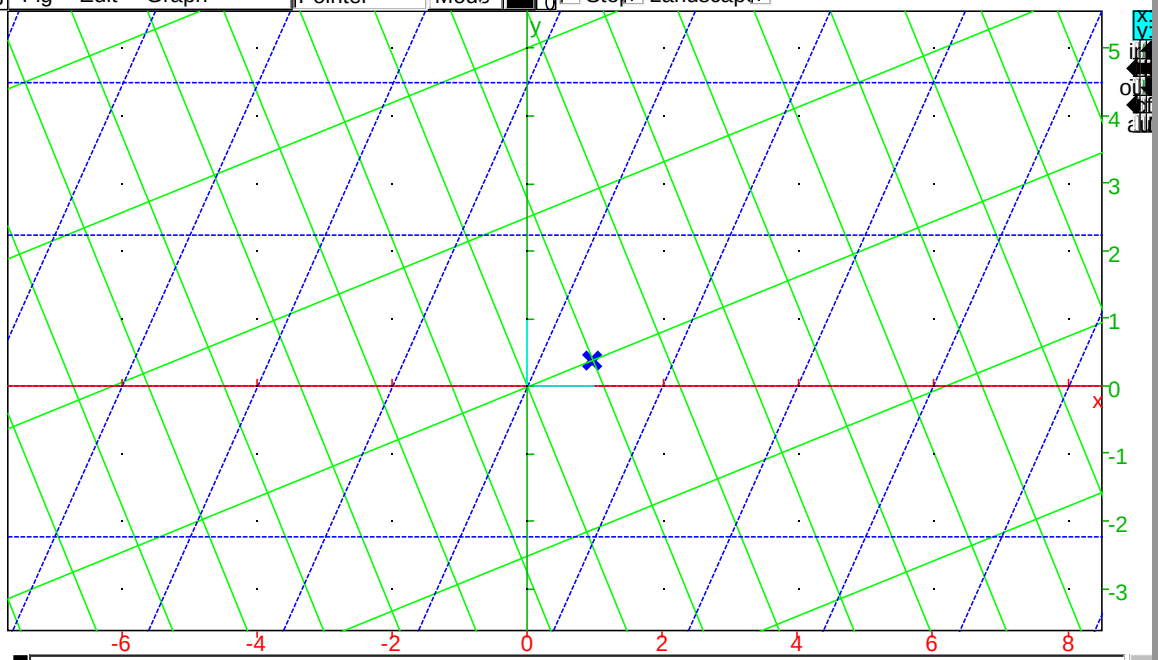


16 Zisq5:=evalf(Zc1,Zl1)



17 Pour illustrer que I n'est pas principal, on l'affiche en bleu et on affiche en vert les multiples du r\'eseau Z(i.sqrt(5)). Dans le menu mode on peut selectionner le mode pointeur, et en deplacant le point a on obtient tous les multiples de Z(i.sqrt(5)) par l'affixe de a. (similitude). Les 2 reseaux ne coincident jamais.

18 Fig Edit Graph Pointer Mode Step Landscape



1 a:=point(1+i,'affichage'=(bleu+epaisseur_point_4))-0.038-0.6087*i

point(0.962,0.3913)

2 aZisq5:=affixe(a)*[Zisq5]:;

"Done"

3 couleur(aZisq5,vert);

Done

4 (Zc2,Zl2);

Done

5

6

19 Attention, le fait d'avoir dessiner des mailles est tendencieux, ca n'est pas par qu'un maillage n'est pas orthogonal qu'il n'en existe pas un autre orthogonal. Pour conclure correctement que I n'est pas principal, il faut dire qu'on ne voit pas de rectangle d'aire 2 semblable au rectangle de cot\'e 1 et sqrt(5)

20
```
gcd(6,3);//Attention, a priori pour lui c'est des polynomes, on a dela chance il les normalise bien.
```

$$3$$

21
```
gcd(x*(x+3),2*x);
```

$$x$$

22
```
igcd(4,6,8);
```

$$2$$

23
```
iquo(13,6);igcd(6,0);
```

$$(2,\ 6)$$

24
```
[a,b,d]:=igcdex(4,15);//pour un couple de bezout:
```

$$[4,\ -1,\ 1]$$

25
```
4*a+15*b-d;
```

$$0$$

26
```
delrows(matrix([[1,1],[2,3]]),0..0); //pour une colonne: delcols
```

$$[2,\ 3]$$

27
```
A:=diag([3,6,18,36]);B:=matrix(4,4,(ii,jj)->rand(7)-3);det(B);
```

// Success

$$\left(\begin{bmatrix} 3, & 0, & 0, & 0 \\ 0, & 6, & 0, & 0 \\ 0, & 0, & 18, & 0 \\ 0, & 0, & 0, & 36 \end{bmatrix}, \begin{bmatrix} -1, & -2, & 1, & -3 \\ 0, & 3, & -1, & 1 \\ 2, & 3, & 0, & 3 \\ -1, & 3, & 0, & 3 \end{bmatrix}, -27\right)$$

28
```
ismith(A*B);
```

$$\left(\begin{bmatrix} -1, & 0, & 0, & 0 \\ 2, & 1, & 0, & 0 \\ -12, & -6, & -1, & 0 \\ -108, & -54, & -10, & -1 \end{bmatrix}, \begin{bmatrix} 3, & 0, & 0, & 0 \\ 0, & 6, & 0, & 0 \\ 0, & 0, & 18, & 0 \\ 0, & 0, & 0, & 972 \end{bmatrix}, \begin{bmatrix} 0, & -1, & -2, & 9 \\ 0, & 0, & 0, & -1 \\ -1, & -1, & 1, & -8 \\ 0, & 0, & 1, & -5 \end{bmatrix}\right)$$

29 ----------

30 Prog Edit Add | 1 | nxt | OK (F9) | Save

```
minval:=proc(A)
local m,u,v,i0,j0;
m:=0;u:=-1;v:=-1;
// NB: On retourne -1,-1 si A est nulle
for(i0:=0; i0<dim(A)[0];i0++){
for(j0:=0; j0<dim(A)[1];j0++){
if ((abs(A[i0,j0])>0) and ((m=0) or (abs(A[i0,j0])<m))) then m:=abs(A[i0,j0]);u:=i0;v:=
};
};
u,v;end proc;
```

```
(A)->
{ local m,u,v,i0,j0;
  m:=0;
  u:=-1;
  v:=-1;
  for (i0:=0;i0<((dim(A))[0]);i0++) {
     for (j0:=0;j0<((dim(A))[1]);j0++) {
        if ((((abs(A[i0,j0]))>0) && (((m=0) || ((abs(A[i0,j0]))<m))))) {
           m:=abs(A[i0,j0]);
           u:=i0;
           v:=j0;
        };
```

31 A:=matrix([[6, 0, 7, 9], [6, 8, 6, 9], [3, 2, 4, 6], [3, 2, 3, 3]]);

$$\begin{bmatrix} 6, & 0, & 7, & 9 \\ 6, & 8, & 6, & 9 \\ 3, & 2, & 4, & 6 \\ 3, & 2, & 3, & 3 \end{bmatrix}$$

M

32 minval(A);

(2, 1)

M

33 (i0,j0):=minval(A);

(2, 1)

M

34 U:=identity(4);

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

M

35 normalement il faudrait faire bouger l dans {1..4} sauf j, c'est plus simple de
ne pas mettre de test, et de corriger U[j,j] ensuite

36

undef

M

37 for l from 0 to 3 do U[j0,l]:=-iquo(A[i0,l],A[i0,j0]) od:;U[j0,j0]:=1:;U;

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

M

38 A*U;

$$\begin{bmatrix} 6, & 0, & 7, & 9 \\ -2, & 8, & -10, & -15 \\ 1, & 2, & 0, & 0 \\ 1, & 2, & -1, & -3 \end{bmatrix}$$

M

39

40 Prog Edit Add | 1 | nxt | | OK (F9) | Save |

```
trans:=proc(A,i0,j0)
local n,U,V,l;
n:=dim(A)[0];
U:=identity(n);
V:=identity(n);
for l from 0 to n-1 do
V[l,i0]:=-iquo(A[l,j0],A[i0,j0]);
od;
//on corrige
V[i0,i0]:=1;
for l from 0 to n-1 do
U[j0,l]:=-iquo(A[i0,l],A[i0,j0]);
od;
//on corrige
U[j0,j0]:=1;
V*A*U;
end proc;
```

(A,i0,j0)->

```
n:=(dim(A))[0];
U:=identity(n);
V:=identity(n);
for (l:=0;l<=(n-1);l:=l+abs(1)) {
   V[l,i0]:=-(iquo(A[l,j0],A[i0,j0]));
```

41 A:=matrix([[6, 1, 7, 9], [6, 8, 6, 9], [3, 1, 4, 6], [3, 2, 1, 3]]);

$$\begin{bmatrix} 6, & 1, & 7, & 9 \\ 6, & 8, & 6, & 9 \\ 3, & 1, & 4, & 6 \\ 3, & 2, & 1, & 3 \end{bmatrix}$$

42 trans(A,3,1);

$$\begin{bmatrix} 5, & 1, & 7, & 8 \\ -6, & 0, & 2, & -3 \\ 2, & 1, & 4, & 5 \\ 1, & 2, & 1, & 1 \end{bmatrix}$$

43

44 Prog Edit Add | 1 | nxt | OK (F9) | Save

```
Zequiv:=proc(A)
local k,n,l,B,i0,j0;
k:=0;
n:=dim(A)[1];
l:=[seq(0,k=1..n)];
B:=A;
(i0,j0):=minval(B);
//minval retourne les coordonnees du minimum non nul,
//ou un couple impossible (Ex ici (-1,-1) en mode xcas) si B est nulle.
while(([i0,j0]<>[-1,-1])and size(B)>1)
{
// attention, ne pas oublier de declarer les variables locales
// dans trans pour  eviter les melanges
B:=trans(B,i0,j0);
if [i0,j0]=[minval(B)]then
// on n'a que des zeros sur la ligne i et la colonne j sauf
// en (i,j) on sauve donc B[i0,j0] et on raye ligne et colonne.
l[k]:=B[i0,j0]; k:=k+1;
B:=delcols(B,j0..j0);
B:=delrows(B,i0..i0);
//le cas B de taille 1 pose probleme car on ne peut pas rayer
//une colonne puis une ligne
fi;
// ASTUCE: voici comment assigner 2 valeurs d'un coup.
(i0,j0):=minval(B);
};
// On a eventuellement change le signe du determinant
diag([seq(l[k],k=0..n-2),B[0,0]]);
end proc;
```

```
(A)->
{ local k,n,l,B,i0,j0;
  k:=0;
  n:=(dim(A))[1];
  l:=[seq(0,k=(1 .. n))];
  B:=A;
  i0,j0:=minval(B);
  while((((i0,j0])!=[-1,-1]) && ((size(B))>1))){
     B:=trans(B,i0,j0);
     if ((([i0,j0])==[minval(B)])) {
        l[k]:=B[i0,j0];
        k:=k+1;
```

45 A:=matrix([[2,2,2],[6,12,6],[6,4,6]]); //Exemple:

$$\begin{bmatrix} 2, & 2, & 2 \\ 6, & 12, & 6 \\ 6, & 4, & 6 \end{bmatrix}$$

46 Zequiv(A);

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

47 A[2,2]:=12:;A;

$$\left(\text{Done}, \begin{bmatrix} 2, & 2, & 2 \\ 6, & 12, & 6 \\ 6, & 4, & 12 \end{bmatrix} \right)$$

48 Zequiv(A);

$$\begin{bmatrix} 2, & 0, & 0 \\ 0, & -2, & 0 \\ 0, & 0, & 18 \end{bmatrix}$$

49 On v\'erifie que A et Zequiv(A) ont bien meme forme de smith:

50 ismith(A)[1], ismith(Zequiv(A))[1];

$$\left(\begin{bmatrix} 2, & 0, & 0 \\ 0, & 2, & 0 \\ 0, & 0, & 18 \end{bmatrix}, \begin{bmatrix} 2, & 0, & 0 \\ 0, & 2, & 0 \\ 0, & 0, & 18 \end{bmatrix} \right)$$

51 Attention, on n'obtient pas forcement les diviseurs elementaires, par exemple:

52 A:=matrix([[4,0,0],[0,6,0],[0,0,8]]);

$$\begin{bmatrix} 4, & 0, & 0 \\ 0, & 6, & 0 \\ 0, & 0, & 8 \end{bmatrix}$$

53 ismith(A)[1];

$$\begin{bmatrix} 2, & 0, & 0 \\ 0, & 4, & 0 \\ 0, & 0, & 24 \end{bmatrix}$$

54 Zequiv(A);

$$\begin{bmatrix} 4, & 0, & 0 \\ 0, & 6, & 0 \\ 0, & 0, & 8 \end{bmatrix}$$

55 ca sera forc\'ement le pgcd(d_1,...,d_n) car l'algorithme reste sur la premiere ligne.

56 f:=(ii,jj)->if (ii-jj)*(ii-1)=0 then 1 else 0 fi;

```
// Success
// End defining f
```

```
if ((((ii-jj)*(ii-1))==0)) {
  1;
}
else {
  0;
( ii, jj )-> };
```

57 matrix(3,3,f);A:=matrix(3,3,f)*A;Zequiv(A);

$$\left(\begin{bmatrix} 1, & 0, & 0 \\ 1, & 1, & 1 \end{bmatrix}, \begin{bmatrix} 4, & 0, & 0 \\ 4, & 6, & 8 \end{bmatrix}, \begin{bmatrix} 4, & 0, & 0 \\ 0, & 2, & 0 \end{bmatrix} \right)$$

58 On recopie trans et Zequiv pour qu'ils ne travaillent que sur les colonnes

59 Prog Edit Add | 1 | nxt | OK (F9) | Save

```
transC:=proc(A,i0,j0)
//attention, si on ne met pas un local n, il modifiera la valeur de n dans ZequivC
local n,U,l;
n:=dim(A)[0];
U:=identity(n);
for l from 0 to n-1 do
U[j0,l]:=-iquo(A[i0,l],A[i0,j0]);
od;
//pour ne pas mettre de test dans la boucle precedente, on n'ecarte pas l=i, mais
```

```
// Success
// End defining transC
```

```
(A,i0,j0)->
{ local n,U,l;
  n:=(dim(A))[0];
  U:=identity(n);
  for (l:=0;l<=(n-1);l:=l+abs(1)) {
     U[j0,l]:=-(iquo(A[i0,l],A[i0,j0]));
   };
  U[j0,j0]:=1;
  A*U;
}
```

60

61 Prog Edit Add | 1 | nxt | OK (F9) | Save

```
ZequivC:=proc(A)
k:=0;
n:=dim(A)[0];
l:=[seq(0,k=1..n)];
B:=A;
(i0,j0):=minval(B);
while(([i0,j0]<>[-1,-1])and size(B)>1)
{
// attention, ne pas oublier de declarer les variables locales dans trans pour  evit
B:=transC(B,i0,j0);
if [i0,j0]=[minval(B)]then
l[k]:=B[i0,j0]; k:=k+1;
B:=delcols(B,j0..j0);
B:=delrows(B,i0..i0);
fi;
// ASTUCE: voici comment assigner 2 valeurs d'un coup.
(i0,j0):=minval(B);
};
// On a eventuellement change le signe du determinant
diag([seq(l[k],k=0..n-2),B[0,0]]);
end proc;
```

```
(A)->
{ local NULL;
  k:=0;
  n:=(dim(A))[0];
  l:=[seq(0,k=(1 .. n))];
  B:=A;
  i0,j0:=minval(B);
  while((((i0,j0])!=[-1,-1]) && ((size(B))>1))){
     B:=transC(B,i0,j0);
     if ((((i0,j0])==[minval(B)])) {
        l[k]:=B[i0,j0];
        k:=k+1;
        B:=delcols(B,j0..j0);
```

62

63 Prog Edit Add | 1 | nxt | OK (F9) | Save

```
elem:=proc(A)
n:=dim(A)[1];
d:=Zequiv(A);
L:=[];
for(i0:=0;i0< n-1;i0++)
{
  T:=matrix(n-i0,n-i0,f);
d:=ZequivC(T*d);
L:=[op(L),d[0,0]];
d:=delrows(delcols(d,0..0),0..0);
};
[op(L),d[0,0]];
end proc;
```

```
(A)->
{ local NULL;
  n:=(dim(A))[1];
  d:=Zequiv(A);
  L:=[];
  for (i0:=0;i0<(n-1);i0++) {
     T:=matrix(n-i0,n-i0,f);
     d:=ZequivC(T*d);
     L:=[op(L),d[0,0]];
     d:=delrows(delcols(d,0 .. 0),0 .. 0);
```

M

64 A:=matrix([[2,2,2],[6,12,6],[6,4,12]]);

$$\begin{bmatrix} 2, & 2, & 2 \\ 6, & 12, & 6 \\ 6, & 4, & 12 \end{bmatrix}$$

M

65 elem(A);ismith(A)[1];

$$\left([2, -2, 18], \begin{bmatrix} 2, & 0, & 0 \\ 0, & 2, & 0 \\ 0, & 0, & 18 \end{bmatrix}\right)$$

M

66 elem(diag(4,6,16));ismith(diag(4,6,16))[1];

$$\left([4, 2, 48], \begin{bmatrix} 2, & 0, & 0 \\ 0, & 4, & 0 \\ 0, & 0, & 48 \end{bmatrix}\right)$$

M

67 ----------Exercice----------

68 cardi:=A->if det(A)=0 then afficher("cardinal infini") else abs(det(A)) fi;

// Success
// End defining cardi

```
A  ->  if (((det(A))==0)) {
          print("cardinal infini");
        }
        else {
          abs(det(A));
       };
```

M

69 A:=matrix([[2*4,4,15*4,0],[8,12,18,36],[16,16,32,32],[32,32,32,32]]);

$$\begin{bmatrix} 8, & 4, & 60, & 0 \\ 8, & 12, & 18, & 36 \\ 16, & 16, & 32, & 32 \end{bmatrix}$$

70 B:=ismith(A)[2];cardi(A);

$$\left(\begin{bmatrix} 16, & -96, & -21603, & 64170 \\ 0, & 1, & 224, & -666 \\ 1, & -6, & -1350, & 4010 \\ -4, & 24, & 5401, & -16043 \end{bmatrix}, 86016\right)$$

M

71 M/N=M/image(A) il est donc isomorphe a M/B ou B=ismith(A) le
cardinal est donc infini s'il y a un 0 sur la diagnonale de B,
sinon c'est le produit des termes diagonaux de B, donc aussi |detB|=|det A| (dans le cas d'une matrice carree)

72 abs(det(B)),abs(det(A)); //dans les 2 cas, c'est le cardinal.

$$(1, 86016)$$

M

73 A:=matrix([[2*4,4,15*4,0],[8,12,18,36],[16,16,32,32],[24,28,50,68]]);

$$\begin{bmatrix} 8, & 4, & 60, & 0 \\ 8, & 12, & 18, & 36 \\ 16, & 16, & 32, & 32 \\ 24, & 28, & 50, & 68 \end{bmatrix}$$

M

74 ismith(A)[2]; cardi(A);

cardinal infini

$$\left(\begin{bmatrix} 16, & -96, & 39, & -72 \\ 0, & 1, & -8, & 114 \\ 1, & -6, & 2, & 2 \\ -4, & 24, & -7, & -23 \end{bmatrix}, 1\right)$$

M

75 A:=matrix([[2*4,4,15*4,0,16],[8,12,18,36,28],[16,16,32,32,32],[24,28,50,68,68]]);

$$\begin{bmatrix} 8, & 4, & 60, & 0, & 16 \\ 8, & 12, & 18, & 36, & 28 \\ 16, & 16, & 32, & 32, & 32 \\ 24, & 28, & 50, & 68, & 68 \end{bmatrix}$$

M

76 ismith(A)[2]; cardi(A);

cardinal infini

$$\left(\begin{bmatrix} 12, & -72, & 7947, & -6161, & 72 \\ 0, & 1, & -110, & 46, & -114 \\ 1, & -6, & 662, & -516, & -2 \\ 0, & 0, & 0, & 8, & 23 \\ -4, & 24, & -2649, & 2062, & 0 \end{bmatrix}, 1\right)$$

M

77 v1:=[1,2];v2:=[-1,1];v3:=[0,2];

$$([1, 2], [-1, 1], [0, 2])$$

M

78 On fait quelques combinaisons lineaires au hasard, ca serait mieux de trouver une base (on le fera plus dans l'exer

79 seq(seq(couleur(point(ii*v1+j*v2+k*v3),blue+point_width_1+hidden_name),k=-5..5),j=-5..5),ii=-5..5):;

Done

M

80 ismith(transpose(matrix([v1,v2,v3]))); //le reseau engendre est bien Z^2

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

81 v1:=[1,i]*v1;v2:=[1,i]*v2;v3:=[1,i]*v3; //notation complexe plus pratique pour les dessins.

$$(1+2*i\ ,\ -1+i\ ,\ 2*i\ )$$

82 d1:= [seq(droite(0,v1)+ii*v2,ii=-10..10)]:;

Done

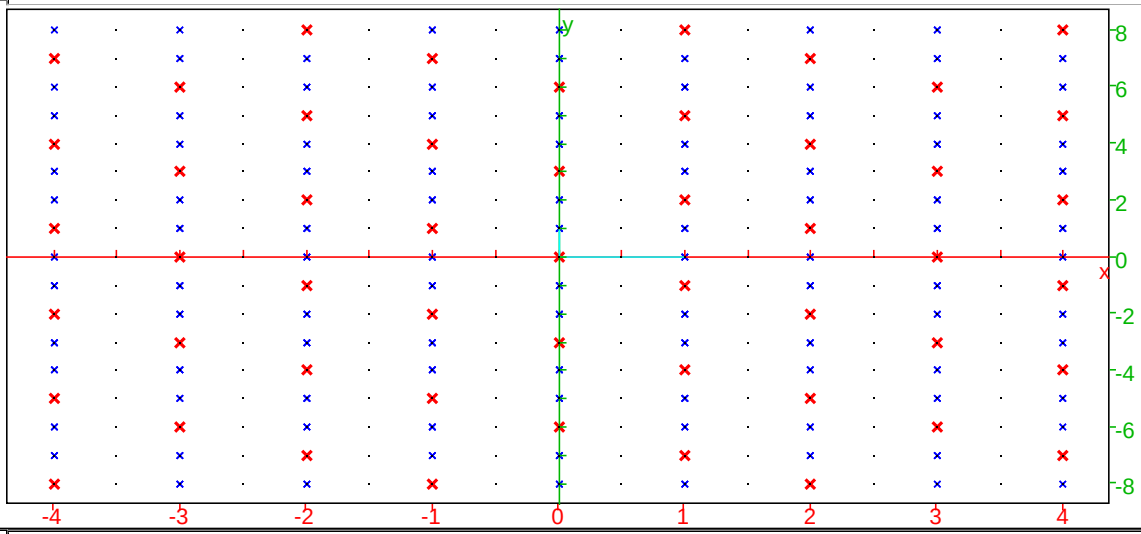
83 d2:= [seq(droite(0,v2)+ii*v1,ii=-10..10)]:;

Done

84 l2:= couleur(inter(d1,d2),red+point_width_2+hidden_name):;

Done

85 (l,l2);// ca ne rempli pas tout



86 aucun des 3 determinants (v1,v2) (v1,v3) (v2,v3) n'est inversible dans Z, on ne peut donc pas en extraire une base.

87 ----------Exercice----------

88 A:=matrix([[2*4,4,15*4,0],[8,12,18,36],[16,16,32,32],[32,32,32,32]]);

$$\begin{bmatrix} 8, & 4, & 60, & 0 \\ 8, & 12, & 18, & 36 \\ 16, & 16, & 32, & 32 \\ 32, & 32, & 32, & 32 \end{bmatrix}$$

89 (U,B,V):=ismith(A);

$$\left(\begin{bmatrix} 0, & 1, & 0, & 0 \\ -57, & -50, & 0, & 26 \\ -12, & -24, & 43, & -11 \\ -184, & -352, & 608, & -149 \end{bmatrix}, \begin{bmatrix} 2, & 0, & 0, & 0 \\ 0, & 4, & 0, & 0 \\ 0, & 0, & 16, & 0 \\ 0, & 0, & 0, & 672 \end{bmatrix}, \begin{bmatrix} 16, & -96, & -21603, & 64170 \\ 0, & 1, & 224, & -666 \\ 1, & -6, & -1350, & 4010 \\ -4, & 24, & 5401, & -16043 \end{bmatrix} \right)$$

90 (U*A*V-B);//verification.

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

91 (U^(-1)),(U^(-1)*B);//est une base de M resp N;

$$\left(\begin{bmatrix} 94, & -281, & -15808, & 1118 \\ 1, & 0, & 0, & 0 \\ 80, & -236, & -13277, & 939 \\ 208, & -616, & -34656, & 2451 \end{bmatrix}, \begin{bmatrix} 188, & -1124, & -252928, & 751296 \\ 2, & 0, & 0, & 0 \\ 160, & -944, & -212432, & 631008 \\ 416, & -2464, & -554496, & 1647072 \end{bmatrix} \right)$$

92 ----------Exercice----------

93 M:=matrix(7,7,(ii,jj)->if ii=jj-1 then 1 else 0 fi):;M[1,1]:=1:;M[2,2]:=1:;M[2,3]:=0:;M[4,5]:=0:;M;

// Success

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

M

94 pari(); //on charge pari

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 divi
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

M

95 Dans pari l'instruction pour la forme de smith est matsnf. Ici il faut mettre l'option 2 pour preciser que la matrice est

96 matsnf(M-x*identity(7),2);

$$\left[x^4-2\cdot x^3+x^2,\ x^2,\ x,\ 1,\ 1,\ 1,\ 1\right]$$

M

97