
1 Logiciels, niveau, objectifs

1. Logiciel : un logiciel de géométrie dynamique.
2. Connaissances mathématiques : notion de partie entière.
3. Objectifs : réfléchir sur la notion d'équation d'un ensemble géométrique (la justification oblige à une réflexion sur le lien donné par l'équation entre l'abscisse et l'ordonnée d'un point de l'ensemble).
4. Connaissances TICE : notion de test, notion de trace.
5. Classes : TS, 1S.

2 Travail avec un logiciel de géométrie dynamique

[.] désigne la fonction partie entière.

Avec un logiciel de géométrie dynamique, construire le lieu des points M dont les coordonnées vérifient :

1.

$$[x]^2 + [y]^2 = 4$$

2.

$$[x] + [y] = 1$$

3.

$$[y] = [x]^2$$

3 Justification

4 Prolongement

L'espace étant muni d'un repère orthonormé, quel est le lieu des points $M(x; y; z)$ tels que :

$$[x]^2 + [y]^2 + [z]^2 = 4 \quad ?$$

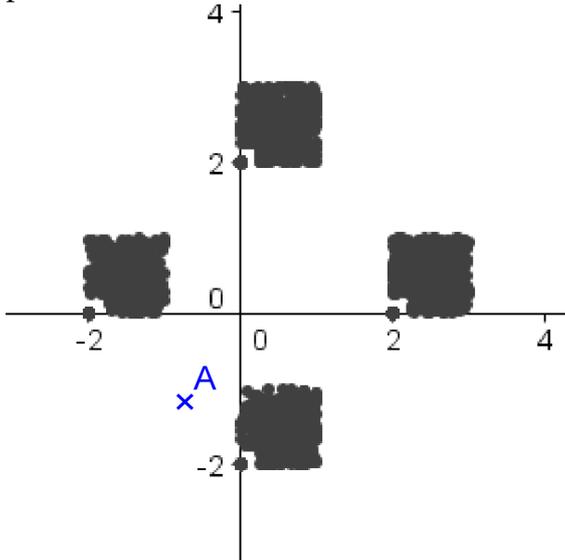
Éléments de réponse

5 Construction geogebra

On place un point A libre dans le plan. On définit ensuite un point M par :

$$M = \text{Si}[\text{floor}(x(A))^2 + \text{floor}(y(A))^2 \stackrel{?}{=} 4, A]$$

On active ensuite la trace de M , il suffit alors de déplacer le point A pour que M laisse sa trace sur les positions solutions.



6 Construction Xcas

On ouvre un module de géométrie 2D (alt g ou edit, ajouter, geo 2d). On définit un point A par

```
A:=point(0,0)
```

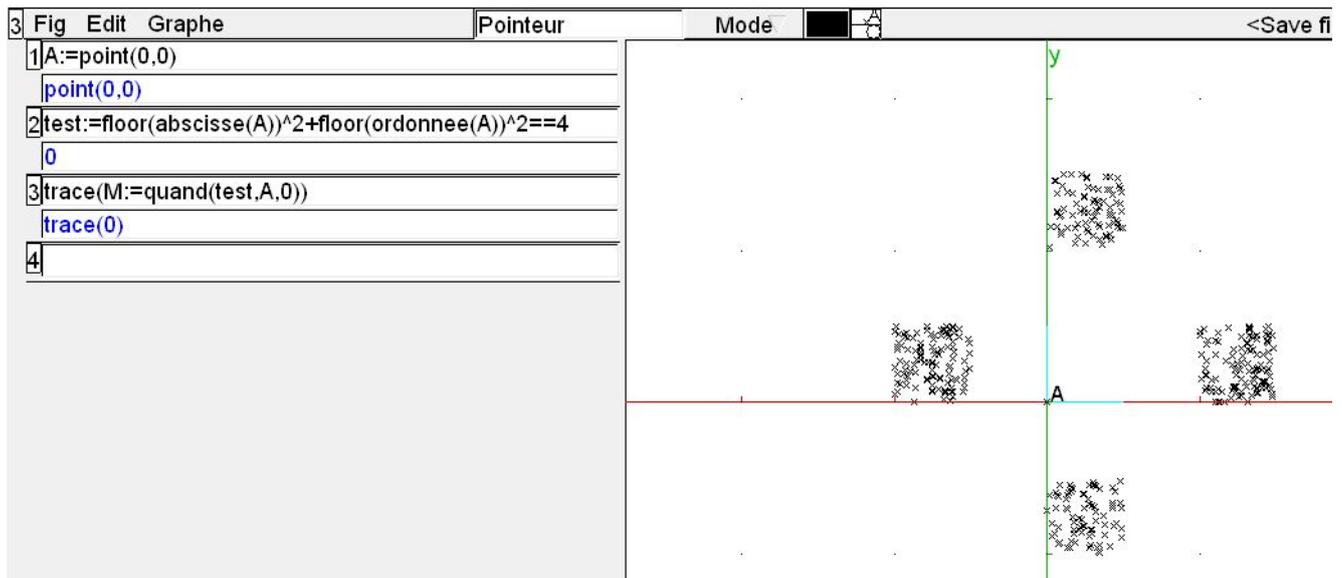
Au niveau 2, on définit le test :

```
test:=floor(abscisse(A))^2+floor(ordonnee(A))^2==4
```

Au niveau 3 :

```
trace(M:=quand(test,A,0))
```

On passe ensuite en mode pointeur et on balade le point A à la souris.



7 Construction geoplan

On définit un point libre A dans le plan.

On définit une fonction f de deux variables :

Fonction (expression à 2 variables) ✕

Variable muette 1:

Variable muette 2:

Expression de la fonction: R

Nom de la fonction:

On définit ensuite l'abscisse et l'ordonnée du point A :

Abscisse d'un point ✕

Nom du point: R

Nom de l'abscisse:

ordonnée d'un point ✕

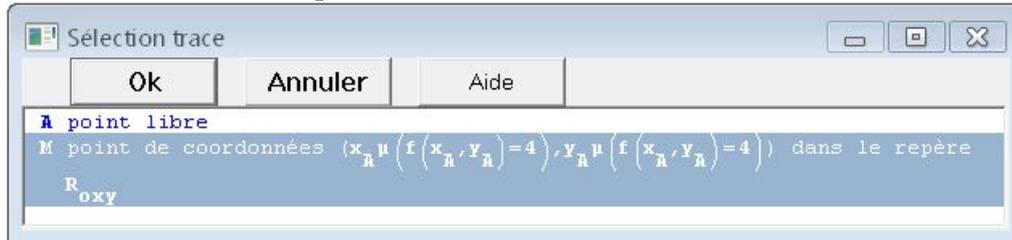
Nom du point: R

Nom de l'ordonnée:

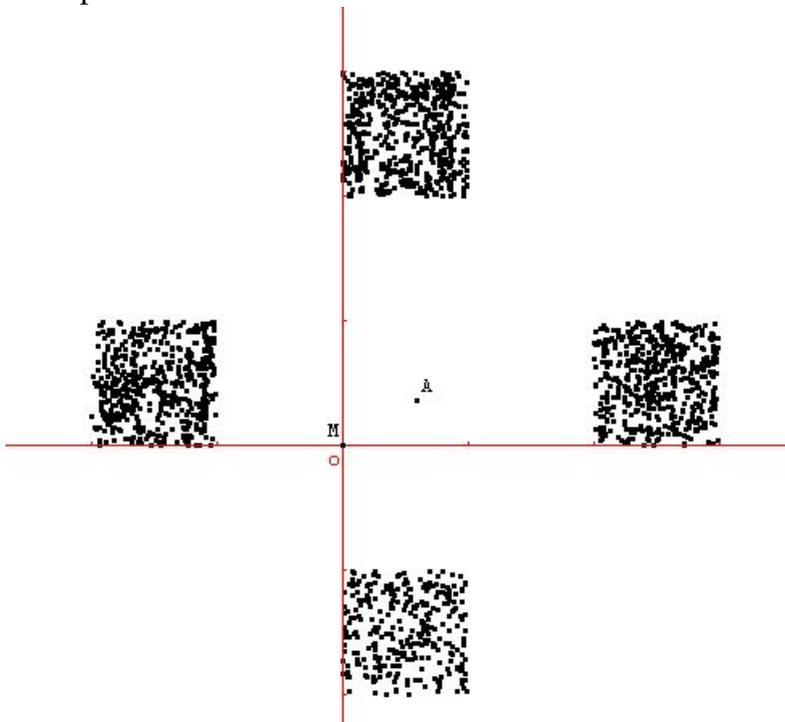
On définit ensuite le point M comme suit :



On sélectionne alors le point M avec l'outil trace



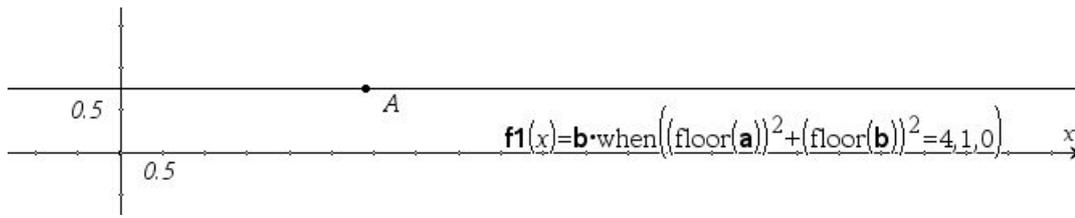
et on passe en mode trace :



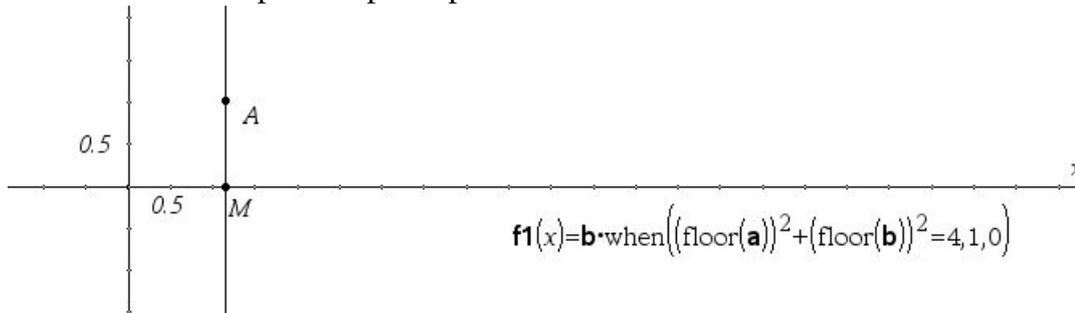
8 Avec la ti nspire

On place un point libre A dans le plan. On affiche les coordonnées du point A et on stocke l'abscisse dans a et l'ordonnée dans b .

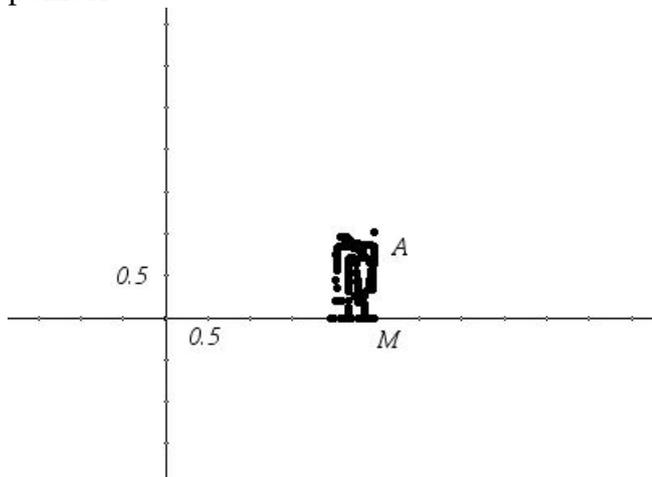
On définit ensuite la fonction $f1$ par l'expression :



On construit alors un point M à l'intersection de la courbe de la fonction f_1 et de la perpendiculaire à l'axe des abscisses passant par le point A .

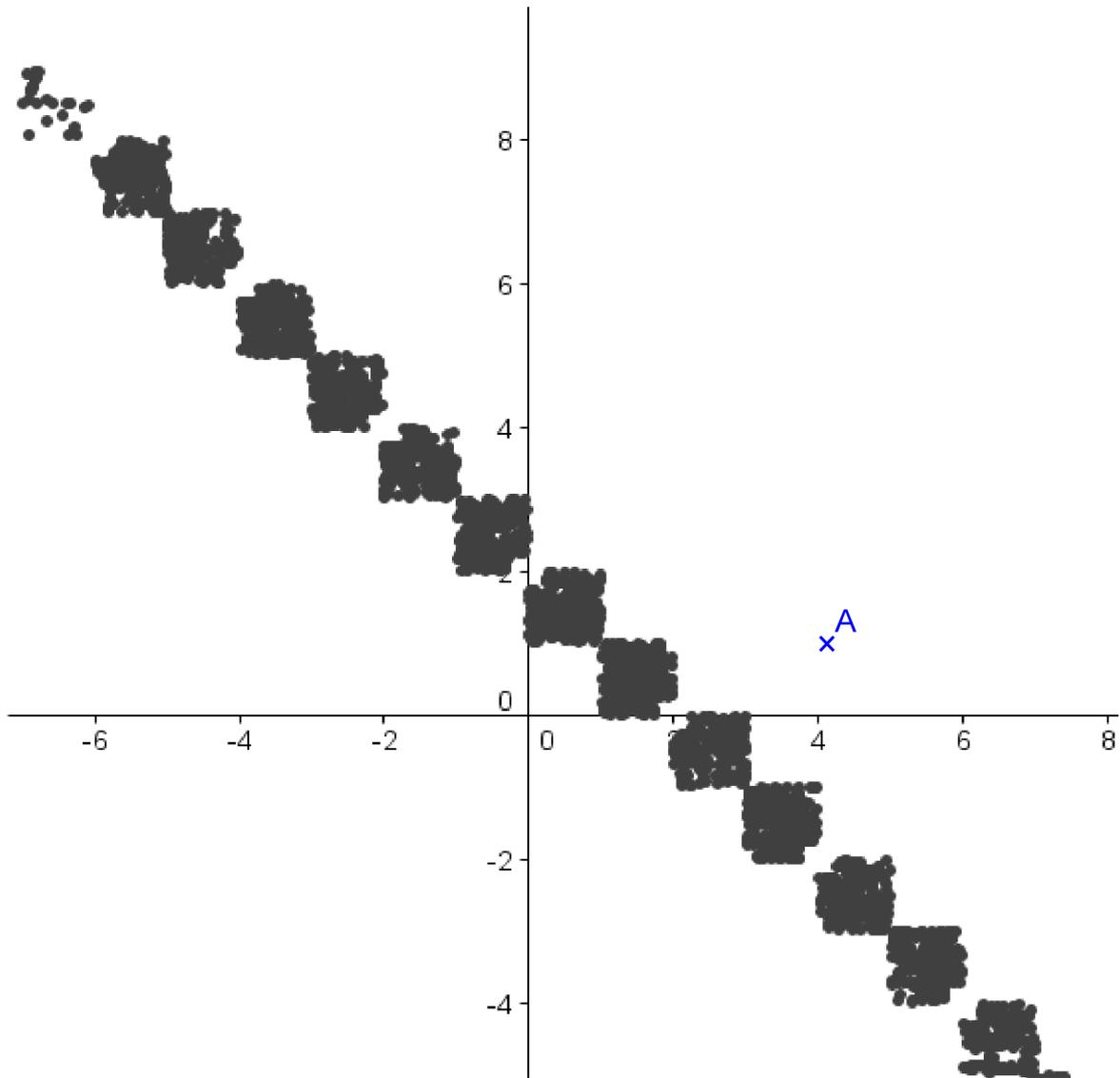


On cache alors cette perpendiculaire ainsi que la courbe de f_1 , on active la trace de M et on déplace le point A :

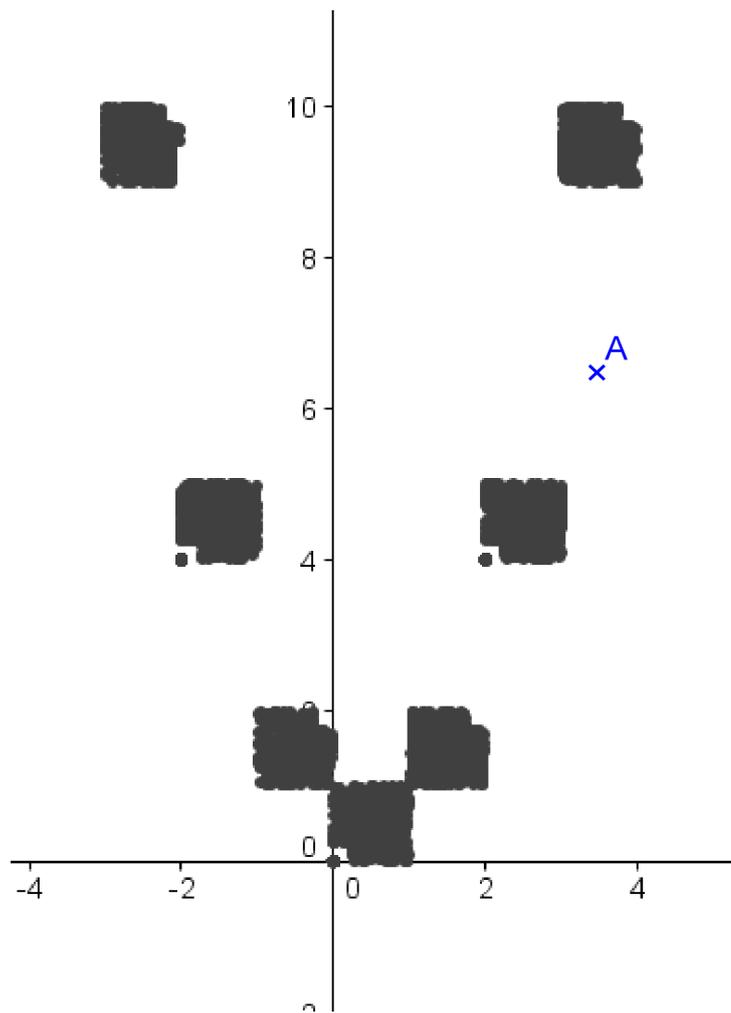


Le nombre de points laissés en trace étant malheureusement limité, on n'obtient rien de bien convaincant.

9 $E(x) + E(y) = 1$



10 $E(y) = (E(x))^2$



11 Dans l'espace

Dans l'espace, on pourra faire une approche à l'aide de Xcas.

Par exemple, avec le programme ci-dessous on détermine une liste de points solutions :

1	Prog Edit Add	nxt	OK
---	---------------	-----	----

```

partiedulieu() := (
  local despoints, a, b, c;
  despoints := [];
  for (a:=0; a<=3; a:=a+0.1) {
    for (b:=0; b<=3; b:=b+0.1) {
      for (c:=0; c<=3; c:=c+0.1) {
        if ((floor(a))^2+(floor(b))^2+(floor(c))^2==4)
          despoints:=append(despoints, point(a,b,c));
      }
    }
  }
  return despoints;
);

```

```

// Parsing partiedulieu
// Success compilind partiedulieu

```

Done

```

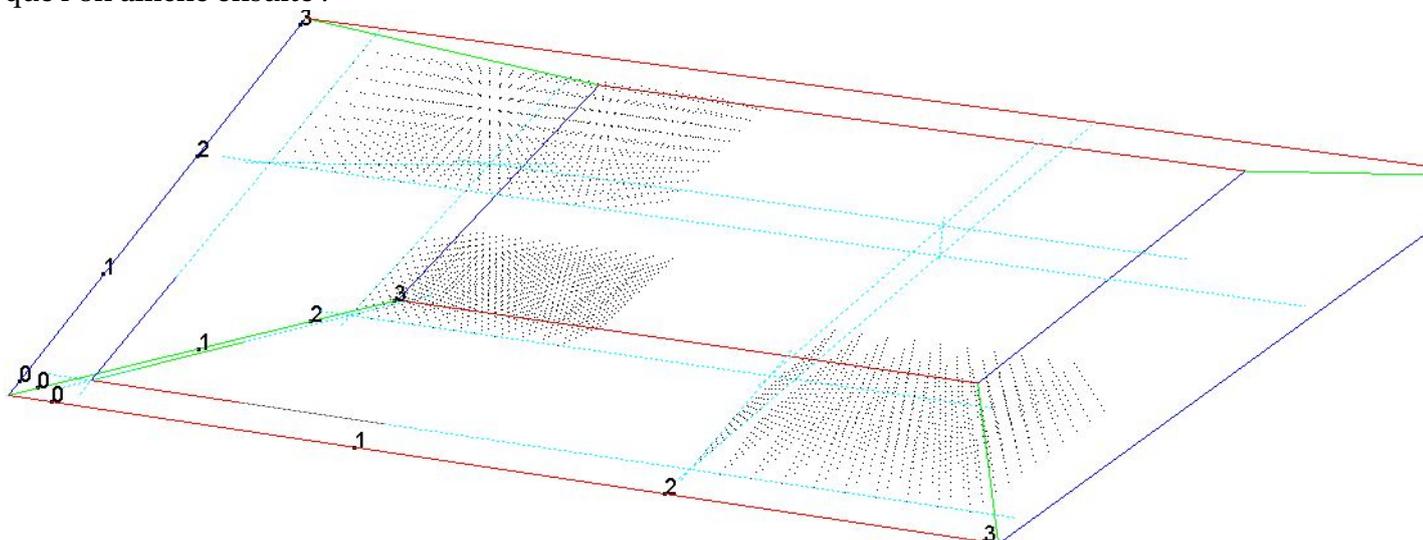
2 A:=partiedulieu();

```

Evaluation time: 37.439

Done

que l'on affiche ensuite :



Un programme plus rapide (voir explications ici : <http://pcm1.e.ujf-grenoble.fr/XCAS/viewtopic.php?t=275>) :

```

partiedulieu() := {
  local a, b, c, dx, dy, dz, resa, resb, resc;
  dx:=0.1; dy:=0.1; dz:=0.1;
  resa:=NULL;
  for (a:=0; a<=3; a+=dx) {
    resb:=NULL;
    for (b:=0; b<=3; b+=dy) {
      resc:=NULL;
      for (c:=0; c<=3; c+=dz) {
        if (floor(a)^2+floor(b)^2+floor(c)^2=4)
          resc:=resc, point(a, b, c, affichage=epaisseur_point_4);
      }
      resb:=resb, resc;
    }
    resa:=resa, [resb];
  }
  return resa;
}
:;

```

```

// Parsing partiedulieu
// Success compiling partiedulieu

```

Done

partiedulieu()

Evaluation time: 7.021

et un résultat :

