Géométrie 2-d et traduction pour `Xcas`

Renée De Graeve

8 septembre 2019

Index

Chapitre 1

Pour débuter en géométrie

Tapez `Alt+g` pour ouvrir un niveau de géométrie 2-d.
Tapez `Alt+h` pour ouvrir un niveau de géométrie 3-d.
Ou bien utilisez le menu `Geo`▶ `New figure 2-d` (resp `Geo`▶ `New figure 3-d`) pour ouvrir une fenêtre graphique 2-d (resp 3-d) avec sa barre de menus, ses lignes de commandes (à gauche de lécran graphique) et ses boutons (à droite de lécran graphique).
Les menus des commandes graphiques se trouvent dans le menu `Geo` ou dans le bandeau et si on a choisit `Voir Bandeau` dans le menu `Cfg`▶`Montrer`▶`Bandeau`.
Chaque bouton du bandeau ouvre les sous menus et il faut cliquez sur `home` pour revenir au bandeau initial.

1.1 Réglage de la fenêtre

`Xcas` fait de la géométrie analytique : vous avez donc sous-jacent un système de coordonnées.
Cliquez sur le menu `Cfg` pour initialiser le graphique et changer la configuration des futurs graphiques.
On fait ainsi apparaitre le réglage des coordonnées :
`X-, Y-, Z-, X+, Y+, Z+` les coordonnées où sont faits les calculs,
`WX-, WY-, WX+, WY+` désignent les coordonnées de ce qui est visible,
`t-, t+` désigne les bornes du paramètre des tracés paramétriques ou polaires.
`x_rot, z_rot` permettent de changer la vue d'un dessin en 3-d.
Si vous voulez avoir un autre réglage au cours de votre travail, il faut régler alors la configuration avec le bouton `cfg` situé à droite de l'écran graphique. Si vous ne voulez pas voir les axes :
décocher `Montrer les axes` de la fenêtre du réglage des coordonnées, puis, validez votre choix par `OK` ou
taper `switch_axes(0).`
En 3-d, utilisez aussi le menu `M->3-d` situé à droite de l'écran graphique qui vous permet d'avoir une :

```
vue de face x=cst
vue de cote y=cst
vue de dessus z=cst
```

1.2 Comment imprimer un graphique 2-d

1.2.1 Pour avoir un fichier Latex : `graph2tex`

`graph2tex` a pour argument le nom d'un fichier.
`graph2tex` transforme le graphique en un fichier Latex de nom, le nom spécifié. Vous pouvez alors utiliser ce fichier de façon indépendante ou l'insérer dans un texte Latex. Dans ce cas il faudra enlever l'en tête :
`\documentclass{article}...\begin{document}`,
et enlever à la fin :
`\end{document}`
et de rajouter :
`\usepackage{pstricks}`
dans l'en-tête du fichier dans lequel on l'insère.

1.2.2 Avoir le graphique dans l'impression de l'historique : `graph2tex()`

`graph2tex()` lorsque `graph2tex` n'a pas d'argument cela a pour effet d'insérer dans l'historique ce qu'il faut, pour avoir le graphique dans l'impression de l'historique : vous pouvez voir ce que vous allez imprimer avec le menu `Fich` sous-menu `hist (Historique) imprimer` en choisisant :
`Pré-visualisation`.
Remarque Pour toutes les commandes qui commencent par `plot` on n'a pas besoin de la commande `graph2tex()`, le graphique est inséré automatiquement dans l'historique...si vous ne le voulez pas utiliser une commande synonyme ne commençant pas par `plot` !!!

1.2.3 En utilisant le menu `M` de `Xcas`

En utilisant dans le menu `M` du bloc des boutons situé à droite d'une sortie graphique :
`Exporter/Imprimer` vous pouvez sauver votre graphique en Latex, en postscript et en png.

1.3 Comment imprimer un graphique 3d

1.3.1 Pour avoir un fichier Latex : `graph3d2tex`

`graph3d2tex` a pour argument le nom d'un fichier.
`graph3d2tex` transforme le graphique 3-d en un fichier Latex de nom, le nom spécifié. Vous pouvez alors insérer ce fichier dans un texte Latex en ajoutant :
`\usepackage{pstricks}`
dans l'en-tête du fichier dans lequel on l'insère.

1.3.2 En utilisant les menus de `Xcas`

En utilisant dans le menu `M` du bloc des boutons situé à droite d'une sortie graphique :

`Exporter/Imprimer` vous pouvez sauver votre graphique en Latex, en postscript et en png.

1.4 Instructions élémentaires

1.4.1 La fenêtre graphique

On peut régler la fenêtre graphique avec la commande `xyztrange`, on écrira par exemple :
`xyztrange(-6.0,6.0,-7.0,4.0,-10.0,10.0,-1.0,6.0,`
`-5.0,5.0,-2.0,4.0,1,0.0,1.0)` pour avoir :
`X-=-6, X+=6` (valeurs des x calculés),
`Y-=-7, Y+=4` (valeurs des y calculés),
`Z-=-10, Z+=10`,
`t-=-1, t+=6` (valeurs du paramètre en paramétrique ou en polaire),
`WX-=-5, WX+=5`, (valeur des x visibles),
`WY-=-2, WY+=4` (valeurs des y visibles),
`1` pour voir les axes (ou `0` pour ne pas les voir),
`0.0` pour définir la valeur de `class_min` (pour faire des statistiques)
`1.0` pour définir la valeur de `class_size` (pour faire des statistiques).
La commande `xyztrange` a 15 paramètres, il est donc préférable lorsque on veut changer un paramètre ou régler la fenêtre graphique de le faire à l'aide du bouton `cfg`.

1.4.2 Les axes

Si vous voulez voir, (ou ne pas voir) les axes sans être obligé d'ouvrir la fenêtre d'initialisation graphique :
taper `switch_axes()`.
Pour faire réapparaitre les axes on fait la même chose :
`switch_axes()` dans la ligne de commande puis `Entrée`.
Dans un programme on utilisera la commande :
`switch_axes(0)` pour ne pas voir les axes et `switch_axes(1)` pour les voir.

1.4.3 Gestion de la fenêtre graphique

- `erase()` ou `erase` efface lécran graphique `DispG` et influe sur la commande `graph2tex` pour que seules les sorties graphiques faites aprés la commande `erase()` soient prises en compte par la commande `graph2tex`,
- `ClrGraph()` ou `ClrGraph` efface l'écran `DispG`,
- `couleur` permet de changer la couleur du graphique.
  `couleur` a un argument (une couleur) ou deux arguments (un objet géométrique et une couleur) :
  `noir` ou `0`,
  `rouge` ou `1`,
  `vert` ou `2`,

`blanc` ou `3`,
`bleu` ou `4`,
`rose` ou `5`,
`vert` ou `6` etc...
Par exemple : `couleur(point(1+i),rouge)`

— `legende` permet de mettre une legende sur le graphique.
`legende` a deux arguments un nombre complexe (ou un point) et une chaine de caractères.
Par exemples :
`legende(1+i,"le point 1+i")` ou
`couleur(legende(1+i,"le point 1+i"),rouge)`

1.4.4 Un point

Pour obtenir un point il suffit de cliquer avec la souris (bouton gauche) pour qu'un point s'affiche avec un nom.
Ce nom est crée automatiquement : `A`, `B`, `C` puis `E` etc...
Attention Il ne sera pas créer de point `D` automatiquement car pour `Maple` `D` représente la fonction dérivée d'une fonction.
En dehors du mode `Maple` vous pouvez utiliser `D` comme nom de variable (par exemple `D:=point(1-i)`).
On peut aussi taper :
`M:=point((1,2))` ou `N:=point(1+3*i)` ou
`P:=point(1)` ou `O:=point(0).`
cela dessine les points `M N P O` dans le repère défini dans la fenêtre d'initialisation graphique (bouton `cfg`).
Tous les points ainsi crées peuvent se déplacer ainsi que les constructions qui en dépendent. Pour cela on clique sur le point :
le nom du point se met dans la ligne de commande et le point change de couleur et devient bleu, sans relacher le bouton de la souris on déplace le point.

1.4.5 Un point sur un objet géomètrique

Pour dire que le point A doit se trouver sur l'objet géomètrique `G` on tape :
`A:=element(G)` ou bien
`A:=element(G,1)` ou bien
`A:=element(G,t)` où `t` est un paramètre qui représente le paramètrage de l'objet `G`.
Si on veut pouvoir faire bouger `t` à l'aide de la souris il faudra définir `t` avec la commande `element`.
Par exemple :

— l'objet géomètrique est un cercle
On tape pour tracer le cercle de centre l'origine et de rayon `2` :
`C:=cercle(0,2)`
puis,
`t:=element(0..2*pi)` pour faire apparaître en haut et à droite de l'écran un segment (ici $[0, 2*\pi]$) muni d'un trait vertical (situé au début au milieu de l'intervalle) qui indique la valeur choisie comme `t`, trait que l'on

peut faire bouger avec la souris pour modifier la valeur de `t`.
`A:=element(C,t)` dessine le point `A` du cercle `C` tel que :
t=angle($\overrightarrow{Ox}, \overrightarrow{OA}$) au début $\mathtt{t} = \pi$ puisque π est le milieu de l'intervalle qui définit `t`.
On peut alors faire varier `t` en bougeant le trait vertical avec la souris, et quand on bouge ce trait vertical, `A` se déplace sur le cercle `C`.

— l'objet géomètrique est une droite
  On tape pour tracer la droite `AB` :
  `A:=point(1); B:=point(3);D:=droite(A,B);`
  puis,
  `t:=element(-1..2)` pour faire apparaître en haut et à droite de l'écran un segment (ici $[-1, 2]$) muni d'un trait vertical (situé au début au milieu de l'intervalle) qui indique la valeur choisie comme `t`, trait que l'on peut faire bouger avec la souris pour modifier la valeur de `t`.
  `M:=element(D,t)` dessine le point `M` de la droite `D` tel que :
  $M - A = t * (B - A)$ soit `M:=(1-t)*A+t*B`.
  On peut alors faire varier `t` en bougeant le trait vertical avec la souris, et quand on bouge ce trait vertical, `M` se déplace sur le segment EF, défini par `E:=point(-1)` $(t = -1)$ `F:=point(5)` $(t = 2)$

— l'objet géomètrique est un segment
  On tape pour tracer pour tracer le segment `AB` :
  `A:=point(1); B:=point(3);S:=segment(A,B);`
  puis,
  `t:=element(0..1)` pour faire apparaître en haut et à droite de l'écran un segment (ici $[-1, 2]$) muni d'un trait vertical (situé au début au milieu de l'intervalle) qui indique la valeur choisie comme `t`, trait que l'on peut faire bouger avec la souris pour modifier la valeur de `t`.
  Attention dans ce cas `M:=element(S,t)` dessine le point `M` du segment `S` avec `M` entre `A` et `B` tel que :
  $M - A = t * (B - A)$ soit `M:=(1-t)*A+t*B`.
  On peut alors faire varier `t` en bougeant le trait vertical avec la souris, et quand on bouge ce trait vertical, `M` se déplace sur le segment AB et cela même si on a tapé `t:=element(-1..2)` car pour $t < 0$ `M` restera en `A` et pour $t > 0$ `M` restera en `B`.

1.4.6 Fonctions s'appliquant à un point

Dans ce qui suit on suppose que l'on a défini deux points `A` et `B` et que Ox et Oy désignent les axes du repère défini dans la fenêtre d'initialisation graphique.
`re(A)` désigne le point projection orthogonale de `A` sur Ox.
`im(A)` désigne le point projection orthogonale de `A` sur Oy.
`abscisse(A)` désigne l'abscisse de `A`.
`ordonnee(A)` désigne l'ordonnée de `A`.
`coordonnees(A)` désigne la liste des coordonnées de `A`.
`affixe(A)` désigne le nombre complexe qui est l'affixe de `A` dans ce repère.
$\lambda *$ `A`, la multiplication d'un nombre complexe $\lambda = k * e^{i*\theta}$ par un point `A`, est un point qui se déduit de `A` par une similitude de rapport k et d'angle θ.
`B-A`, la différence de deux points, est un nombre complexe égal à la diffèrence des

affixes de ces deux points (c'est l'affixe du vecteur $\overrightarrow{AB}$).
`A+B`, la somme de deux points, est un nombre complexe égal à la somme des affixes de ces deux points (c'est l'affixe du point C tel que $OACB$ soit un parallélogramme).
`C+A-B` où `A,B,C` sont des points, désigne le point transformé de `C` dans la translation de vecteur $\overrightarrow{BA}$ (`A-B` est un nombre complexe qui est l'affixe de $\overrightarrow{BA}$).
`C+u` où `C` est un point et `u` est un nombre complexe, désigne le point transformé de `C` dans la translation de vecteur $\overrightarrow{U}$ d'affixe u.
Attention
si `C` a été défini comme `element(L)` cela définit la pojection orthogonale de ce translaté sur `L`.
Dans ce cas c'est `point(affixe(C)+A-B)` (resp `point(affixe(C)+u)`) qui est le translaté de `C` dans la translation de vecteur $\overrightarrow{BA}$ (resp $\overrightarrow{U}$).
`longueur(A,B)` désigne la longueur du segment AB.
`longueur2(A,B)` désigne le carré de la longueur du segment AB.
`milieu(A,B)` désigne le point milieu du segment AB.
`isobarycentre(A,B,C)` désigne l'isobarycentre des points A, B, C c'est à dire le centre de gravité du triangle A, B, C.

1.4.7 Affixe d'un vecteur ou différence de 2 points

Un vecteur libre est défini par deux points (son représentant) et est entièrement défini par son affixe qui est le nombre complexe égal à la différence de ces deux points :
en effet la différence de deux points est un nombre complexe égal à la différence des affixes de ces deux points, et ce nombre complexe représente un vecteur d'origine O.
Exemples
Si on a deux points `A` et `B` le vecteur $\overrightarrow{AB}$ sera désigné par le nombre complexe `B-A`.
`D:=C+A-B` : D est un point tel que $ABCD$ est un parallélogramme puisque $\overrightarrow{CD} = \overrightarrow{BA}$ (`D-C=A-B`).
`abs(B-A)` désigne la longueur du vecteur $\overrightarrow{AB}$,
`inv(B-A)` désigne l'inverse du nombre complexe `B-A` : c'est donc l'affixe d'un vecteur paralléle à `conj(B-A)`,
`conj(B-A)` désigne le conjugué du nombre complexe `B-A` : c'est donc l'affixe d'un vecteur paralléle au symétrique-`Ox` de `B-A`.

1.4.8 Segment, demi-droite, droite

Pour dessiner un segment : on clique pour avoir un point et on déplace la souris en gardant le bouton gauche enfoncé, puis, on déplace la souris jusqu'au point désiré et on relâche le bouton de la souris. Là encore les objets crées sont nommés automatiquement (par exemple `A`, `B` pour les points et `AB` pour le segment reliant `A` et `B`).
On peut aussi utiliser les commandes :
`segment, droite, demi_droite` avec comme paramètres 2 points (ou une liste de 2 points) ou 2 nombres complexes (ou une liste de 2 nombres complexes)

ou encore un point et un nombre complexe et vis et versa.
On peut donc écrire si on a auparavant défini deux points `A` et `B` :
`segment(A,B)` ou `segment(-1+2*i,1+i)` ou `segment([A,B])` pour définir un segment.
`droite(A,B)` ou `droite(-1+2*i,1+i)` ou `droite([A,B])` pour définir une droite.
`demi_droite(A,B)` ou `demi_droite(i,1-i)` ou `demi_droite([A,B])` pour définir une demi-droite d'origine `A` et contenant `B` (ou une demi-droite d'origine `point(-1+2*i)` et contenant `point(1+i)`).
Remarques
- Si on a défini un segment par deux points ou par des nombres complexes on a la possibilité de donner un nom á ses extrémités en rajoutant dans les arguments deux noms de variables.
On tape par exemple :
`segment(-1+2*i,1+i,K,L)` pour définir le point `K` d''affixe $-1+2*i$ et le point `L` d''affixe $1+i$ ou encore
`segment(point(-1+i),1-i,M,N)` pour définir le point `M` d''affixe $-1+i$ et le point `N` d''affixe $1-i$.
- Si on a défini une demi_droite par deux points ou par des nombres complexes on a la possibilité de donner un nom á son extrémité en rajoutant dans les arguments un nom de variables.
On tape par exemple :
`demi_droite(-1+2*i,1+i,P)` pour définir le point `P` d''affixe $-1+2*i$.

1.4.9 Le vecteur en géométrie plane : `vecteur`

`vecteur`, en géométrie plane, a comme arguments soit :
- deux points A, B ou deux nombres complexes représentant l'affixe de ces points ou deux listes représentant les coordonnées de ces points.
  `vecteur` définit et dessine le vecteur $\overrightarrow{AB}$
- un point A (ou un nombre complexe représentant l'affixe de ce point ou une liste représentant les coordonnées de ce point) et un vecteur V (définition récursive).
  `vecteur` définit et dessine le vecteur $\overrightarrow{AB}$ tel que $\overrightarrow{AB}=\overrightarrow{V}$.

On tape :

```
vecteur(point(-1),point(i))
```

Ou on tape :

```
vecteur(-1,i)
```

Ou on tape :

```
vecteur([-1,0],[0,1])
```

On obtient :

```
Le tracé du vecteur d'origine -1 et d'extrémité i
```

On tape :

```
V:=vecteur(point(-1),point(i))
```

On tape :

```
vecteur(point(-1+i),V)
```

Ou on tape :

```
vecteur(-1+i,V)
```

Ou on tape :

```
vecteur([-1,1],V)
```

On obtient :

```
Le tracé du vecteur d'origine -1+i et d'extrémité 2*i
```

Remarque
En calcul formel, on travaille sur la liste des coordonnées des vecteurs que l'on obtient avec la commande `coordonnees` (cf 1.4.6).

Exercice1

Soient 3 points A, B, C d'affixe a, b, c. Montrer que le triangle ABC est équilatéral direct si et seulement si $a + b * j + c * j^2 = 0$ où $j = \exp(2i\pi/3)$.

La solution sans `Xcas`

On a $j^2 = \exp(4i\pi/3)$ et $-j^2 = \exp(i\pi/3)$ et $1 + j + j^2 = 0$
ABC est équilatéral direct est équivalent à :
$(c - a) = -j^2(b - a)$ donc est équivalent à :
$c - a(1 + j^2) + bj^2 = c + a * j + b * j^2 = 0$ ou encore après multiplication par j^2 est équivalent à :
$a + b * j + c * j^2 = 0$.
La solution avec `Xcas`
On tape dans un niveau de géométrie :

```
supposons(b1=[1.3,-5,5,0.1]);
supposons(b2=[2.3,-5,5,0.1]);
b:=b1+(i)*b2;
B:=point(b);
supposons(c1=[-2.2,-5,5,0.1]);
supposons(c2=[3.5,-5,5,0.1]);
c:=c1+i*c2;
C:=point(c);
j:=exp(2*i*pi/3);
a:=-b*j-c*j^2;
A:=point(a);
est_equilateral(A,B,C);
normal((c-a)+j^2*(b-a));
```

La réponse pour `est_equilateral(A,B,C);` est `1` et
la réponse pour `normal((c-a)+j^2*(b-a));` est `0` et comme tout les calculs sont faits avec des paramètres formels cela vaut une démonstration. Pour la réciproque, on tape :

```
supposons(b1=[1.3,-5,5,0.1]);
supposons(b2=[2.3,-5,5,0.1]);
b:=b1+(i)*b2;
B:=point(b);
supposons(c1=[-2.2,-5,5,0.1]);
supposons(c2=[3.5,-5,5,0.1]);
c:=c1+i*c2;
C:=point(c);
j:=exp(2*i*pi/3);
triangle_equilateral(B,C,A);
a:=affixe(A)
normal(a+b*j+c*j^2);
```

la réponse pour `normal(a+b*j+c*j^2);` est `0` et comme tout les calculs sont faits avec des paramètres formels cela vaut une démonstration.

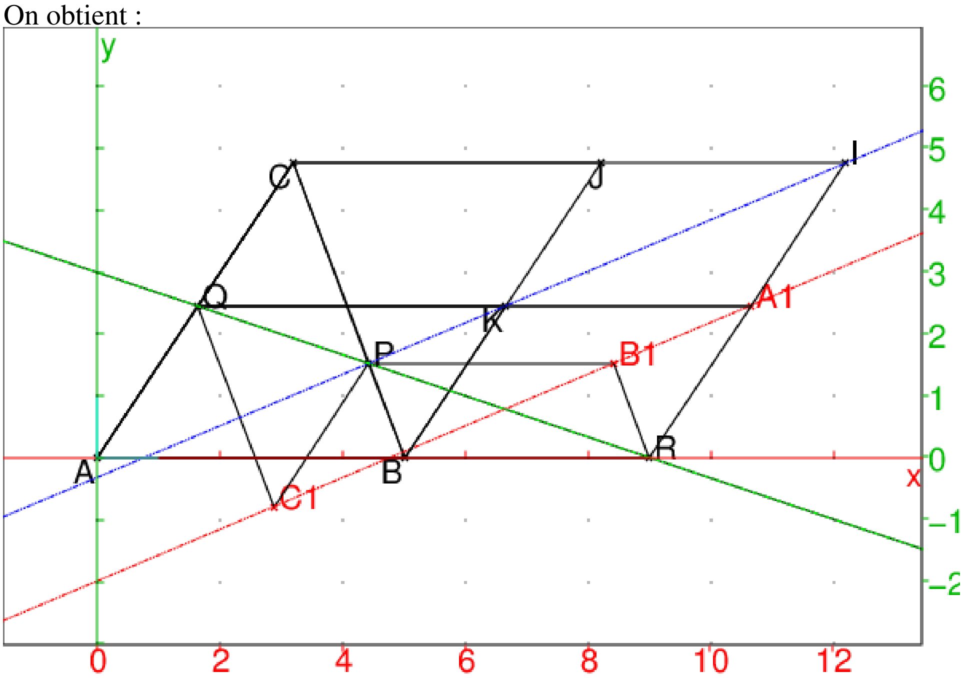
Exercice2

Soit ABC un triangle et une droite d qui coupe BC, AC, AB en P, Q, R.
On construit les trois parallélogrammes AQA_1R, BPB_1R, CQC_1P.
Démontrer que A_1, B_1, C_1 sont alignés.
Avec `Xcas`, on tape dans un niveau de géométrie :

```
A:=point(0);
B:=point(5);
C:=point(3+5*i);
triangle(A,B,C);
d:=droite(y=-x/3+3,affichage=2);
P:=inter_unique(d,droite(B,C));
Q:=inter_unique(d,droite(A,C));
R:=inter_unique(d,droite(B,A));
A1:=affichage(point(A+(R-A)+(Q-A)),1);
polygone(A,R,A1,Q);
B1:=affichage(point(B+(P-B)+(R-B)),1);
polygone(B,P,B1,R);
C1:=affichage(point(C+(P-C)+(Q-C)),1);
polygone(C,P,C1,Q);
droite(A1,C1,affichage=1+ligne_tiret_pointpoint);
I:=Q+(A1-Q)+(C-Q);
polygone(C,Q,A1,I);polygone(J,C,A,B)
K:=A+(B-A)+(Q-A);
polygone(K,Q,A,B);
J:=A+(B-A)+(C-A);
```

```
polygone(J,C,A,B);
droite(P,I,affichage=4+ligne_tiret_pointpoint)
```

On obtient :



On tape :
`est_aligne(A1,B1,C1)`
On obtient : `1`

Comment le démontrer ?

On utilise le **théorème de Ménélaius** :

Une condition necessaire et suffisante pour que 3 points $P,\ Q,\ R$ situés respectivement sur les côtés $BC,\ CA,\ AB$ d'un triangle ABC, soient alignés est :

$$\frac{\overline{PB}}{\overline{PC}} \times \frac{\overline{QC}}{\overline{QA}} \times \frac{\overline{RA}}{\overline{RB}} = 1$$

On a donc cette relation puisque $P,\ Q,\ R$ sont alignés.
On construit 3 autres parallélogrammes :
$ABKQ$, $ACIR$ et $ABJC$ ce qui définit les points I, J et K.
Montrons que $P,\ K,\ I$ sont alignés. Pour cela on utilise le théorème de Ménélaius appliqué au triangle BCJ.
On a :
$\frac{\overline{RA}}{\overline{RB}} = \frac{\overline{IC}}{\overline{IJ}}$ et $\frac{\overline{QC}}{\overline{QA}} = \frac{\overline{KJ}}{\overline{KB}}$
Donc :
$\frac{\overline{PB}}{\overline{PC}} \times \frac{\overline{IC}}{\overline{IJ}} \times \frac{\overline{KJ}}{\overline{KB}} = 1$ ce qui prouve que $P,\ K,\ I$ sont alignés.
On a :
$\overrightarrow{C_1A_1} = \overrightarrow{C_1Q} + \overrightarrow{QA_1}$
CQC_1P est un parallélogramme donc : $\overrightarrow{C_1Q} = \overrightarrow{PC}$.
CQA_1I est un parallélogramme donc : $\overrightarrow{QA_1} = \overrightarrow{CI}$.
Donc : $\overrightarrow{C_1A_1} = \overrightarrow{PI}$
De même on a :

$\overrightarrow{A_1B_1} = \overrightarrow{A_1R} + \overrightarrow{RB_1}$
BRA_1K est un parallélogramme donc : $\overrightarrow{A_1R} = \overrightarrow{KB}$.
BRB_1P est un parallélogramme donc : $\overrightarrow{RB_1} = \overrightarrow{BP}$.
Donc : $\overrightarrow{A_1B_1} = \overrightarrow{PK}$
Donc puisque $P,\ K,\ I$ sont alignés les vecteurs $\overrightarrow{PI}$ et $\overrightarrow{PK}$ sont colinéaires et donc les vecteurs $\overrightarrow{C_1A_1}$ et $\overrightarrow{A_1B_1}$ sont colinéaires donc A_1, B_1, C_1 sont alignés.

1.4.10 Un cercle

Un cercle se définit par deux paramètres :
- soit par les extémités de son diamètre (le deuxième paramètre doit être un point),
- soit par son centre et son rayon (le deuxième paramètre doit être un nombre complexe).
On peut aussi définir des arcs de cercle en rajoutant deux paramètres : les angles au centre des points qui définissent l'arc mesurés en radians ou en degrés selon le choix fait (cf `Cfg->Configuratioon du CAS`) et à partir de l'axe `Ox`.
Dans la suite on suppose que l'on a définit deux points `A` et `B` :

— Cercle de diamètre AB
On tape :
`cercle(A,B)` trace le cercle de diamètre AB et
`cercle(point(i),point(1+2*i))` ou `cercle(i,point(1+2*i))` trace le cercle de diamètre défini par les points d'affixe i et 1+2*i.

— Cercle de centre A et de rayon r
Remarque
Si r est un nombre complexe le rayon est égal à `abs(r)`.
On tape :
`cercle(A,2.1)` trace le cercle de centre A et de rayon 2.1.
`cercle(i,1+2*i)` trace le cercle de centre, le point d'affixe i, et de rayon, `abs(1+2*i)`=$\sqrt{5}$ c'est à dire le cercle passant par le point d'affixe 1+3*i (alors que `cercle(i,1.0+2.0*i)` trace le cercle de centre, le point d'affixe i, et de rayon, `abs(1.0+2.0*i)`=2.23607000000).

— Cercle de centre A et passant par B
`cercle(A,B-A)` trace le cercle de centre `A` passant par `B`.

— Arc de cercle
`cercle(A,B,pi/4,pi/2)` dessine (si on est en radians) un arc appartenant au cercle de diamètre AB et allant du point d'angle au centre $\pi/4$ au point d'angle au centre $\pi/2$ (ces angles sont mesurés à partir de l'axe AB).
`cercle(A,2.1,pi/3,2*pi/3)` dessine (si on est en radians) un arc appartenant au cercle de centre A, de rayon 2.1 et allant du point d'angle au centre $\pi/3$ au point d'angle au centre $2 * \pi/3$ (ces angles sont mesurés à partir d'une parallèle à l'axe Ox passant par le centre du cercle).
`cercle(A,B-A,pi/3,2*pi/3)` dessine (si on est en radians) un arc appartenant au cercle de centre A, passant par B, et allant du point d'angle au centre $\pi/3$ au point d'angle au centre $2 * \pi/3$ (ces angles sont mesurés à partir de l'axe AB).
`arc(A,B,pi/3)` dessine (si on est en radians) l'arc `AB` d'angle au centre `pi/3` et appartenant au cercle de centre $C = (a + b)/2 + i * (b - a)/(2 * \tan(pi/6))$.

Exemples
Pour définir un arc AB situé sur le cercle de diamètre CD, on tape :

```
arccercle1(C,D,A,B):=
      cercle(C,D,arg(A-milieu(C,D)),arg(B-milieu(C,D)));
```

Pour définir un arc AB situé sur le cercle de centre C et de rayon r, on tape :

```
arccercle2(C,r,A,B):=cercle(C,r,arg(A-C),arg(B-C));
```

On se repotera à 1.17 pour les cercles inscrits, exinscrits et circonscrits à un triangle.

1.4.11 Fonctions s'appliquant à un cercle

On suppose que l'on a défini un cercle `C`.
`centre(C)` désigne le centre du cercle `C`.
`rayon(C)` désigne le réel égal au rayon du cercle `C`.

1.4.12 Avoir l'un des points d'intersection de deux objets géométriques : `inter_droite inter_unique`

L'un des points d'intersection de deux objets géométriques est obtenu par la commande `inter_droite` ou `inter_unique`.
On tape :

```
inter_droite(droite(i,1),cercle(0,1))
```

On obtient :

```
Le point 1 est dessiné dans un écran graphique
```

On tape :

```
inter_droite(droite(i,1),droite(0,1+i))
```

On obtient :

```
Le point (1+i)/2 est dessiné dans un écran graphique
```

Remarque
Lorsque l'intersection comporte plusieurs points `inter_unique` renvoie l'un de ces points. Mais on peut mettre un 3ième argument pour le spécifier qui est un point ou une liste de points :

— un point et alors `inter_unique` renverra le point d'intersection le plus proche de ce point,
— une liste de points et alors `inter_unique` renverra un point d'intersection qui ne se trouve pas dans cette liste.

1.4.13 Liste des points d'intersection de deux objets géométriques : `inter`

L'intersection de deux objets géométriques est une liste de points qui est obtenu par la commande `inter`.
Attention :
On utilise la commande `inter` pour des objets géométriques et la commande `intersect` pour des ensembles ou des listes.
Exemples :
Soient `D1,D2` deux droites concourantes et `C` un cercle qui coupe `D1`.
`E:=inter(D1,C)` désigne la liste des deux points d'intersection de `D1` et `C` :
`E[0]` est le premier point de la liste et `E[1]` en est le second.
`inter(D1,D2)[0]` désigne le point d'intersection de `D1` et `D2`.

Soient `C1,C2` deux cercles concourants.
`inter(C1,C2)` désigne la liste des deux points d'intersection de `C1` et `C2` : ainsi `droite(inter(C1,C2))` désigne l'axe radical de `C1` et `C2`. On tape :
`inter(droite(i,1),cercle(0,1))`
On obtient :
`Les points i et 1 sont dessinés dans un écran graphique`
On tape :
`inter(droite(i,1),droite(0,1+i))[0]`
On obtient :
`Le point (1+i)/2 est dessiné dans un écran graphique`

1.5 Constructions élémentaires

1.5.1 Médiatrice d'un segment AB

Étant donné deux points `A` et `B` la commande :
`mediatrice(A,B)` trace la médiatrice du segment AB.
Activité
Créer un segment `AB`.
Construire la médiatrice de `AB`, en utilisant la même construction qu'avec un compas.
Réponse
On clique avec la souris pour avoir deux points `A` et `B` et le segment `AB`.
On tape :
`C1:=cercle(A,B-A)` trace le cercle de centre `A` passant par `B`.
`C2:=cercle(B,A-B)` trace le cercle de centre `B` passant par `A`.
`D:=droite(inter(C1,C2))` trace la droite joignant les deux points de l'intersection de `C1` et de `C2` (`inter(C1,C2)` est la liste des points de cette intersection).
La liste des instructions se trouve dans `geo1` : cliquer avec la souris pour avoir deux points `A` et `B` puis, faire `Ouvrir` du menu `Fich` de `Xcas` et sélectionner `geo1.xws`.
On peut aussi comme exercice de programmation définir la fonction `Mediatrice` (il faut commencer la fonction par une majuscule car `mediatrice` est une commande de `Xcas`).
On tape :

```
Mediatrice(A,B):=droite(inter(cercle(A,longueur(A,B)),
cercle(B,longueur(A,B))))
```

1.5.2 Milieu d'un segment [AB]

Étant donné deux points `A` et `B` la commande :
`M:=milieu(A,B)` trace le milieu `M` du segment $[AB]$.
Activité
Créer un segment `[AB]`.
Construire le milieu de `AB`, soit en utilisant les coordonnées, soit en utilisant la même construction qu'avec un compas.

Réponse
On tape : `M:=point((coordonnees(A)+coordonnees(B))/2)` ou bien on rajoute à la construction de la médiatrice (cf ci-dessus) :
```
M:=inter_unique(segment(A,B),D)
```
On peut aussi comme exercice de programmation définir la fonction `Milieu` (il faut commencer la fonction par une majuscule car `milieu` est une commande de `Xcas`).
On tape :
```
Milieu(A,B):=point((coordonnees(A)+coordonnees(B))/2)
```
ou encore si on a défini la fonction `Mediatrice` :
```
Milieu(A,B):=inter_unique(segment(A,B),Mediatrice(A,B))
```

1.5.3 Le barycentre

`isobarycentre` définit l'isobarycentre de n points.
On tape par exemple :
```
G:=isobarycentre(A,B,C,D,E)
```
ou
```
G:=isobarycentre([A,B,C,D,E])
```
et `G` est l'isobarycentre des points `A,B,C,D,E`.
`barycentre` définit le barycentre de n listes formées d'un point et d'un coefficient.
On tape par exemple :
```
K:=barycentre([A,2],[B,1],[C,-2],[D,3],[E,1])
```
ou
```
K:=barycentre([[A,2],[B,1],[C,-2],[D,3],[E,1]])
```
et `K` est le barycentre des points `A,B,C,D,E` affectés des coefficients `2,1,-2,3,1`.
Activités 1
Créer 4 points `A,B,C,D`.
Définir l'isobarycentre de `A,B,C,D`, en utilisant les coordonnées.
Réponse
On tape :
```
G:=point((coordonnees(A)+coordonnees(B)+coordonnees(C)+
coordonnees(D))/4)
```
On peut, comme exercice de programmation définir la fonction `Isobarycentre` (il faut commencer la fonction par une majuscule car `isobarycentre` est une commande de `Xcas`).
On tape :
```
Isobarycentre(L):={local d:=dim(L);point(sum(affixe(L[k]),k,0,d-1)/d)
```
ou bien
```
Isobarycentre(L):={local d:=dim(L);sum(L[k],k,0,d-1)/d;}
```

Créer 4 points `A,B,C,D`.
Définir le barycentre de `[A,1],[B,-2],[C,1],[D,3]`, en utilisant les coordonnées.
Réponse
On tape :
```
G:=point((coordonnees(A)-2*coordonnees(B)+coordonnees(C)+
```

`3*coordonnees(D))/3)`
On peut aussi comme exercice de programmation définir la fonction `Barycentre` (il faut commencer la fonction par une majuscule car `barycentre` est une commande de `Xcas`).
On tape :

```
Barycentre(L):={
local s,d:=dim(L);
  s:=sum(L[k],k,1,d-1,2);
  si (s==0) alors return "pas defini" fsi;
retourne sum(L[k+1]*L[k],k,0,d-2,2)/s;
}
```

Activité 2 : Partager un triangle en 3 triangles de même aire
Soit un triangle ABC.

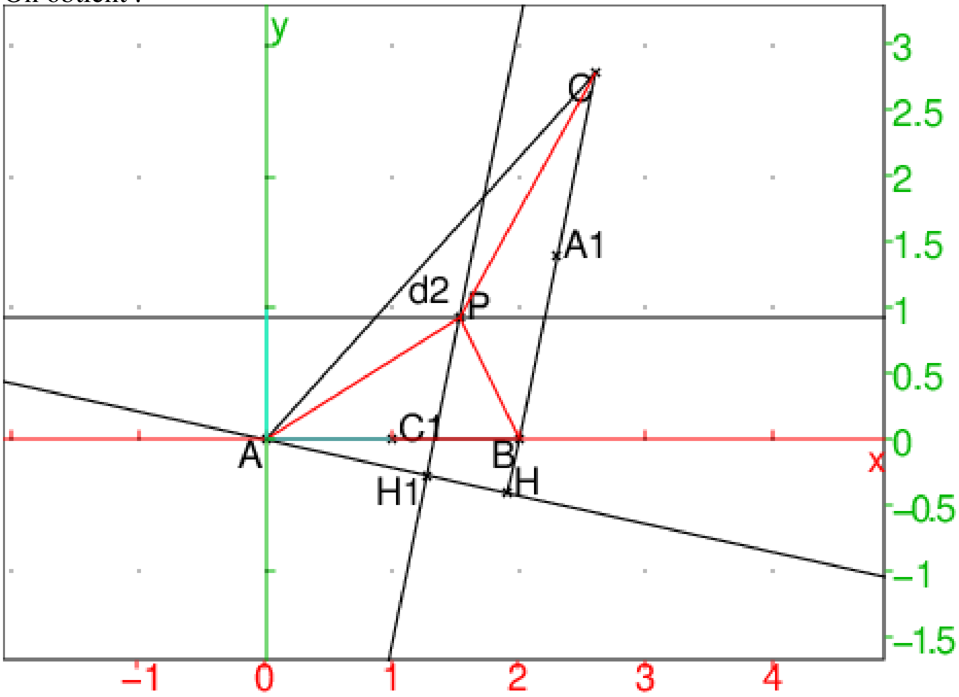
- On cherche un point P intérieur au triangle ABC tel que les aires des triangles ABP, BCP et CAP soient égales.
  Montrer que P est l'isobarycentre des points A, B, C.
- On cherche un point P tel que les aires des triangles ABP, BCP et CAP soient proportionnelles à 1,2,3. Définir P à l'aide d'un barycentre des points A, B, C.
- On cherche un point P tel que les aires des triangles ABP, BCP et CAP soient soient proportionnelles à n, p, q. Définir P à l'aide d'un barycentre des points A, B, C.
- prolongement
  Peut-on toujours trouver un point P intérieur au quadrilatère convexe $ABCD$ qui partage ce quadrilatère en 4 triangles de même aire ?
  A quelle condition cela est-il possible ?

Avec `Xcas`

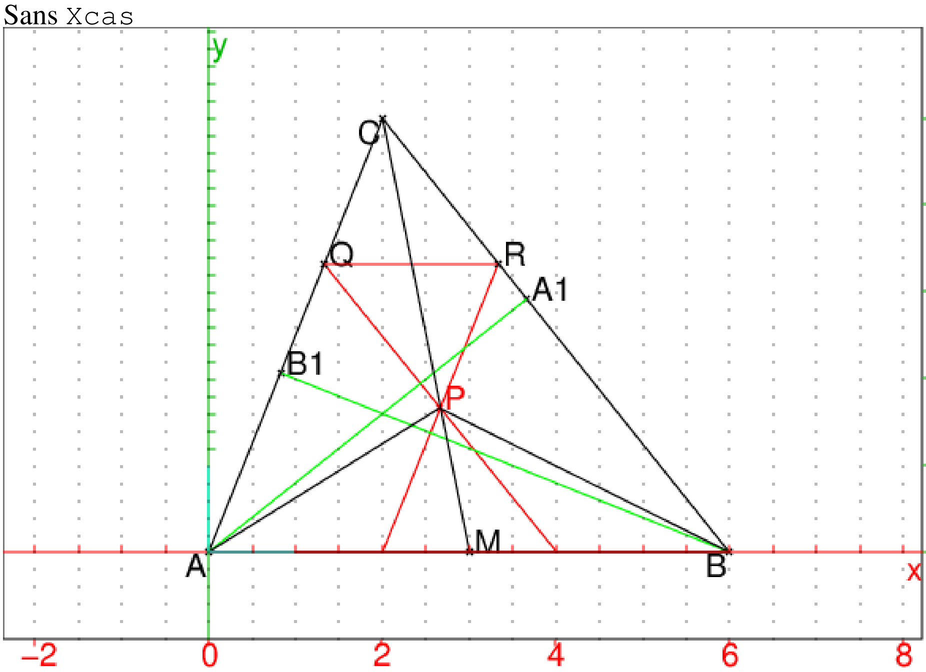
- On tape :
  ```
  supposons(a=[2.6,-5,5,0.1]);
  supposons(b=[2,0,5,0.1]);
  supposons(c=[2.8,0,5,0.1]);
  A:=point(0);
  B:=point(b);
  C:=point(a+i*c);
  triangle(A,B,C);
  H:=projection(droite(B,C),A);
  H1:=point((H-A)*2/3);
  h:=hauteur(A,B,C);
  d1:=perpendiculaire(H1,h);
  d2:=droite(y=c/3);
  P:=inter_unique(d1,d2);
  segment(P,A,affichage=1);
  segment(P,B,affichage=1);
  segment(P,C,affichage=1);
  A1:=inter_unique(droite(C,B),droite(A,P));
  C1:=inter_unique(droite(A,B),droite(C,P));
  ```

`G:=isobarycentre(A,B,C):;`

On obtient :



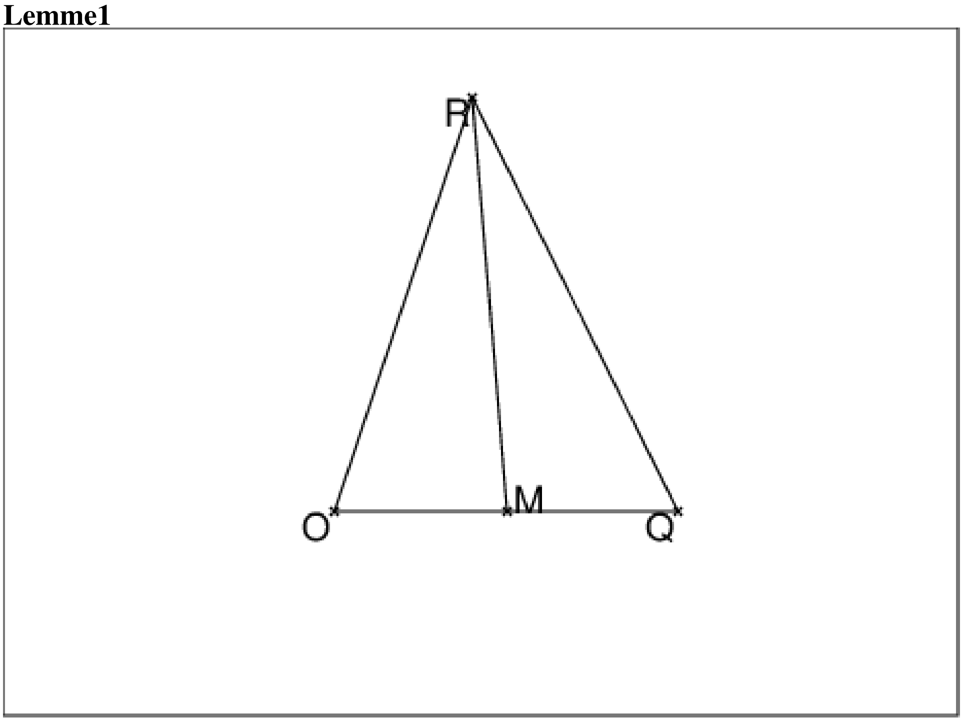
On trouve : P est le `point(1/3*a+1/3*b+(i)/3*c)`,
$A1$ est le `point(1/2*a+1/2*b+(i)/2*c)`
$C1$ est le `point(1/2*b)`
G est le `point((b+a+(i)*c)/3)`
Donc
$A1$ est le milieu de BC et $C1$ est le milieu de BC P est l'isobarycentre des points A, B, C.
Sans `Xcas`



Soient H_1, (resp H_2, H_3) les projections de P sur BC, (resp AC, AB), A_1,

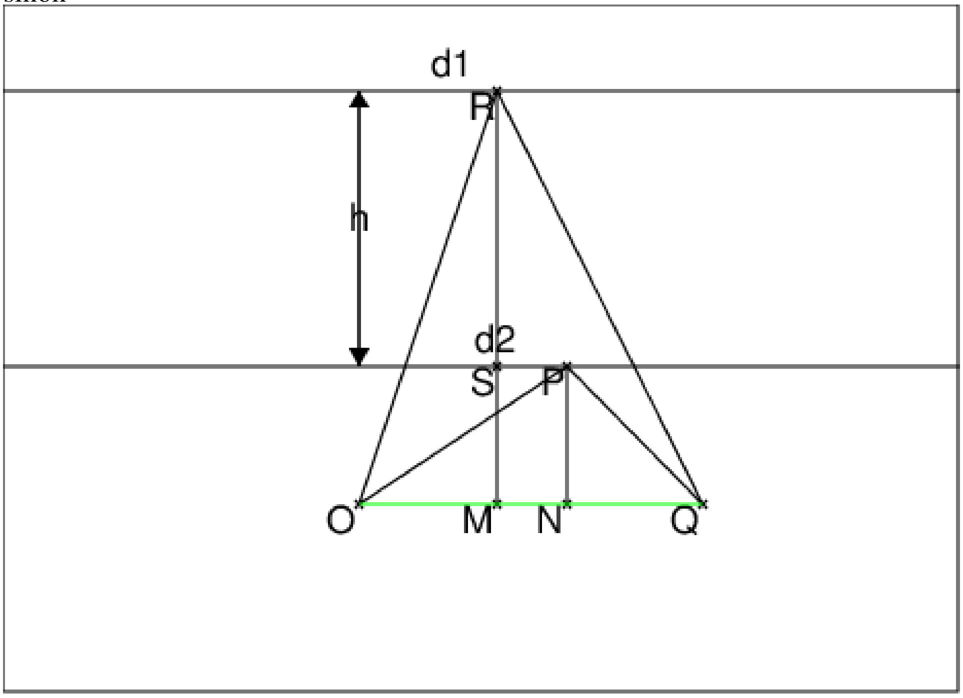
(resp B_1, C_1) les projections de A, (resp B, C) sur BC, (resp AC, AB) et P_1, P_2, P_3 les intersections de AP, (resp BP, CP) avec BC, (resp AC, AB). Si les aires des triangles ABP, BCP et CAP sont égales, c'est que $AA_1 = 3PH_1$, $BB_1 = 3PH_2$, $CC_1 = 3PH_3$. et donc que :
$AP_1 = 3PHP_1$, $BP_2 = 3PP_2$, $CP_3 = 3PP_3$.
Soient s_1 (resp Q) l'homothétique du segment BC (resp du point C) dans l'homothétie de centre A et de rapport 2/3 et
s_2 (resp R) l'homothétique du segment BC (resp du point C) dans l'homothétie de centre B et de rapport 2/3.
Q (resp R) est l'homothétique du point A (resp du point B) dans l'homothétie de centre C et de rapport 1/3.
Donc P est l'intersection des 2 segments s_1 et s_2 et
QR est parallèle à AB.
Le quadrilatère $PRCQ$ est un parallélogramme (ses côtés sont parallèles 2 à 2).
Donc CP passe par le milieu de QR et aussi par le milieu M de AB puisque QR est parallèle à AB.
Donc puisque $CP/CM = 2/3$ et que CM est une médiane du triangle ABC, P est le centre de gravité du triangle ABC.

- P, le barycentre de $C, 1$, $A, 2$, $B, 3$ répond à la question.
- P, le barycentre de C, n, A, p, B, q répond à la question.
- **Lemme1**

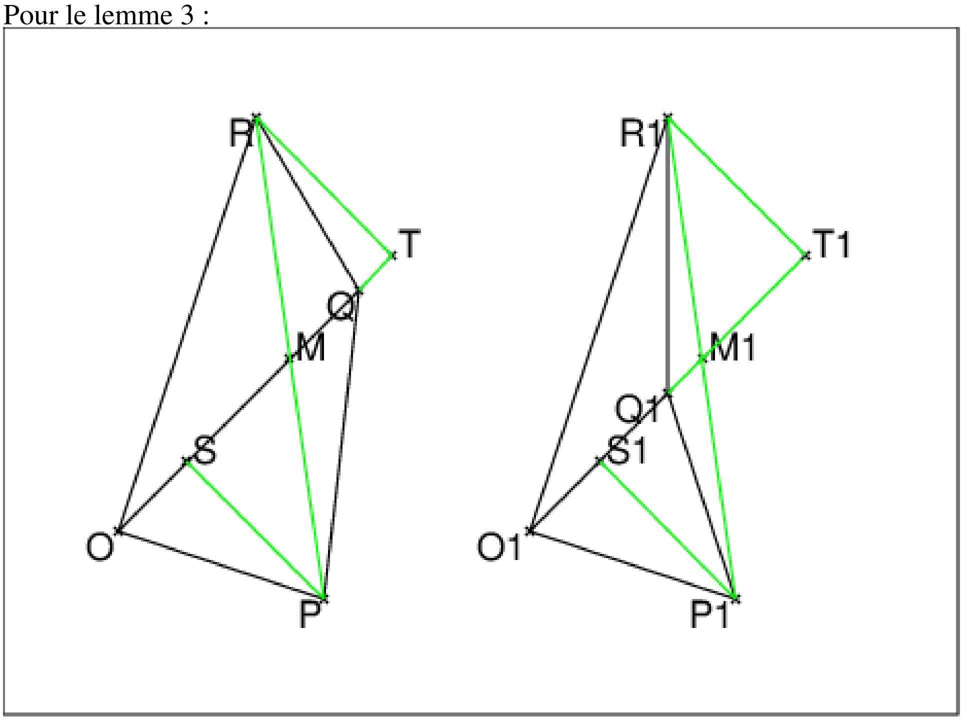


Soit un triangle OQR et un point M sur le segment OQ alors aire de OMR=aire de MQR est équivalent à M est le milieu de OQ **Lemme2**
Soit un quadrilatère $OPQR$.
On note les parallèles $d1$ et $d2$à OQ et h la distance entre $d1$ et $d2$, alors :
aire($OPQR$)=longueur(OQ)*$h/2$. **Lemme3** Soit un quadrilatère $OPQR$ alors :
aire(OPQ)=aire(OQR) est équivalent à OQ passe par le milieu de PR.

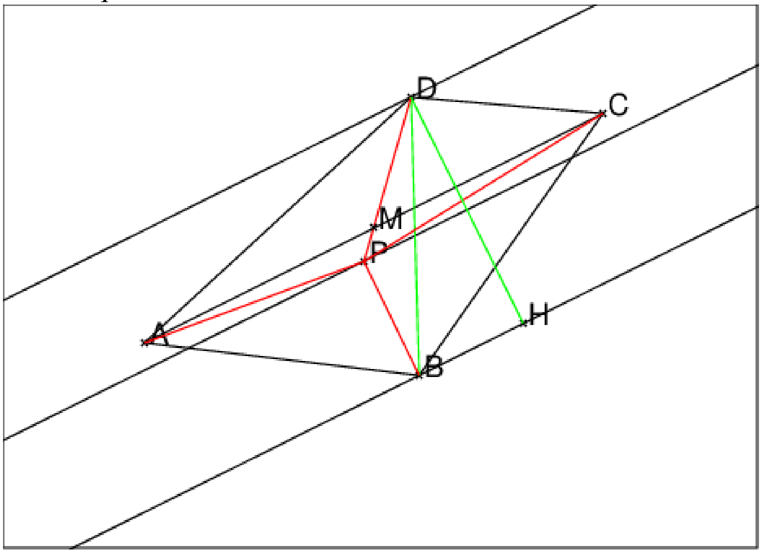
Les démonstrations sont èvidentes car l'aire d'un triangle est égale à : base*hauteur/2.
Pour le lemme 1 :
OPR (resp PQR) ont comme base OP (resp PQ) et la même hauteur.
Pour le lemme 2 :
évident si $OPQR$ est convexe.
sinon



aire($OPQR$)=aire(OQR)-aire(OPQ)=
longueur(OQ)*(longueur(RM)-longueur(PN)/2=longueur(OQ)*h/2.
Pour le lemme 3 :

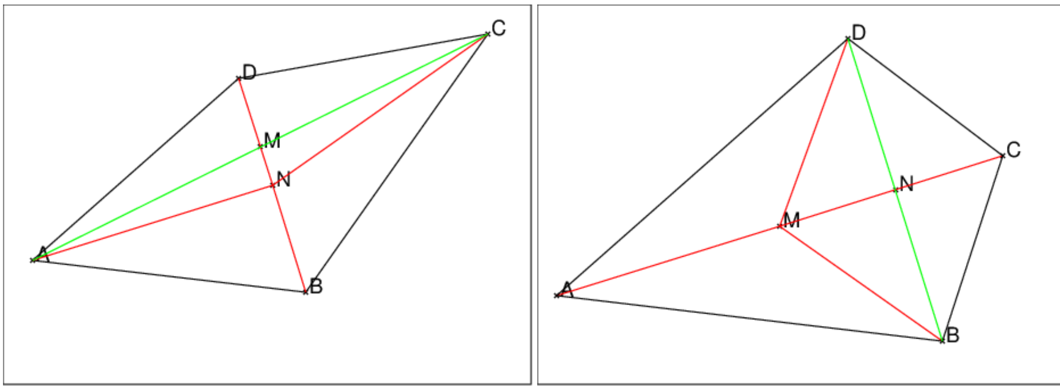


OPQ et OQR ont comme base OQ donc
aires égales est équivalent à même hauteur qui est équivalent à OQ passe par le milieu de PR.
Soit un quadrilatère convexe $ABCD$.



On cherche tout d'abord un point P tel que :
aire($ABCP$)=aire($APCD$).
On mène par B et D les parallèles $d1$ et $d2$à AC. D'après le `lemme 2` P se trouve sur la parallèle équidistante à $d1$ et $d2$.
On veut aussi que :
aire(ADP)=aire(CDP)
La base commune est DP, d'après le `lemme 3`, DP passe par le milieu M de AC i.e D, P, M sont aligǹes (c'est vrai sur la figure).
aire(ABP)=aire(CBP)
La base commune est BP, d'après le `lemme 3`, BP passe par le milieu M de AC i.e B, P, M sont aligǹes (ce n'est pas vrai sur la figure).
Donc, si les 4 aires sont égales, on a : P est en M ou P est différent de M et donc les points D, P, M, B sont alignés.
Si P est en M, P se trouve sur AC donc AC est la parallèle équidistante à $d1$ et $d2$. Donc AC passe par le milieu de BD.
Si P est différent de M, les points D, P, M, B sont alignés. Donc BD passe par le milieu M de AC et P est en N milieu de BD d'après le `lemme 1`
`En résumé` On peut un point P intérieur à un quadrilatère convexe $ABCD$ qui partage ce quadrilatère en 4 triangles de même aire si et seulement si l'une des diagonales de $ABCD$ passe par le milieu de l'autre diagonale : P est alors le milieu de la diagonale qui passe par le milieu de l'autre diago-

nale.



1.5.4 Bissectrice d'un angle

Étant donné trois points `A`, `B` et `C` les commandes :
`bissectrice(A,B,C)` trace la bissectrice intérieure de l'angle A du triangle ABC.
`exbissectrice(A,B,C)` trace la bissectrice extérieure de l'angle A du triangle ABC.
Activité
Créer un triangle `ABC`.
Construire les bissectrices de l'angle A du triangle ABC, en utilisant la même construction qu'avec un compas et en utilisant l'instruction `mediatrice`.
Réponse
On clique avec la souris pour avoir un triangle `ABC`.
On tape :
`C1:=cercle(A,2)` trace un cercle de centre `A` et de rayon `2`.
`D:=inter(C1,droite(A,B))` définit l'intersection du cercle `C1` et de la droite AB.
`E:=inter(C1,droite(A,C))` définit l'intersection du cercle `C1` et de la droite AC.
`mediatrice(D[0],E[0])`
`mediatrice(D[1],E[0])`
On peut aussi comme exercice de programmation définir les fonctions `Bissectrice` et `Exbissectrice` (il faut commencer les noms des fonctions par une majuscule car `bissectrice` et `exbissectrice` sont des commandes de `Xcas`).
On tape si on a défini la fonction `Mediatrice` :

```
Bissectrice(A,B,C):=Mediatrice(inter_unique(demi_droite(A,B),
cercle(A,2)),inter_unique(demi_droite(A,C),cercle(A,2)))
Exbissectrice(A,B,C):={local C1:=A+(A-C);Bissectrice(A,B,C1)}
```

1.5.5 Report d'une longueur

Étant donnés trois points `A` `B` et `C`, on veut construire un point `D` pour que $AD = BC$.
On utilise la commande `cercle` et on tape :
`D:= element(cercle(A,B-C))`
L'instruction `longueur(B,C)` renvoie la longueur du segment BC (les unités

étant définies par le choix de `WX-, WX+` et de `WY-, WY+`) effectué dans la fenêtre d'initialisation graphique.
Si l'on veut reporter une longueur dans une direction donnée, on multiplie cette longueur par le vecteur unitaire de cette direction.
Exemple :
Étant donnés trois points `A B` et `C`, construire sur la demi-droite AB, un point D tel que $AD = AC$.
On tape :
```
D:=A+longueur(A,C)*(B-A)/longueur(A,B)
```
ou encore
```
D:=inter_unique(cercle(A,C-A),demi_droite(A,B))
```

1.5.6 Report d'un angle

Étant donnés deux points `A` et `B`, on veut construire `C` pour que l'angle $(\overrightarrow{AB}, \overrightarrow{AC})$ soit de mesure donnée par exemple 72 degrés ou $2 * \pi/5$ radians.
On tape, si on a coché `radian` dans la fenêtre de configuration du `CAS` :
```
D:=rotation(A,2*pi/5,droite(A,B))
```
ou, si on est en degré (on n'a pas coché `radian`) :
```
D:=rotation(A,72,droite(A,B))
```
puis on tape :
```
C:=element(D)
```
L'instruction `angle(A,B,C)` donne la mesure en radians (ou en degrés) de l'angle $(\overrightarrow{AB}, \overrightarrow{AC})$, on peut donc vérifier la construction demandée.
Étant donné deux points `A` et `B`, on veut construire `C` pour que l'angle $(\overrightarrow{AB}, \overrightarrow{AC})$ soit égal à l'angle $(\overrightarrow{OM}, \overrightarrow{OP})$.
On tape :
```
D:=rotation(A,angle(O,M,P), droite(A,B));
C:=element(D)
```

1.5.7 Perpendiculaire à la droite D passant par A

Étant donné trois points `A B` et `C`, on veut construire la perpendiculaire, passant par A, à la droite BC.
On tape :
`perpendiculaire(A,droite(B,C))` ou,
`hauteur(A,B,C)` (voir aussi 1.16) ou,
on utilise le nombre complexe `i` : `droite(A,A+i*(C-B))`.
Activité
Créer un point `A` et une droite `BC` ne passant pas par `A`.
Construire la perpendiculaire à la droite BC passant par A, en utilisant la même construction qu'avec un compas.
Réponse
On clique avec la souris pour avoir un point `A` et deux points `B` et `C` et le segment `BC`.
On tape :
`C1:=cercle(B,A-B)` trace le cercle de centre B passant par A
`C2:=cercle(C,A-C)` trace le cercle de centre C passant par A

`droite(inter(C1,C2))` trace la droite qui joint les points d'intersection des deux cercles précédents.
Activité
Créer un segment `AB`.
Construire la perpendiculaire à la droite AB passant par A en utilisant la même construction qu'avec un compas.
Réponse
On clique avec la souris pour avoir deux points `A` et `B` et le segment `AB`.
On tape :
`C3:=cercle(A,B-A)` trace le cercle de centre A passant par B
`mediatrice(inter(C3,droite(A,B)))` trace la mediatrice des deux points définit par `inter(C3,droite(A,B))`, cette médiatrice passe par A et est perpendiculaire à AB.
On peut aussi comme exercice de programmation définir la fonction `Perpendiculaire` (il faut commencer le nom de la fonction par une majuscule car `perpendiculaire` est une commande de `Xcas`).
On tape (`A` est un point et `d` est une droite) :

```
Perpendiculaire(A,d):={
local L,M,E:=element(d);
si est_element(A,d) alors
  L:=inter(d,cercle(A,1));
  retourne simplify(affixe(A)),simplify(equation(mediatrice(L)));
fsi;
si angle(E,d,A)==pi/2 or angle(E,d,A)==-pi/2  alors
  retourne simplify(affixe(E)),equation(droite(A,E));
fsi;
M:=milieu(op(inter(d,cercle(A,E-A))));
retourne simplify(affixe(M)),equation(droite(A,M));
}:;
```

ainsi `Perpendiculaire` renvoie l'affixe de la projection orthogonale `E` de `A` sur `d` et l'équation de la droite `A,E`.
On tape :
`Perpendiculaire(point(1),droite(0,1+i))`
On obtient :
`i/2+1/2,y=(-x+1)`
On tape :
`Perpendiculaire(point(1),droite(-2,1+i))`
On obtient :
`i/2-1/2,y=(-3*x+3)`

1.5.8 Parallèle à une droite passant par A

Étant donné trois points `A B` et `C`, on veut construire la parallèle à la droite BC passant par A.
On tape :
`parallele(A,droite(B,C))`
Activité

Créer un point A et un segment BC ne passant pas par A. Construire la parallèle à la droite BC passant par A en utilisant l'instruction perpendiculaire.
Réponse
On clique avec la souris pour avoir un point A et deux points B et C et le segment BC.
On tape :
`D:=perpendiculaire(A,droite(B,C))`
cela trace la perpendiculaire à BC passant par A
`P:=perpendiculaire(A,D)`
cela trace la perpendiculaire à D passant par A.
La liste des instructions se trouve dans geo5 : créer trois points A, B, C, puis faire Charger session du menu Fich de Xcas et selectionner geo5 du répertoire examples/geo pour exécuter ce fichier.

On peut, comme exercice de programmation définir la fonction Parallele (il faut commencer le nom de la fonction par une majuscule car parallele est une commande de Xcas).
On tape (A est un point et d est une droite) :

```
Parallele(A,d):={
local d1:=droite(Perpendiculaire(A,d)[1]);
Perpendiculaire(A,d1)[1];
}:;
```

Parallele renvoie l'équation de la parallèle à d passant par A.
On tape :
`Parallele(point(i),droite(0,1+i))`
On obtient :
`y=(x+1)`

1.5.9 Parallèles à une droite situées à une distance d de cette droite

Activité
Créer un point A et un segment BC ne passant pas par A.
Puis créer un point D pour définir d =longueur(A,D).
Construire les parallèles à la droite BC situées à une distance d =longueur(A,D) de BC.
Réponse
On clique avec la souris pour avoir un point A et deux points B et C et le segment BC, puis on clique avec la souris pour avoir un point D ($d = longueur(A, D)$).
On tape :

```
D1:=perpendiculaire(B,droite(B,C));
C1:=cercle(B,longueur(A,D));
I:=inter(D1,C1);
E:=I[0];
F:=I[1];
```

ou on utilise les nombres complexes pour définir E et F :
`E:=B+i*(C-B)*longueur(A,D)/longueur(B,C)` le point E est à une distance $d = longueur(A, D)$ de la droite BC,

`F:=B-i*(C-B)*longueur(A,D)/longueur(B,C)` le point `F` est à une distance $d = longueur(A, D)$ de la droite BC (`E` et `F` sont symètriques par rapport à BC),
puis,
`parallele(E,droite(B,C))` trace une parallèle à la droite BC situées à une distance $d = longueur(A, D)$ de BC.
`parallele(F,droite(B,C))` trace l'autre parallèle à la droite BC situées à une distance $d = longueur(A, D)$ de BC.

1.5.10 Tangentes à un cercle

Étant donné un point `A` et un cercle `C`, la commande :
`tangent(C,A)` dessine les deux tangentes à `C` passant par `A` si le point `A` est extérieur au cercle.
Si le point `A` est extérieur au cercle, `tangent(C,A)` est une liste de deux droites (les deux tangentes à `C` passant par `A`) et
si le point `A` est sur le cercle `tangent(C,A)` est une droite (la tangente à `C` passant par `A`).
Activité
Tracer les tangentes à un cercle passant par un point.
Cette activité doit se faire sans utiliser la commande `tangent`.
Créer deux points `A` et `B`.
Créer un cercle `C1` de centre `A` et passant par `B`.
Construire la tangente au cercle `C1` passant par `B`.
Créer un point `C` extérieur au cercle `C1` et construire les tangentes au cercle `C1` passant par `C`.
Réponse
On clique avec la souris pour avoir trois points `A`, `B` et `C` puis on exécute la liste des instructions qui se trouve dans `geo7` (faire `Charger session` du menu `Fich` de `Xcas` et selectionner `geo7` du répértoire `examples/geo` pour exécuter ce fichier).
Voici le détail de `geo7` :
`C1:=cercle(A,B-A)` ($C1$ est le cercle de centre A qui passe par B),
`perpendiculaire(B,droite(A,B))` (on dessine la tangente à $C1$ au point B),
`C2:=cercle(C,A)` ($C2$ est le cercle de diamètre AC),
`E:=inter(C1,C2)` (E désigne les 2 points d'intersection de $C1$ et de $C2$),
`droite(E[0],C)` (c'est une tangente à $C1$ passant par C),
`droite(E[1],C)` c'est l'autre tangente à $C1$ passant par C,
Activité
Tracer un cercle tangent à une droite donnée en un point donné et passant par un autre point.
Créer trois points `A`, `B` et `C`.
Tracer la droite `D` passant par `A` et `B`.
Construire un cercle `C1` passant par `C` et tangent à `D` en `A`.
Réponse
On clique avec la souris pour avoir trois points `A`, `B` et `C` puis on exécute la liste des

instructions qui se trouve dans `geo8` (faire `Charger session` du menu `Fich` de `Xcas` et selectionner `geo8` du répértoire `examples/geo` pour exécuter ce fichier).
Voici le détail de `geo8` :
`D:=droite(A,B)` (trace la droite D passant par A et B),
`P:=perpendiculaire(A,D)` (trace la perpendiculaire à D passant par A),
`M:=mediatrice(A,C)` (trace la médiatrice de AC),
`N:=inter(P,M)[0]` (N est le point d'intersection des deux droites précédentes),
`C1:=cercle(N,A-N)` (trace le cercle de centre N passant par A).
`C1` repond à la question.

1.5.11 Tangentes communes à 2 cercles

Soient `C1:=cercle(A,r1)` et `C2:=cercle(B,r2)`. Supposons `r1!=r2` alors :

- si `longueur(A,B)<|r2-r1|` il n'y a pas de tangente commune.
- si `longueur(A,B)=|r2-r1|` les 2 cercles sont tangents en `C` et il y a 1 tangente commune passant par `C` et perpendiculaire à `AB`.
- si `|r2-r1|<longueur(A,B)<r2+r1` il y a 2 tangentes communes extérieures qui se coupent en `C` (`CA/CB=r1/r2`).
- si `longueur(A,B)=r2+r1` les 2 cercles sont tangents en `D` et il y a 1 tangente commune passant par `D` et perpendiculaire à `AB` et 2 tangentes communes qui se coupent en `C` (`CA/CB=r1/r2` et `DA=r1, DB=r2`).
- si `r2+r1<longueur(A,B)` il y a 2 tangentes communes extérieures qui se coupent en `C` et 2 tangentes communes intérieures qui se coupent en `D` (`CA/CB=r1/r2` et `DA/DB=r1/r2` avec `D` entre `A` et `B`).

Si `r1=r2` et `A!=B` il y a 2, 3, 4 tangentes communes et si `r1=r2` et `A=B` une infinité de tangentes communes Voici le programme général :

```
tangentes(A,r1,B,r2):={
local C1,C2,C,D,p,L,t1,t2;
C1:=cercle(A,r1);
C2:=cercle(B,r2);
si r1==0 alors
return tangente(C2,A);
fsi;
si r2==0 alors
return tangente(C1,B);
fsi;
si longueur2(A,B)<(r2-r1)^2 alors
return [];
fsi;
si r1==r2 alors
si A==B alors return [infinity]; fsi;
p:=perpendiculaire(B,droite(A,B));
L:=inter(C2,p);
t1:=parallele(L[0],droite(A,B));
t2:=parallele(L[1],droite(A,B));
```

```
si longueur2(A,B)<(r2+r1)^2 alors
return [t1,t2];
sinon
return concat([t1,t2],tangente(C1,milieu(A,B)));
fsi;
fsi;
C:=division_point(A,B,r1/r2);
si longueur2(A,B)<(r2+r1)^2 alors
return tangente(C1,C);
sinon
D:=division_point(A,B,-r1/r2);
return concat(tangente(C1,C),tangente(C1,D));
fsi;
}:;
```

Activité Tracer les tangentes extérieures communes à deux cercles sécants ou extérieurs avec une construction géométrique niveau 3ième sans utiliser la primitive `tangente`.
On suppose que si `C1:=cercle(A,r1)` et `C2:=cercle(B,r2)` alors :
`longueur(A,B)>=|r2-r1|`.

— Si les 2 cercles ont des rayons égaux, la solution est facile car les tangentes extérieures communes sont parallèles à la droite des centres et les points de contact sont sur les perpendiculaires à la droite des centres passant respectivement par les centres.
On tape :

```
A:=point([-3.5,0.5],affichage=4);
c1:=cercle(A,4,affichage=4);
B:=point([1.3,1.4],affichage=4);
c2:=cercle(B,4,affichage=4);
p:=perpendiculaire(B,droite(A,B),affichage=ligne_tiret_point);
L:=inter(c2,p);
t1:=parallele(L[0],droite(A,B),affichage=1);
t2:=parallele(L[1],droite(A,B),affichage=1);
```

On obtient :

— Si les 2 cercles ont des rayons différents, par exemple si :
`C1:=cercle(A,r1)` et `C2:=cercle(B,r2)` avec `r1<r2`.
Si `K1` et `K2` sont les points de contact d'une tangente extérieure commune à `C1` et `C2`, le quadrilatère $A, B, K2, K1$ est un trapèze rectangle que l'on peut partager en un rectangle $A, R, K2, K1$ et un triangle rectangle A, B, R.

Donc R se trouve sur le cercle de diamètre AB et sur le cercle de centre B et de rayon $r2 - r1$.
On tape :

```
A:=point([-3.5,0.5],affichage=4);
```

```
r1:=1.5;
C1:=cercle(A,r1,affichage=4);
B:=point([1.3,1.4],affichage=4);
r2:=4;
C2:=cercle(B,r2,affichage=4);
C3:=cercle(B,r2-r1);
C4:=cercle(A,B);
L:=inter(C3,C4);
T1:=inter_unique(demi_droite(B,L[0]),C2);
T2:=inter_unique(demi_droite(B,L[1]),C2);
t1:=parallele(T1,droite(A,L[0]),affichage=1);
t2:=parallele(T2,droite(A,L[1]),affichage=1);
segment(B,T1,affichage=ligne_tiret_point);
segment(B,T2,affichage=ligne_tiret_point);
segment(A,B);
```

On obtient :

— Avec un programme :
Si `C1:=cercle(A,r1)` et `C2:=cercle(B,r2)` alors on doit avoir :
`longueur2(A,B)>=(r2-r1)^2`.
On tape dans l'éditeur de programmes :

```
tangente2c(A,r1,B,r2):={
local C1,C2,C3,C4,L,T1,T2,t1,t2,r,C,p;
si r1>r2 alors
r:=r1;r1:=r2;r2:=r;
C:=A;A:=B;B:=C;
fsi;
C1:=cercle(A,r1);
C2:=cercle(B,r2);
si longueur2(A,B)<(r2-r1)^2 alors
```

```
return [C1,C2];
fsi;
si r1==r2 alors
p:=perpendiculaire(B,droite(A,B));
L:=inter(C2,p);
t1:=parallele(L[0],droite(A,B),affichage=1);
t2:=parallele(L[1],droite(A,B),affichage=1);
return [C1,C2,t1,t2];
fsi;
C3:=cercle(B,r2-r1);
C4:=cercle(A,B);
L:=inter(C3,C4);
T1:=inter_unique(demi_droite(B,L[0]),C2);
T2:=inter_unique(demi_droite(B,L[1]),C2);
t1:=parallele(T1,droite(A,L[0]),affichage=1);
t2:=parallele(T2,droite(A,L[1]),affichage=1);
return [C1,C2,t1,t2]
}:;
```

Activité Tracer les tangentes intérieures communes à deux cercles extérieurs.
Si les 2 cercles sont extérieurs, par exemple si :
`C1:=cercle(A,r1)` et `C2:=cercle(B,r2)`, on doit avoir `AB>r1+r2`.
Si `K1` et `K2` sont les points de contact d'une tangente intérieure commune à `C1` et `C2`, la parallèle à $K1K2$ menée par A coupe $BK2$ en R. On a $BR = r2 + r1$ et le triangle A, B, R est rectangle en R.

Donc R se trouve sur le cercle de diamètre AB et sur le cercle de centre B et de rayon $r2 - r1$.

— On tape dans un niveau de géométrie 2D :

```
A:=point([-3.5,0.5],affichage=4);
r1:=1.5;
C1:=cercle(A,r1,affichage=4);
B:=point([3,2],affichage=4);
r2:=4;
C2:=cercle(B,r2,affichage=4);
C3:=cercle(B,r2+r1);
C4:=cercle(A,B);
L:=inter(C3,C4);
T1:=inter_unique(demi_droite(B,L[0]),C2);
T2:=inter_unique(demi_droite(B,L[1]),C2);
t1:=parallele(T1,droite(A,L[0]),affichage=1);
t2:=parallele(T2,droite(A,L[1]),affichage=1);
segment(B,L[0],affichage=ligne_tiret_point);
segment(B,L[1],affichage=ligne_tiret_point);
segment(A,B);
```

On obtient :

— Avec un programme :
Si `C1:=cercle(A,r1)` et `C2:=cercle(B,r2)` alors on doit avoir :
`longueur2(A,B)>=(r2-r1)^2`.
On tape dans l'éditeur de programmes :

```
tangenti2c(A,r1,B,r2):={
local C1,C2,C3,C4,L,T1,T2,t1,t2,r,C,p;
si r1>r2 alors
r:=r1;r1:=r2;r2:=r;
C:=A;A:=B;B:=C;
fsi;
C1:=cercle(A,r1);
C2:=cercle(B,r2);
```

```
    si longueur2(A,B)<(r2+r1)^2 alors
    return [C1,C2];
    fsi;
    si longueur2(A,B)<(r2+r1)^2 alors
    return [C1,C2];
    fsi;
    C3:=cercle(B,r2+r1);
    C4:=cercle(A,B);
    L:=inter(C3,C4);
    T1:=inter_unique(demi_droite(B,L[0]),C2);
    T2:=inter_unique(demi_droite(B,L[1]),C2);
    t1:=parallele(T1,droite(A,L[0]),affichage=1);
    t2:=parallele(T2,droite(A,L[1]),affichage=1);
    return [C1,C2,t1,t2]
    }:;
```

Voici le programme général :

```
tangentei2c(A,r1,B,r2):={
local C1,C2,C3,C4,C5,L,L1,T1,T2,T3,T4,t1,t2,t3,t4,r,C,p;
si r1>r2 alors
r:=r1;r1:=r2;r2:=r;
C:=A;A:=B;B:=C;
fsi;
C1:=cercle(A,r1);
C2:=cercle(B,r2);
si longueur2(A,B)<(r2-r1)^2 alors
return [C1,C2];
fsi;
C4:=cercle(A,B);
si longueur2(A,B)>=(r2+r1)^2 alors
C5:=cercle(B,r2+r1);
L1:=inter(C5,C4);
T3:=inter_unique(demi_droite(B,L1[0]),C2);
T4:=inter_unique(demi_droite(B,L1[1]),C2);
t3:=parallele(T3,droite(A,L1[0]),affichage=1);
t4:=parallele(T4,droite(A,L1[1]),affichage=1);
fsi;
si r1==r2 alors
si A==B alors return [C1,C2,infinity]; fsi;
p:=perpendiculaire(B,droite(A,B));
L:=inter(C2,p);
t1:=parallele(L[0],droite(A,B),affichage=1);
t2:=parallele(L[1],droite(A,B),affichage=1);
si longueur2(A,B)<(r2+r1)^2 alors
return [C1,C2,t1,t2];
sinon return [C1,C2,t1,t2,t3,t4];
fsi;
fsi;
C3:=cercle(B,r2-r1);
```

```
L:=inter(C3,C4);
T1:=inter_unique(demi_droite(B,L[0]),C2);
T2:=inter_unique(demi_droite(B,L[1]),C2);
t1:=parallele(T1,droite(A,L[0]),affichage=1);
t2:=parallele(T2,droite(A,L[1]),affichage=1);
si longueur2(A,B)<(r2+r1)^2 alors
return [C1,C2,t1,t2]
sinon return [C1,C2,t1,t2,t3,t4];
fsi;

return [C1,C2,t1,t2,t3,t4];
}:;
```

On tape :
`tangentei2c(point(0,0),1,point(-4,0),2);`
On obtient :

On tape :
`tangentei2c(point(0,0),1,point(-3,0),2);`
On obtient :

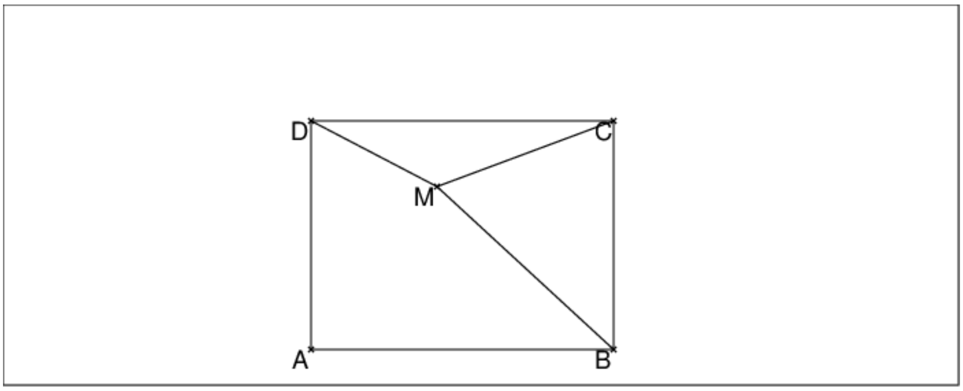
On tape :
`tangentei2c(point(0,0),1,point(-2,0),2);`
On obtient :

On tape :
`tangentei2c(point(0,0),1,point(-1,0),2)`
On obtient :

On tape :
`tangentei2c(point(0,0),1,point(-0.5,0),2)`
On obtient :

1.6 Une construction plus difficile

Soit un rectangle $ABCD$ tel que $AB = a$ et $AD = b$.
On cherche les conditions que doivent vérifier x et y pour que l'on puisse construire un point M, de coordonnèes (x, y), à l'intérieur du rectangle qui vérifie :
$MD = 3$, $MC = 4$ et $MB = 5$.

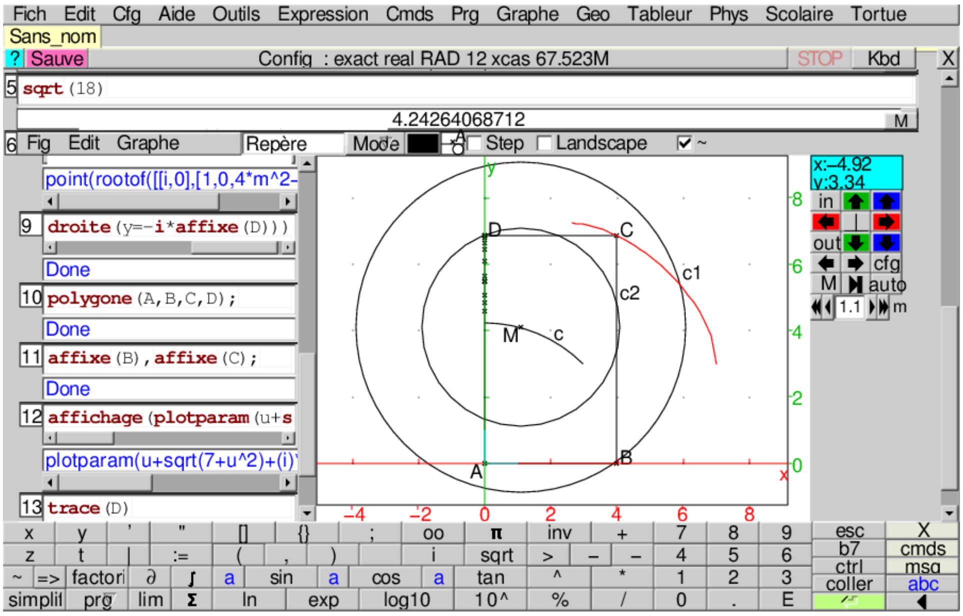


Si l'affixe de M est $m + i * n$, écrire les relations qui lient m, n, a, b.
Faire la construction avec `Xcas`.

On a donc :
$0 \leq m \leq 3$ et $0 \leq n \leq 5$
On pose :
```
M:=point(m+i*n)
A:=point(0)
B:=point(a)
C:=point(a+i*b)
D:=point(i*b)
```
On tape :
```
longueur2(M,D)-9
```
On obtient :
```
(m)^2+(-b+n)^2-9
```
On tape :
```
longueur2(M,C)-16
```
On obtient :
```
(-a+m)^2+(-b+n)^2-16
```
On tape :
```
longueur2(M,B)-25
```
On obtient :
```
n^2+(-a+m)^2-25
```
Donc m, n, a, b vérifient les équations :
$[(m)^2 + (-b + n)^2 = 9, (-a + m)^2 + (-b + n)^2 = 16, n^2 + (-a + m)^2 = 25]$
Donc :
$m^2 + n^2 = (m)^2 + (-b + n)^2 - ((-a + m)^2 + (-b + n)^2) + n^2 + (-a + m)^2 = 9 - 16 + 25 = 18$
Le point M se trouve donc sur le cercle c de centre A et de rayon $\sqrt{18}$. Comme $0 \leq m \leq 3$ et que le point d'affixe $3 + 3 * i$ est sur le cercle c, on peut dire que M est sur l'arc $\pi/4, \pi/2$ ($0 \leq n \leq 5$ sera vérifié car $\sqrt{18} < 5$).
On tape :

```
A:=point(0);
c:=cercle(0,3*sqrt(2),pi/4,pi/2);
supposons(m=[2.7,0,3,0.1]);
M:=point(m+i*sqrt(18-m^2));
c1:=cercle(M,5);
```

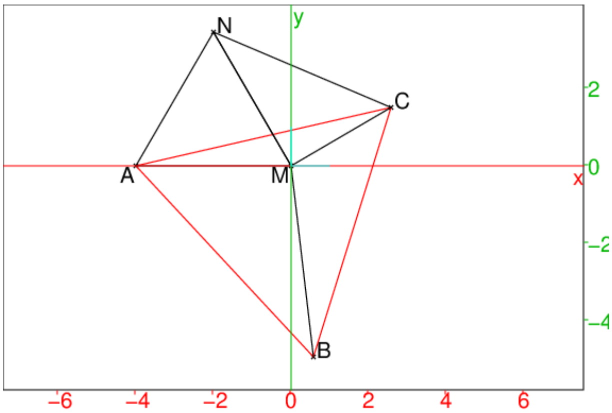
```
c2:=cercle(M,3);
B:=inter_unique(droite(y=0),c1);
D:=inter_unique(droite(x=0),c2);
C:=inter_unique(droite(x=affixe(B)),droite(y=-i*affixe(D)));
polygone(A,B,C,D);
affixe(B),affixe(C);
affichage(plotparam(u+sqrt(7+u^2)+i*(sqrt(18-u^2)+sqrt(9-u^2)),
                    u=0..sqrt(9),tstep=0.1),1);
trace(D);
```



Remarque
Le problème est plus facile si on modifie les conditions.
On cherche les conditions que doivent vérifier x et y pour que l'on puisse construire un point M, de coordonnèes (x, y), à l'intérieur du rectangle vérifiant :
$MD = 3$, $MC = 5$ et $MB = 4$.
On a alors :
$x^2 + (y - b)^2 = 9$ $(x - a)^2 + (y - b)^2 = 25$ $(x - a)^2 + y^2 = 16$ Donc :
$x^2 + (y - b)^2 - (x - a)^2 - (y - b)^2 + (x - a)^2 + y^2 = x^2 + y^2 = 0$ Donc $x = 0$, $y = 0$ et M se trouve en A.
Le rectangle a alors comme dimension $a = 4$ et $b = 3$.

1.7 Une autre construction avec un triangle équilatéral

Soit un triangle équilatéral ABC tel que $AB = a$.
On cherche la valeur de a pour que l'on puisse construire un point M à l'intérieur du triangle ABC tel que $MA = 4$, $MB = 5$ et $MC = 3$.
Faire la construction de ABC et calculer les aires des 4 triangles :
ABC, ABM, ACM, BCM.



On pourra effectuer la rotation de centre A et d'angle $\pi/3$ qui transforme M en N.

La solution

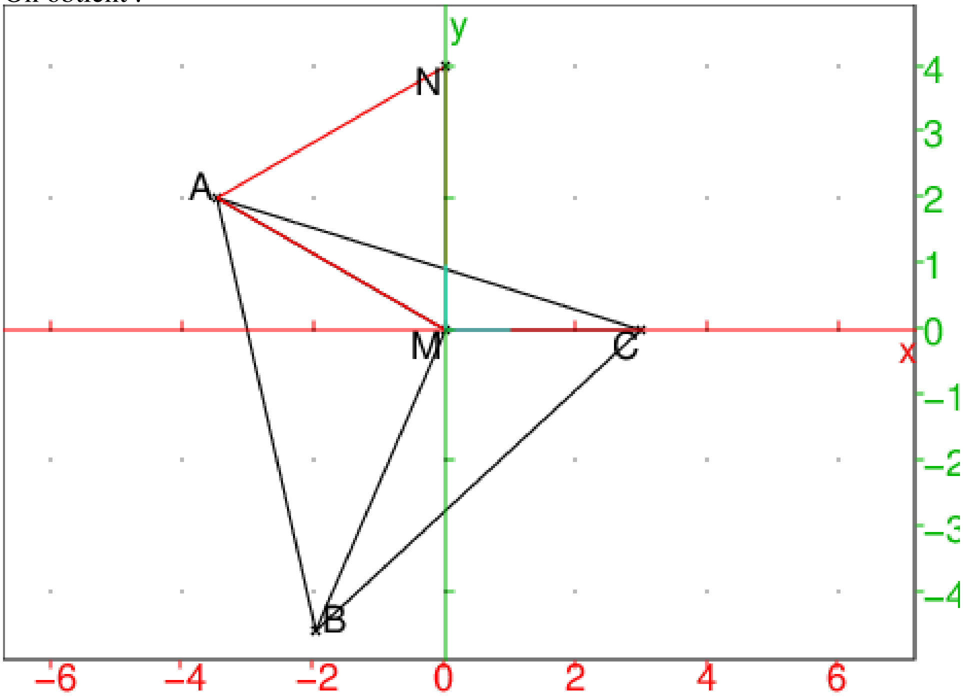
Considérons la rotation de centre A et d'angle $\pi/3$ qui transforme M en N.
Le triangle AMN est isocèle de sommet A et on a :
$\widehat{MAN} = \widehat{MAC} + \widehat{CAN} = \widehat{MAC} + \widehat{BAM} = \widehat{BAC} = \pi/3$
Donc le triangle AMN est équilatéral et $MN = 4$.
Le triangle CNM a donc pour côtés : $CM = 3$, $MN = 4$, $CN = 5$.
Le triangle CNM est donc rectangle en M. et l'angle $\widehat{CMA} = 5\pi/6$.
On a donc puisque $\cos(5\pi/6) = -\sqrt{3}/2$, $MC = 3$ et $AM = 4$:
$a^2 = AC^2 = 3^2 + 4^2 - 2 * 3 * 4 * (-\sqrt{3}/2) = 25 + 12\sqrt{3}$.

La construction

On tape :

```
C:=point(3);
A:=point(-4*sqrt(3)/2+4*i/2,affichage=quadrant2);
B:=rotation(C,pi/3,A);
triangle(A,B,C);
M:=point(0);
triangle_equilateral(A,M,N,affichage=1);
segment(B,M);
segment(C,M);
```

On obtient :



On vérifie, on tape :

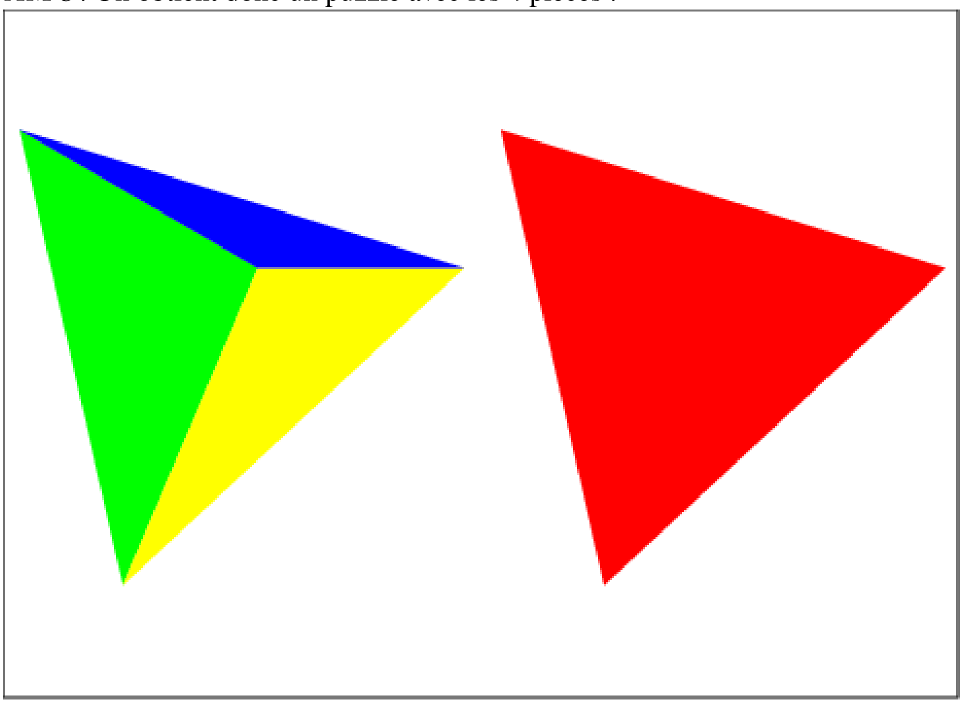
`normal(longueur(A,M),longueur(B,M))`

On obtient :

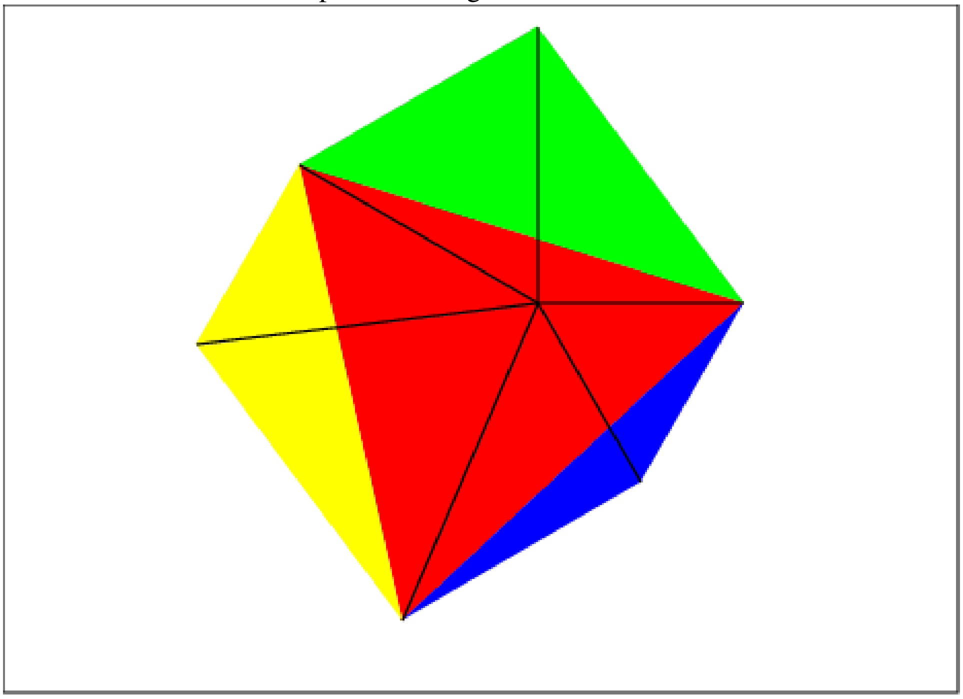
`4,5`

Calcul des aires à l'aide d'un puzzle

On considère 2 exemplaires du triangle équilatéral ABC défini ci-dessus. On découpe un des exemplaires en 3 morceaux qui sont les triangles ABM, BCM et AMC. On obtient donc un puzzle avec les 4 pièces :



On réalise alors avec ces 4 pièces l'octogone :



On a tracé les segments joignant M au sommet de l'octogone.
Cet octogone est compse de 3 triangles rectangles de côtés 3,4,5 et de 3 triangles èquilatéraux de côtés respectifs 3,4 et 5. On a donc si S est l'aire de ABC :
$2S = a^2\sqrt{3}/2 = 3 * 3 * 2 + (9 + 16 + 25)\sqrt{3}/4 = 18 + 25\sqrt{3}/2$
Donc $S = 9 + 25\sqrt{3}/4$ et on retrouve que $a^2 = 25 + 12\sqrt{3}$
Soient s_1, s_2, s_3, s_4 les aires des triangles de couleurs 1 (rouge),2(vert), 3(jaune),4(bleu).
On va donc résoudre le système linéaire :
$[s_1 - s_2 - s_3 - s_4 = 0, s_2 + s_3 = 6 + 25\sqrt{3}/4, s_2 + s_4 = 9\sqrt{3}/4 + 6, s_3 + s_4 = 4\sqrt{3} + 6]$
On tape :
```
linsolve([s1-s2-s3-s4=0,s2+s3=6+25*sqrt(3)/4,
s2+s4=9*sqrt(3)/4+6,s3+s4=4*sqrt(3)+6],[s1,s2,s3,s4])
```
On obtient :
```
[(sqrt(3)*25+36)/4,(sqrt(3)*9+12)/4,sqrt(3)*4+3,3]
```
On a tapé pour obtenir l'octogone :

```
triangle(A,B,C,affichage=1+rempli);
triangle(A,C,N,affichage=2+rempli);
P:=rotation(C,pi/3,M):;
Q:=rotation(B,pi/3,M):;
triangle(B,C,P,affichage=4+rempli))
triangle(B,A,Q,affichage=3+rempli);
```

1.8 Les transformations

Les transformations ci-dessous peuvent toujours être considérées, soit comme des fonctions (les arguments sont les paramètres servant à définir la transformation), soit comme agissant sur le dernier argument (les arguments sont les paramètres servant à définir la transformation et alorsl'objet géométrique à transformer est mis comme dernier paramètre).

1.8.1 La translation

Étant donné trois points `A, B, C` le point :
`translation(C-B,A)` est le transformé du point `A` dans la translation de vecteur $\overrightarrow{BC}$.
On a aussi `t:=translation(C-B)` est la translation de vecteur $\overrightarrow{BC}$.
On écrira alors `t(A)` pour désigner le transformé du point `A` dans la translation `t` de vecteur $\overrightarrow{BC}$.
Par exemple si :
`D:=t(A)` ou si
`D:=translation(C-B,A)` on a :
$\overrightarrow{AD} = \overrightarrow{BC}$.
Activité
Créer un polygone quelconque `ABCDE`.
Créer un point `F` à l'extérieur de ce polygone.
Construire le polygone `FGHJK` translaté de `ABCDE` dans la translation de vecteur $\overrightarrow{AF}$.
Faites bouger le point `A` et observer.
Faites bouger le point `F` et observer.

1.8.2 L'homothétie

Étant donné deux points `A, B` et un nombre réel k, le point :
`homothetie(B,k,A)` est le transformé du point `A` dans l'homothétie de centre `B` et de rapport k.
On a aussi `h:=homothetie(B,k)` est l'homothétie de centre `B` et de rapport k.
Par exemple si :
`C:=h(A)` ou si
`C:=homothetie(B,2,A)`, `C` est tel que $\overline{\texttt{BC}} = 2 * \overline{\texttt{BA}}$
Activité
Créer un polygone quelconque `ABCDE`.
Créer un point `F` à l'extérieur de ce polygone.
Construire le polygone `GHJKL` homothétique de `ABCDE` dans l'homothétie de vecteur centre `F` et de rapport 2.
Faites bouger le point `A` et observer.
Faites bouger le point `F` et observer.

1.8.3 La symétrie droite et la symétrie point

Étant donné trois points `A, B, C` le point :
`symetrie(droite(B,C),A)` est le transformé du point `A` dans la symétrie par rapport à la droite BC,
On a aussi `sd:=symetrie(droite(B,C))` est la symétrie par rapport à la droite BC,
`symetrie(B,A)` est le transformé du point `A` dans la symétrie de centre `B`.

On a aussi `sp:=symetrie(B)` est la symétrie de centre `B`. On peut aussi comme exercice de programmation définir les fonctions `Symetrie_point` `Symetrie_droite` (`symetrie` est la commande de `Xcas` qui réalise ces 2 fonctions).
On tape (`A` et `M` sont des points et `d` est une droite) :

```
Symetrie_point((A,M)):=A+(A-M)
Symetrie_droite(d,M):= {local N:=inter_unique(Perpendiculaire(M,d),d)
```

1.8.4 La rotation

Étant donné deux points `A,` `B` et un réel u, le point :
`rotation(B,u,A)` est le transformé du point `A` dans la rotation de centre `B` et d'angle de mesure u radians (ou degrés selon la configuration du `CAS` menu `Cfg->Configuration du CAS`).
On a aussi `r:=rotation(B,u)` est la rotation de centre `B` et d'angle de mesure u radians (ou degrés selon la Configuration du `CAS` menu `Cfg`).

1.8.5 La projection

Étant donné trois points `A,` `B,` `C` le point :
`projection(droite(B,C),A)` est la projection orthogonale du point `A` sur la droite BC.
On a aussi `p:=projection(droite(B,C))` est la projection orthogonale sur la droite BC.

1.8.6 La similitude

Étant donné deux points `A,` `B`, un réel k et un réel u, le point :
`similitude(B,k,u,A)` est le transformé du point `A` dans la similitude de centre `B`, de rapport k et d'angle mesure u radians (ou degrés selon la configuration du `CAS` menu `Cfg->Configuration du CAS`).
On a aussi `s:=similitude(B,k,u)` est la similitude de centre `B`, de rapport k et d'angle mesure u radians (ou degrés selon la configuration du `CAS` menu `Cfg->Configuration du CAS`).

1.8.7 L'inversion

Étant donné deux points `A,` `B`, un réel k le point :
`inversion(C,k,A)` est le transformé du point `A` dans l'inversion de centre `C`, de rapport k.
Par exemple si :
`A1:=inversion(C,2,A)`, `A1` est tel que $\overline{\texttt{CA1}} * \overline{\texttt{CA}} = 2$.
`inversion(C,k,M)` est donc le point défini par :
`point(C+k*conj(inv(M-C)))`
On a aussi `inver:=inversion(C,k)` est l'inversion de centre `C`, de rapport k.

1.9 Les lieux géométriques

L'instruction `lieu` permet de tracer le lieu d'un point `M`, lorsque ce point `M` est fonction d'un point `P` pouvant se déplacer sur un objet géométrique `G` et si on a défini `P` par `P:=element(G)`).
Remarque :
Il faut que les paramètres de `lieu` soient des noms de variables.
Donc pour obtenir le lieu d'un point `M`, il faut avoir défini ce point par une affectation à un nom de variable, par exemple `M:=...`
On écrit alors :
`lieu(M,P)`

1.9.1 Des exemples de lieux

Exemple 1 :
Lieu du centre de gravité du triangle ABC quand A se déplace sur une parallèle à BC.
Ici on cherche le lieu du centre de gravité du triangle PBC lorsque P se déplace sur la parallèle à BC passant par A.
On clique pour obtenir trois points `A, B, C` et on tape :

```
D:=parallele(A,droite(B,C));
P:=element(D);
G:=isobarycentre(P,B,C);
lieu(G,P);
```

`lieu(G,P)` trace le lieu du point `G` en fonction de `P`.
Si l'on veut pouvoir animée la figure on tape la liste des instructions qui se trouve dans le fichier `lieu1`, on clique pour obtenir trois points `A, B, C` puis, on fait `Charger session` du menu `Fich` de `Xcas` et on selectionne `lieu1` du répertoire `examples/geo` pour exécuter ce fichier) :
Ou on clique pour créer trois points `A, B, C` et on tape :

```
D:=parallele(A,droite(B,C));
t:=element(-5..5)
P:=element(D,t);
G:=isobarycentre(P,B,C);
lieu(G,P);
```

ainsi lorsqu'on déplace la barre verticale indiquant la valeur de `t`, on voit se déplacer les deux points `P` et `G` simultanément.
Exemples 2 et 3 :
Soient deux points `A` et `B` et `C1` le cercle de diamètre `A B`. Soit `O` le centre de ce cercle et `C` un point variable de ce cercle.
Soit `N` l'intersection de la tangente en `C` à `C1` avec la parallèle à `AC` passant par `O`.
Trouver le lieu de `N` quand `C` varie.

La liste des instructions pour trouver ce lieu se trouve dans le fichier `lieu2`, on clique pour obtenir deux points `A, B` puis, on fait `Charger session` du menu `Fich` de `Xcas` et on selectionne `lieu2` du répertoire `examples/geo`

pour exécuter ce fichier.
Ou on clique pour créer deux points `A, B` et on tape :

```
C1:=cercle(A,B);
O:=milieu(A,B);
C:=element(C1);
T:=tangent(C1,C);
D2:=parallele(O,droite(A,C));
M:=inter(T,D2);
N:=M[0];
lieu(N,C);
```

Attention `M` est une liste ayant un seul élément et on ne peut pas écrire `lieu(M[0],C)` car `M[0]` n'est pas un nom de variable.
On peut aussi écrire des instructions sans utiliser la fonction `tangent` et en rajoutant `t:=element(0..2*pi);` (c'est le fichier `lieu3`) :
On clique pour créer deux points `A, B`.

```
C1:=cercle(A,B);
O:=milieu(A,B);
t:=element(0..2*pi);
C:=element(C1,t);
D1:=perpendiculaire(C,droite(O,C));
D2:=parallele(O,droite(A,C));
M:=inter(D1,D2);
N:=M[0];
lieu(N,C);
```

Les triangles NOC et NOB sont égaux car leurs angles $\widehat{O}$ sont égaux à l'angle $\widehat{CAB}$ et à l'angle $\widehat{OCA}$. Donc NB est perpendiculaire à AB.
La réciproque est évidente car si `N` se trouve sur la perpendiculaire à `AB` en `B`, on méne par `N` l'autre tangente au cercle pour définir le point `C`.
Les triangles rectangles `NOC` et `NOB` sont égaux et donc `ON` est parallèle à `AC`.
Exemple 4 :
Une droite variable `D1` coupe les cotés `AB,AC,BC` du triangle `ABC` en `D,E,M`.
Les cercles circonscrits aux triangles `BDM` et `ECM` se coupent en `P` et `M`. Trouver le lieu de `P` quand `D1` varie.
On peut faire éxécuter le fichier `lieu4` (on clique pour obtenir trois points `A, B, C` puis on fait `Charger session` du menu `Fich` de `Xcas` et on selectionne `lieu4` du répertoire `examples/geo` pour exécuter ce fichier) ou,
on clique pour obtenir trois points `A, B, C` et on tape :

```
Triangle(A,B,C);
u:=element(-4..4);
M:=element(droite(B,C),u);
D:=element(droite(A,B));
D1:=droite(D,M);
E:=(inter(droite(A,C),D1))[0];
C1:=circonscrit(B,D,M);
C2:=circonscrit(E,C,M);
```

```
L:=inter(C1,C2);
if (affixe(L[0])==affixe(M)) {Q:=L[1];} else {Q:=L[0];};
P:=Q;
lieu(P,M);
```

On remarquera que l'on est obligé d'utiliser un point intermédiaire `Q` pour pouvoir définir `P` par une affectation.
Le lieu de P est le cercle circonscrit au triangle ABC car l'angle l'angle $\widehat{P}$ du triangle PBC est égal à l'angle $\widehat{A}$ du triangle ABC en écrivant des égalités d'angles inscrits.
Exemple 5 :
Lieu de l'orthocentre du triangle ABC quand A se déplace sur une parallèle à BC.
Ici on cherche le lieu de l'orthocentre du triangle MBC lorsque M se déplace sur la parallèle à BC passant par A.
On peut faire éxécuter le fichier `lieu5` (on clique pour obtenir trois points `A, B, C` puis, on fait `Charger session` du menu `Fich` de `Xcas` et on selectionner `lieu5` du répértoire `examples/geo` pour exécuter ce fichier) ou,
on clique pour obtenir trois points `A,B,C` et on tape :

```
D:=parallele(A,droite(B,C));
M:=element(D);
H:=(inter(perpendiculaire(M,droite(C,B)),
          perpendiculaire(C,droite(M,B))))[0] ;
lieu(H,M);
```

Le lieu est une parabole passant par B et C et de sommet S orthocentre du triangle isocèle PBC (P sur la médiatrice de BC et sur la droite D). On peut à la main avoir assez facilement l'équation de cette parabole ...mais trouver ce lieu en géométrie pure est plus difficle !

1.9.2 D'autres exemples de lieux

Exemple 1 :
Lieu du centre de gravité du triangle ABC quand A se déplace sur une parallèle à BC.
Ici on cherche le lieu du centre de gravité du triangle PBC lorsque P se déplace sur la parallèle à BC passant par A.
On clique pour obtenir trois points `A, B, C` et on tape :

```
D:=parallele(A,droite(B,C));
P:=element(D);
G:=isobarycentre(P,B,C);
lieu(G,P);
```

`lieu(G,P)` trace le lieu du point `G` en fonction de `P`.
Si l'on veut pouvoir animée la figure on tape la liste des instructions qui se trouve dans le fichier `lieu1`, on clique pour obtenir trois points `A, B, C` puis, on fait `Charger session` du menu `Fich` de `Xcas` et on selectionne `lieu1` du répértoire `examples/geo` pour exécuter ce fichier) :
Ou on clique pour créer trois points `A, B, C` et on tape :

```
A:=point([-3,3/2,'affichage'=0]);
B:=point([-1,-3/2,'affichage'=0]);
C:=point([3,9/4,'affichage'=0]);
D:=parallele(A,droite(B,C));
P:=element(D);
G:=isobarycentre(P,B,C);
lieu(G,P);
```

ainsi lorsqu'on déplace la barre verticale indiquant la valeur de `t`, on voit se déplacer les deux points `P` et `G` simultanément.
`lieu(G,P);` renvoie droite(y=((15*x)/16+17/16)) **Exemples 2 et 3** :
Soient deux points `A` et `B` et `C1` le cercle de diamètre `A B`. Soit `O` le centre de ce cercle et `C` un point variable de ce cercle.
Soit `N` l'intersection de la tangente en `C` à `C1` avec la parallèle à `AC` passant par `O`.
Trouver le lieu de `N` quand `C` varie.

La liste des instructions pour trouver ce lieu se trouve dans le fichier `lieu2`, on clique pour obtenir deux points `A, B` puis, on fait `Charger session` du menu `Fich` de `Xcas` et on selectionne `lieu2` du répértoire `examples/geo` pour exécuter ce fichier.
Ou on clique pour créer deux points `A, B` et on tape :

```
C1:=cercle(A,B);
O:=milieu(A,B);
C:=element(C1);
T:=tangent(C1,C);
D2:=parallele(O,droite(A,C));
M:=inter(T,D2);
N:=M[0];
lieu(N,C);
```

Attention `M` est une liste ayant un seul élément et on ne peut pas écrire `lieu(M[0],C)` car `M[0]` n'est pas un nom de variable.
On peut aussi écrire des instructions sans utiliser la fonction `tangent` et en rajoutant `t:=element(0..2*pi);` (c'est le fichier `lieu3`) :
On clique pour créer deux points `A, B`.

```
C1:=cercle(A,B);
O:=milieu(A,B);
C:=element(C1);
D1:=perpendiculaire(C,droite(O,C));
D2:=parallele(O,droite(A,C));
M:=inter(D1,D2);
N:=M[0];
lieu(N,C);
```

`lieu(N,C)` renvoie `droite(y=((2*x)/3-5/6))`
Les triangles NOC et NOB sont égaux car leurs angles $\widehat{O}$ sont égaux à l'angle $\widehat{CAB}$ et à l'angle $\widehat{OCA}$. Donc NB est perpendiculaire à AB.
La réciproque est évidente car si `N` se trouve sur la perpendiculaire à `AB` en `B`, on

méne par N l'autre tangente au cercle pour définir le point C.
Les triangles rectangles NOC et NOB sont égaux et donc ON est parallèle à AC.
Exemple 4 :
Une droite variable D1 coupe les cotés AB,AC,BC du triangle ABC en D,E,M.
Les cercles circonscrits aux triangles BDM et ECM se coupent en P et M. Trouver le lieu de P quand D1 varie.
On peut faire éxécuter le fichier lieu4 (on clique pour obtenir trois points A, B, C puis on fait Charger session du menu Fich de Xcas et on selectionne lieu4 du répértoire examples/geo pour exécuter ce fichier) ou,
on clique pour obtenir trois points A, B, C et on tape :

```
triangle(A,B,C);
u:=element(-4..4);
M:=element(droite(B,C),u);
D:=element(droite(A,B));
D1:=droite(D,M);
E:=(inter(droite(A,C),D1))[0];
C1:=circonscrit(B,D,M);
C2:=circonscrit(E,C,M);
L:=inter(C1,C2);
if (affixe(L[0])==affixe(M)) {Q:=L[1];} else {Q:=L[0];};
P:=Q;
lieu(P,M);
```

On remarquera que l'on est obligé d'utiliser un point intermédiaire Q pour pouvoir définir P par une affectation.
Le lieu de P est le cercle circonscrit au triangle ABC car l'angle l'angle $\widehat{P}$ du triangle PBC est égal à l'angle $\widehat{A}$ du triangle ABC en écrivant des égalités d'angles inscrits.
Exemple 5 :
Lieu de l'orthocentre du triangle ABC quand A se déplace sur une parallèle à BC.
Ici on cherche le lieu de l'orthocentre du triangle MBC lorsque M se déplace sur la parallèle à BC passant par A.
On peut faire éxécuter le fichier lieu5 (on clique pour obtenir trois points A, B, C puis, on fait Charger session du menu Fich de Xcas et on selectionner lieu5 du répértoire examples/geo pour exécuter ce fichier) ou,
on clique pour obtenir trois points A,B,C et on tape :

```
D:=parallele(A,droite(B,C));
M:=element(D);
H:=(inter(perpendiculaire(M,droite(C,B)),
          perpendiculaire(C,droite(M,B))))[0] ;
lieu(H,M);
```

Le lieu est une parabole passant par B et C et de sommet S orthocentre du triangle isocèle PBC (P sur la médiatrice de BC et sur la droite D). On peut à la main avoir assez facilement l'équation de cette parabole ...mais trouver ce lieu en géométrie pure est plus difficle !

1.9.3 Des exercices de lieux

Exercice1

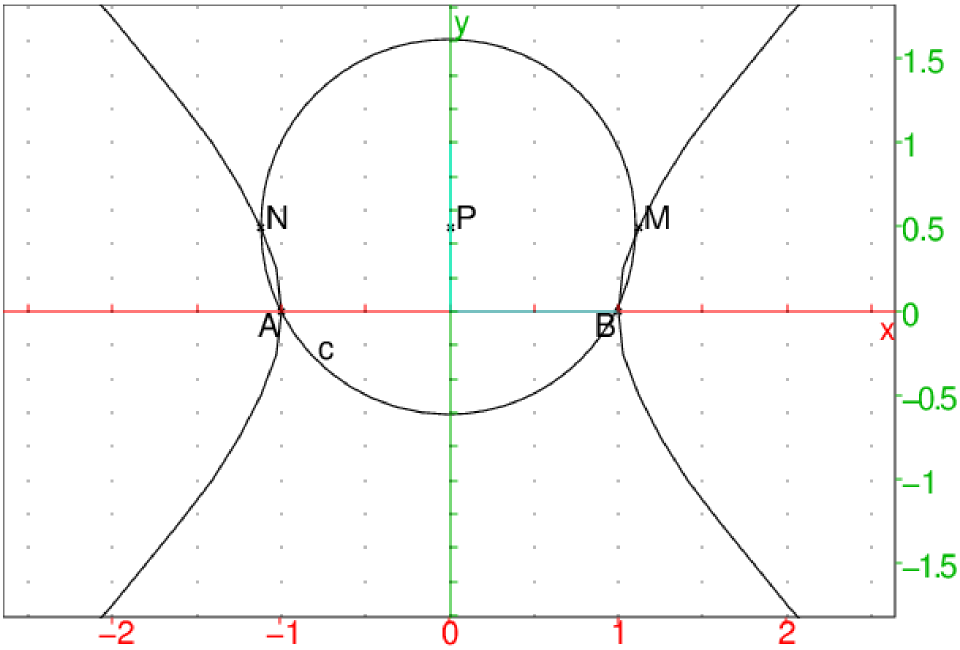
Soient 2 points fixes A et B.
On considère l'ensemble Γ des cercles c passant par A et B.
Pour $c \in \Gamma$, lieu des points M de c qui ont une tangente en M perpendiculaire à AB.
On tape dans un niveau de géométrie 2d (version 1) :

```
A:=point(-1);
B:=point(1);
P:=element(droite(x=0));
c:=cercle(P,A-P);
M:=translation(longueur(A,P),P);
N:=translation(-longueur(A,P),P);
lieu(M,P);
lieu(N,P);
```

ou on tape dans un niveau de géométrie 2d (version 2) :

```
A:=point(-1);
B:=point(1);
assume(u=[0,-5,5,0.1]));
P:=element(droite(x=0),u);
c:=cercle(P,sqrt(u^2+1));
L:=inter(droite(y=u),c);
M:=L[0];
N:=L[1];
lieu(M,u);
lieu(N,u);
```

Pour la version 1 (resp pour la version 2) `lieu(M,P);` (resp `lieu(M,u);`) renvoie :
``plotparam((i)*` t`+sqrt(1+(-` t`)^2),` t`=-10.0..10.0)``
et pour la version 1 (resp pour la version 2) `lieu(N,P);` (resp `lieu(N,u);`) renvoie :
``plotparam((i)*` t`-sqrt(1+(-` t`)^2),` t`=-10.0..10.0)``
et le dessin correspondant :



On peut aussi taper en utilisant directement `plotparam` :

```
supposons(u=[0,-5,5,0.1])
P:=point(i*u)
c:=cercle(P,sqrt(u^2+1));
L:=inter(droite(y=u),c):;
M:=L[0];N:=L[1];
plotparam(affixe(M),u);
plotparam(affixe(N),u);
```

On obtient le dessin ci-dessus.
On tape :
`affixe(M)`
On obtient :
`(i)*u+sqrt(u^2+1)`
On tape :
`affixe(N)`
On obtient :
`(i)*u-sqrt(u^2+1)`

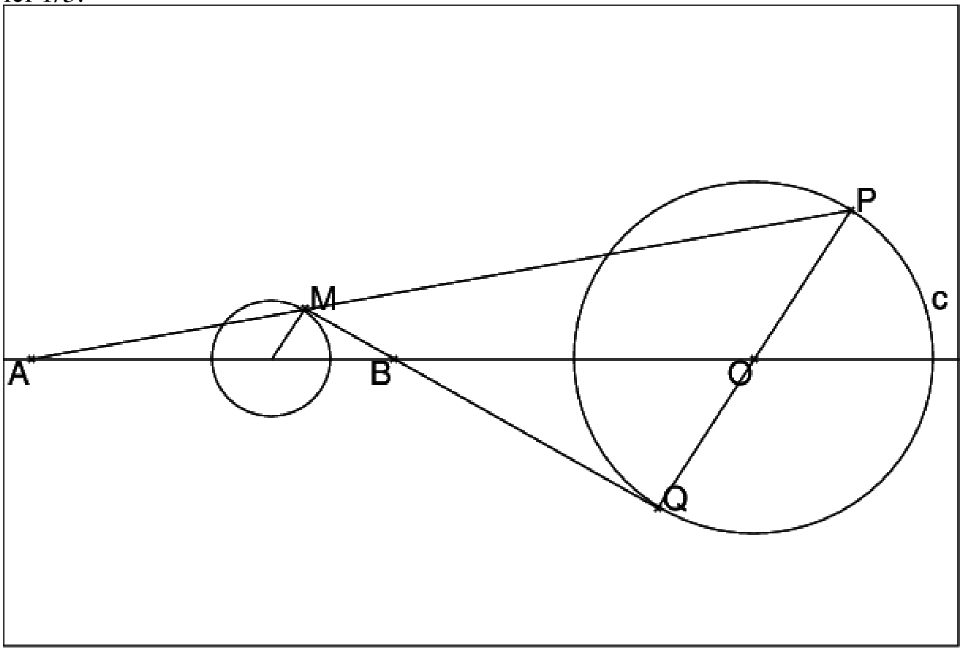
Exercice2

Soient un cercle c de centre O et 2 points fixes A et B situés sur un diamètre de c.
Soit un diamètre variable PQ de c.
Lieu de M intersection de AP et BQ.
On tape :

```
c:=cercle(0,1)
A:=point(-4);
B:=point(-2);
u:=element(-pi..pi);
P:=element(c,u);
```

```
Q:=-P;
M:=inter_unique(droite(A,P),droite(B,Q));
lieu(M,P)
```

On obtient :
`cercle(point(-8/3,0),1/3)`
On tape :
```
droite(A,0);segment(P,A);segment(M,Q)
segment(0,P);segment(M,point(-8/3))
```
On obtient le cercle homothétique de c dans l'homothétie de centre A et de rapport ici 1/3.



On a en effet en appliquant le théoréme de Ménélaüs au triangle OAP on a :
M, B, Q alignés est équivalent à : $\frac{\overline{MA}}{\overline{MP}} \cdot \frac{\overline{QP}}{\overline{QO}} \cdot \frac{\overline{BO}}{\overline{BA}} = 1$
On a $\frac{\overline{QP}}{\overline{QO}} = 2$. Si $\frac{\overline{BA}}{\overline{BO}} = k$ alors $\frac{\overline{MA}}{\overline{MP}} = \frac{k}{2}$ donc
$\frac{\overline{AP}}{\overline{AM}} = 1 + \frac{\overline{MP}}{\overline{AM}} = 1 - \frac{2}{k} = \frac{k-2}{k}$
Ce qui prouve que M se déduit de P dans l'homothétie de centre A et de rapport :
$\frac{k}{k-2}$
Dans l'exemple on a $k = -1$ donc le rapport de l'homothétie est bien : $\frac{1}{3}$

1.10 Figures polygonales

1.10.1 Le triangle

Étant donné trois points `A`, `B` et `C` les commandes suivantes permettent de tracer des triangles :
— Le triangle quelconque
  `triangle(A,B,C)` trace le triangle ABC.

— Le triangle équilatéral
`triangle_equilateral(A,B)` trace le triangle équilatéral direct ABC mais sans définir le point C. Pour définir C, il faut donner un nom au triangle (par exemple T) et utiliser la commande `sommets(T)` qui renvoie la liste des sommets de T.
On écrira alors :
`T:=triangle_equilateral(A,B);C:= sommets(T)[2].`
Mais, pour définir C, il est plus facile d'utiliser :
`triangle_equilateral(A,B,C)` qui trace le triangle équilatéral direct ABC en mettant dans la variable `C` le point C.

— Le triangle isocèle
On suppose que c désigne la mesure en radians (ou en degrés) de l'angle $(\overrightarrow{AB}, \overrightarrow{AC})$.
`triangle_isocele(A,B,c)` trace (sans définir le point C) le triangle isocèle ABC de sommet A, tel que l'angle $(\overrightarrow{AB}, \overrightarrow{AC}) = c$ radians (ou degrés).
`triangle_isocele(A,B,c,C)` trace le triangle isocèle ABC de sommet A, tel que l'angle $(\overrightarrow{AB}, \overrightarrow{AC}) = c$ radians (ou degrés), en mettant dans la variable `C` le point C.
On définit deux points A et B et on tape, par exemple, pour avoir un triangle rectangle isocéle ABC :
`triangle_isocele(A,B,pi/2,C)`
Activité
Créer un segment `AB`.
Construire un triangle isocèle `ABC` de sommet `A` ($AB = AC$) tel que l'angle A mesure 30°=$\frac{\pi}{6}$ radians.
Construire un triangle isocèle `ABD` de sommet `A` ($AB = AD$) tel que l'angle B mesure 30°=$\frac{\pi}{6}$ radians.
Réponse
Supposons que l'on a choisi `radian` dans la fenêtre de configuration du `CAS`.
On clique avec la souris pour avoir le segment `AB` puis, on exécute la liste des instructions :
`C:=rotation(A,pi/6,B)` C est le transformé de B dans la rotation de centre A et d'angle $\frac{\pi}{6}$ radians,
`D:=rotation(A,2*pi/3,B)` D est le transformé de B dans la rotation de centre A et d'angle $\frac{2*\pi}{3}$ radians (car $\pi - 2 * \pi/6 = 2 * \pi/3$).

— Le triangle rectangle
`triangle_rectangle(A,B,k)` trace (sans définir le point C) le triangle ABC rectangle en A tel que $longueur(A, C) = |k| {*} longueur(A, B)$: ce triangle est direct si $k > 0$ et indirect sinon.
`triangle_rectangle(A,B,k,C)` trace le triangle ABC rectangle en A tel que $longueur(A, C) = |k| * longueur(A, B)$, ce triangle est direct si $k > 0$ et indirect sinon et met dans la variable `C` le point C.
Si on suppose que c contient la mesure en radians (ou en degrés) de l'angle $(\overrightarrow{BA}, \overrightarrow{BC})$, on a :
`triangle_rectangle(A,B,tan(c))` ou

`triangle_rectangle(A,B,tan(c),C)` trace le triangle ABC rectangle en A tel que l'angle $(\overrightarrow{BC}, \overrightarrow{BA}) = c$ radians (ou degrés).
On remarquera que le troisième paramètre a pour valeur absolue $\frac{AC}{AB}$ et que le triangle est direct si il est positif et indirect si il est négatif.
Activité
Créer un segment `AB` et un segment `CD` tels que les deux points `A` et `B` soient en dehors de la droite CD.
Construire un point `E` sur la droite CD pour que le triangle `ABE` soit rectangle en `E` (Ê=90°=$\frac{\pi}{2}$ radians).
Réponse
E se trouve sur le cercle de diamètre AB et sur la droite CD.
On clique avec la souris pour avoir les segments `AB` et `CD` puis, on exécute la liste des instructions qui se trouve dans `geo10` (faire `Charger session` du menu `Fich` de `Xcas` et selectionner `geo10` du répértoire `examples/geo` pour exécuter ce fichier).
Voici le détail de `geo10` :
`E:=inter(droite(C,D),cercle(A,B))` est l'intersection de la droite CD avec le cercle de diamètre AB,
`E1:=E[0]` est le premier point de cette intersection,
`E2:=E[1]` est le deuxième point de cette intersection,
`triangle(E1,A,B)` trace le triangle $ABE1$
`triangle(E2,A,B)` trace le triangle $ABE2$
Ces deux triangles répondent à la question.

1.10.2 Les quadrilatères

Les commandes suivantes permettent de tracer des quadrilatères :

- Le quadrilatère quelconque :
  On donne 4 points A, B, C, D et on construit le quadrilatère $ABCD$.
  On a :
  `quadrilatere(A,B,C,D)` trace le quadrilatère $ABCD$.
- Le carré :
  On donne 2 points A, B et on construit le carrè $ABCD$ de sens direct. On a :
  `carré(A,B)` trace le carrè $ABCD$ de sens direct, mais sans définir les points C et D.
  `carré(A,B,C,D)` trace le carrè $ABCD$ de sens direct, et met dans `C` et `D` les points C et D.
- Le losange :
  On donne 2 points A, B et une mesure d'angle t et on construit le losange $ABCD$ tel que $(\overrightarrow{AB},\overrightarrow{AD}) = t$. On a :
  `losange(A,B,t)` trace le losange $ABCD$ tel que :
  $(\overrightarrow{AB},\overrightarrow{AD})$=t, mais sans définir les points C et D.
  `losange(A,B,t,C,D)` trace le losange $ABCD$ tel que :
  $(\overrightarrow{AB},\overrightarrow{AD}) = t$, et met dans `C` et `D` les points C et D.
- Le rectangle :
  On donne 2 points A, B et un réel k et on construit le rectangle $ABCD$ (direct si $k > 0$ et indirect sinon) tel que $AD = |k| * AB$.

On a :
`rectangle(A,B,k)` trace (sans définir les points C et D) le rectangle $ABCD$ tel que $AD = |k| * AB$. Ce rectangle est direct si $k > 0$ et indirect sinon.
`rectangle(A,B,k,C,D)` trace le rectangle $ABCD$ tel que : $AD = |k| * AB$, et met dans `C` et `D` les points C et D. Ce rectangle est direct si $k > 0$ et indirect sinon.

— Le parallélogramme :
On donne 3 points A, B, C et on construit le e parallélogramme $ABCD$ c'est à dire le point D tel que $\overrightarrow{AD} = \overrightarrow{BC}$.
On a :
`parallelogramme(A,B,C)` trace le parallélogramme $ABCD$.
`parallelogramme(A,B,C,D)` trace le parallélogramme $ABCD$ et met dans `D` le point D.

1.10.3 Les polygones

Soient n points (ou n nombres complexes représentant l'affixe de ces points).
`polygone` avec comme argument la liste ou la séquence de ces n points.
`polygone` renvoie et trace le polygone ayant pour sommets ces n points.
`polygone(0,1,1+i,i)` trace le carré de sommets 0,1,1+i,i.
`polygone(makelist(x->exp(i*pi*x/3),0,5,1))` trace l'hexagone de centre 0 et de sommets 1,$e^{\frac{i*pi}{3}}$...$e^{\frac{5*i*pi}{3}}$

1.10.4 Les sommets d'un polygone

Les commandes précédentes peuvent tracer des triangles ou des quadrilatères sans avoir tous les sommets, mais on peut avoir besoin des sommets de ces triangles ou de ces quadrilatères.
Si `T` est un triangle (ou un polygone) `sommets(T)` renvoie la liste circulaire des sommets de `T`, ainsi `sommets(T)[0]` et `sommets(T)[3]` désigne le premier sommet de `T` (attention aux indices ! ! !).
Exemple :
On clique pour avoir deux points `A` et `B`.
On tape :

```
T:=triangle_equilateral(A,B)
C:=sommets(T)[2]
```

Ainsi le triangle ABC est équilatéral.
On tape :

```
Q:=carre(A,B)
E:=sommets(T)[2]
F:=sommets(T)[3]
```

Ainsi, $ABEF$ est un carré.

1.10.5 Activités : constructions de quadrilatères

— Le parallélogramme
Activité

Créer trois points `A, B, C`.
Construire le quatrième sommet `D` du parallélogramme `ABCD` : on décrira trois méthodes de construction du point `D`.
Réponse
1/ D est tel que $\overrightarrow{AD} = \overrightarrow{BC}$.
On tape :
`D:=A+C-B` (ou encore `D:=translation(C-B,A)`)
2/ AC et BD ont même milieu.
On tape :
```
M:=milieu(A,C)
D:=symetrie(M,B)
```
3/ AD est paralléle à BC et AB est paralléle à DC.
On tape :
```
D1:=parallele(A,droite(B,C))
D2:=parallele(C,droite(B,A))
D:=inter(D1,D2)[0]
```
On peut ensuite vérifier avec la commande `parallelogramme`.

— Le carré
Activité
Créer un segment `AB`.
Construire un carré de côté `AB` : on décrira deux méthodes de construction du carré et une méthode qui utilise les nombres complexes et un petit programme.
Réponse
1/ On obtient C et D par rotation.
On tape :
$\mathtt{C} := \mathtt{rotation}(\mathtt{B}, -\pi/2, \mathtt{A})$
$\mathtt{D} := \mathtt{rotation}(\mathtt{A}, \pi/2, \mathtt{B})$
(ou encore `C:=B-i*(A-B) D:=A+i*(B-A)`)
Puis on trace le carré :
```
segment(A,B);segment(B,C);segment(C,D);segment(D,A);
```
2/ On obtient C par rotation et D par symétrie .
On tape :
$\mathtt{C} := \mathtt{rotation}(\mathtt{B}, -\pi/2, \mathtt{A})$
```
M:=milieu(A,C)
D:=symetrie(M,B)
```
Puis on trace le carré :
```
segment(A,B);segment(B,C);segment(C,D);segment(D,A);
```
3/ On utilise les nombres complexes et on écrit le programme :
```
carred(A,B):={
local C,D;
C:=B+i*(B-A);
D:=A+i*(B-A);
return ([segment(A,B),segment(B,C),
          segment(C,D),segment(D,A)]);
};
```
Attention
Lorsque il y a des commandes graphiques intermédiaires, elles sont tra-

cées dans l'écran `DispG`, mais seule la valeur de la fonction (ce qui suit `return`) est tracée.

On peut ensuite vérifier avec la commande `carre`.

1.10.6 Un exercice sur la droite des milieux

L'exercice

Soit un quadrilatère quelconque $ABCD$ et M, N, P, Q les milieux respectifs de AB, BC, CD, DA.

- Quelle est la nature du quadrilatère $MNPQ$?
- Trouver une condition nécessaire et suffisante pour que le quadrilatère $MNPQ$ soit un losange.
- Trouver une condition nécessaire et suffisante pour que le quadrilatère $MNPQ$ soit un rectangle.
- Trouver une condition nécessaire et suffisante pour que le quadrilatère $MNPQ$ soit un carré.

La solution

Il est conseillé de faire une figure avec `Xcas` : on ouvre l'écran de géométrie et on clique sur 4 points `A, B, C, E` on modifie la ligne `E:=...` en `D:=...` puis on tape :

```
M:=milieu(A,B);
N:=milieu(C,B);
P:=milieu(C,D);
Q:=milieu(A,D);
quadrilatere(A,B,C,D,couleur=rouge);
segment(A,C,couleur=vert);
segment(B,D,couleur=vert);
quadrilatere(M,N,P,Q);
```

Puis on se met en mode `Pointer` et on déplace le point `D` avec la souris et on observe......

Cette session se trouve parmi les exemples sous le nom `milieu.xws`.

La démonstration :

- MN joint les milieux des côtés AB et BC du triangle ABC donc MN est parallèle à AC et $MN = AC/2$,
  de même PQ joint les milieux des côtés AD et DC du triangle ADC donc PQ est parallèle à AC et $PQ = AC/2$. Le quadrilatère $MNPQ$ est donc un prallélogramme puisque ses côtés MN et PQ sont égaux et parallèles.
  Remarque
  On a aussi :
  MQ joint les milieux des côtés AB et AD du triangle DAB donc MQ est parallèle à BD et $MQ = BD/2$ et
  NP joint les milieux des côtés AB et AD du triangle DCB donc NP est parallèle à BD et $NP = BD/2$.

— Une condition nécessaire et suffisante pour que le prallélogramme $MNPQ$ soit un losange est que $MN = NP$. D'aprés la remarque précédente l'égalité $MN = NP$ est équivalente à $BD = AC$ c'est à dire à l'égalité des diagonales du quadrilatère $ABCD$.
Donc une condition nécessaire et suffisante pour que le quadrilatère $MNPQ$ soit un losange est que les diagonales du quadrilatère $ABCD$ soient égales.
— Une condition nécessaire et suffisante pour que le prallélogramme $MNPQ$ soit un rectangle est que $MN \perp NP$. D'aprés la remarque précédente, MN est parallèle à AC et NP est parallèle à BD, donc $MN \perp= NP$ est équivalente à $BD \perp AC$.
Donc une condition nécessaire et suffisante pour que le quadrilatère $MNPQ$ soit un losange est que les diagonales du quadrilatère $ABCD$ soient perpendiculaires.
— Une condition nécessaire et suffisante pour que le prallélogramme $MNPQ$ soit un carré est que $MN = NP$ et $MN \perp= NP$. Donc une condition nécessaire et suffisante pour que le quadrilatère $MNPQ$ soit un losange est que les diagonales du quadrilatère $ABCD$ soient égales et perpendiculaires.

Vrai/Faux

— Si $ABCD$ est un rectangle alors $MNPQ$ est un losange
— Si $MNPQ$ est un losange alors $ABCD$ est un rectangle
— Si $ABCD$ est un losange alors $MNPQ$ est un rectangle
— Si $MNPQ$ est un rectangle alors $ABCD$ est un losange
— Si $ABCD$ est un carré alors $MNPQ$ est un carré
— Si $MNPQ$ est un carré alors $ABCD$ est un carré
— Pour que $MNPQ$ soit un carré, il suffit que $ABCD$ soit un carré
— Pour que $MNPQ$ soit un carré, il faut que $ABCD$ soit un carré
— Pour que $MNPQ$ soit un losange, il suffit que $ABCD$ soit un losange
— Pour que $MNPQ$ soit un losange, il faut que $ABCD$ soit un losange
— Pour que $MNPQ$ soit un losange, il suffit que $ABCD$ soit un rectangle
— Pour que $MNPQ$ soit un losange, il faut que $ABCD$ soit un rectangle
— Pour que $MNPQ$ soit un rectangle, il suffit que $ABCD$ soit un losange
— Pour que $MNPQ$ soit un rectangle, il faut que $ABCD$ soit un losange
— Pour que $MNPQ$ soit un rectangle, il suffit que $ABCD$ soit un rectangle
— Pour que $MNPQ$ soit un rectangle, il faut que $ABCD$ soit un rectangle

Répondre par vrai ou faux en donnant des exemples lorsque la proposition est fausse et une démonstration se référant à l'exercice lorsque la proposition est vraie.

1.10.7 Les polygones réguliers : `isopolygone`

`isopolygone` permet de tracer des polygones réguliers ayant k cotés.
Les arguments sont :
- soit deux sommets (deux points ou deux complexes) et le nombre de cotés (un entier positif k),
- soit le centre du polygone et un sommet (deux points ou deux complexes) et un entier négatif $-k$ (si k le nombre de cotés),

Sans se servir de la commande `isopolygone` tracer :

— L'hexagone

Activité

1/ Créer un segment `AB`. Construire un hexagone régulier `ABCDEF`.
2/ Créer un segment `AB`. Construire un hexagone régulier de sorte que `AB` soit un segment qui joint les milieux de deux cotés opposés de l'hexagone.
3/ Créer un segment `AB`. Construire un hexagone régulier de sorte que `AB` soit un segment qui joint les milieux de deux cotés adjacents de l'hexagone.

Réponse

1/ On utilise les nombres complexes et le fait que les angles extérieurs d'un hexagone direct sont de $4\pi/3$ radians.
On peut donc écrire le programme qui trace l'hexagone régulier direct $ABCDEF$:

```
hexagone(A,B):={
local l,C;
l:=[];
for (j:=1;j<=6;j++){
l:=append(l,segment(A,B));
C:=B+(1+i*sqrt(3))*(B-A)/2;
//ou C:=rotation(B,4*pi/3,A)
A:=B;
B:=C;
}
return(l);
};
```

Ainsi `hexagone(A,B)` et `hexagone(B,A)` répondent à la question.
2/ On construit deux sommets consécutifs C et D symétriques par rapport à B et on utilise la fonction `hexagone` précédente.
On tape :

```
C:=similitude(B,1/2/sqrt(3),pi/2,A)
D:=similitude(B,1/2/sqrt(3),-pi/2,A)
hexagone(C,D)
```

3/ C se déduit de A par similiude de centre B et D est le symëtrique de C par rapport à B puis on utilise la fonction `hexagone` précédente.
On tape :

```
C:=similitude(B,1/sqrt(3),pi/6,A)
D:=symetrie(B,C)
hexagone(C,D)
```

— Le N-gone

On peut écrire le programme qui trace le N-gone direct $AB...$:

```
Ngone(A,B,n):={
local l,C;
l:=[];
for (j:=1;j<=n;j++){
l:=append(l,segment(A,B));
C:=rotation(B,(n+2)*pi/n,A)
A:=B;
```

```
B:=C;
}
return(l);
};
```

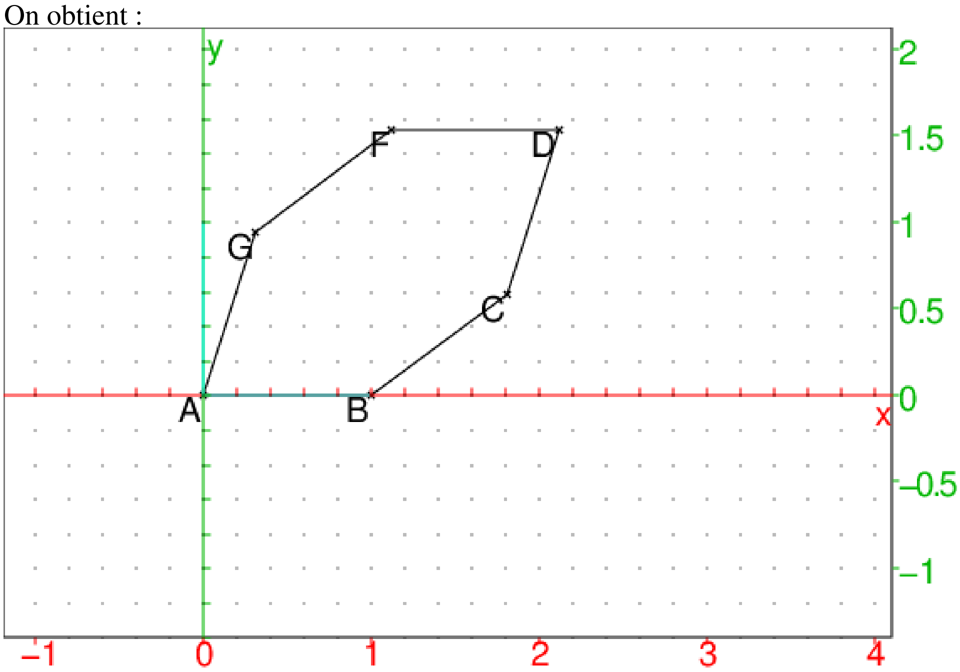
1.11 Un pavage avec un hexagone non régulier

1.11.1 Le pavé

Le pavé est un hexagone $ABCDFG$ ayant ses 6 côtés égaux et ayant comme angles A (resp B, C, D, F, G) $2\pi/5$ (resp $4\pi/5$, $4\pi/5$, $2\pi/5$, $4\pi/5$, $4\pi/5$).
On tape dans un niveau de géométrie 2d :

```
A:=point(0);
B:=point(1);
C:=point(1+exp(i*pi/5));
D:=point(1+exp(i*pi/5)+exp(2*i*pi/5));
G:=point(exp(2*i*pi/5));
F:=point(exp(2*i*pi/5)+exp(i*pi/5));
polygone(A,B,C,D,F,G);
```

On obtient :



1.11.2 Le programme de ce pavage

On note `a,b,c,d,f,g` les affixes des points `A,B,C,D,F,G` et `t` l'angle que fait `AB` avec l'axe des x.

```
hexa(a,t):={
  local b,c,d,f,g;
```

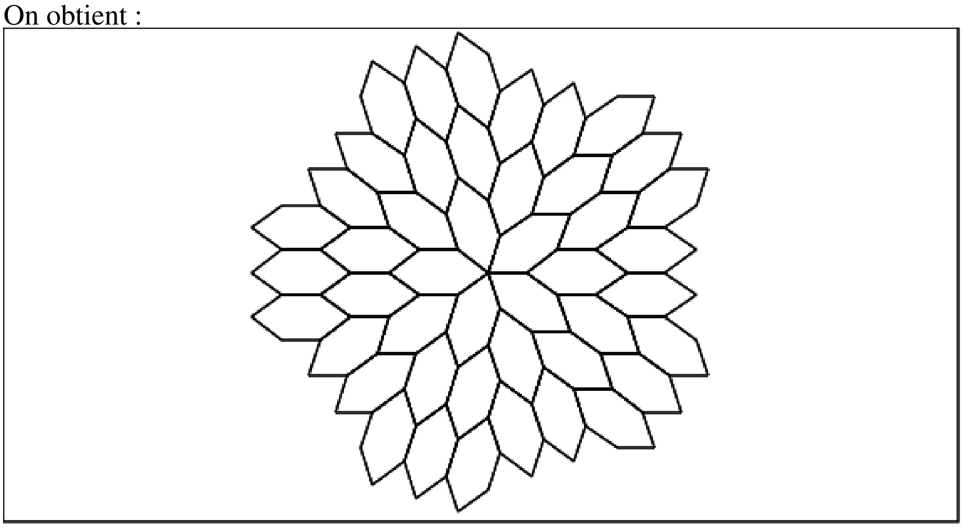
```
  b:=a+exp(i*t);
  c:=b+exp(i*(pi/5+t));
  d:=c+exp(i*(2*pi/5+t));
  g:=a+exp(i*(2*pi/5+t));
  f:=g+exp(i*(pi/5+t));
  return polygone(a,b,c,d,f,g);
  }:;
hexap():={
  local L,t,a,j;
   L:=NULL;
   t:=0;a:=0;
   pour j de 1 jusque 5 faire
   L:=L,hexa(0,t);
   t:=t+2*pi/5;
   fpour;
   a:=1;t:=-pi/5;
   pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
   t:=t+2*pi/5;
   a:=a*exp(2*i*pi/5);
   fpour;
 a:=1+exp(i*pi/5);t:=0;
   pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
   t:=t+2*pi/5;
   a:=a*exp(2*i*pi/5);
   fpour;
a:=exp(i*pi/5)+exp(i*(2*pi/5))
  pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
   t:=t+2*pi/5;
   a:=a*(exp(2*i*pi/5));
   fpour;
 a:=(2+exp(-i*pi/5));t:=-pi/5;
   pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
   t:=t+2*pi/5;
   a:=a*exp(2*i*pi/5);
   fpour;
a:=2+exp(i*pi/5);t:=-pi/5;
  pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
   t:=t+2*pi/5;
   a:=a*(exp(2*i*pi/5));
   fpour;
 a:=2+2*exp(-i*pi/5);t:=-2*pi/5;
  pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
```

```
   t:=t+2*pi/5;
   a:=a*(exp(2*i*pi/5));
   fpour;
   a:=1+2*exp(-i*pi/5)+exp(-2*i*pi/5);t:=-2*pi/5;
  pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
   t:=t+2*pi/5;
   a:=a*(exp(2*i*pi/5));
   fpour;
 a:=2*exp(-i*pi/5)+2*exp(-2*i*pi/5);t:=-2*pi/5;
  pour j de 1 jusque 5 faire
   L:=L,hexa(a,t);
   t:=t+2*pi/5;
   a:=a*(exp(2*i*pi/5));
   fpour;
return L;
   }:;
```

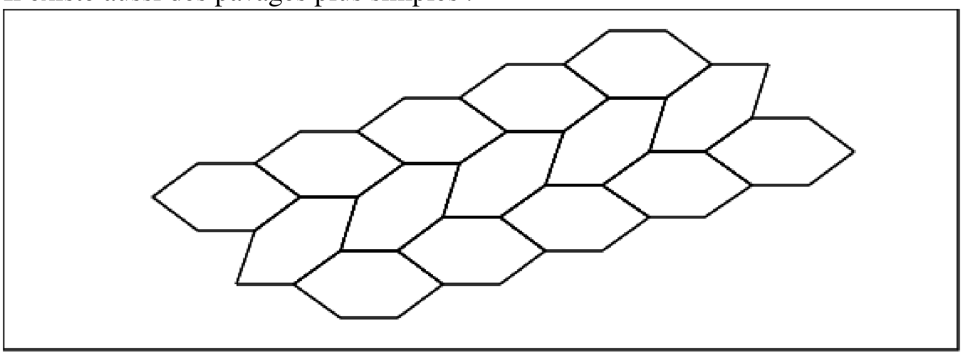
On tape :

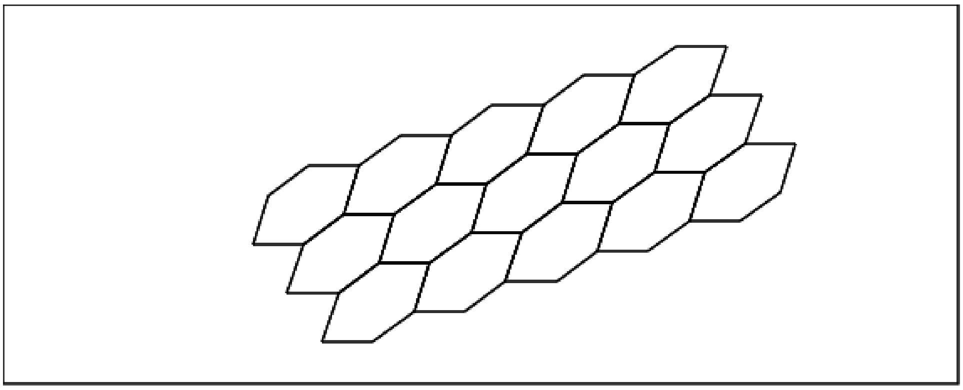
```
hexap()
```

On obtient :



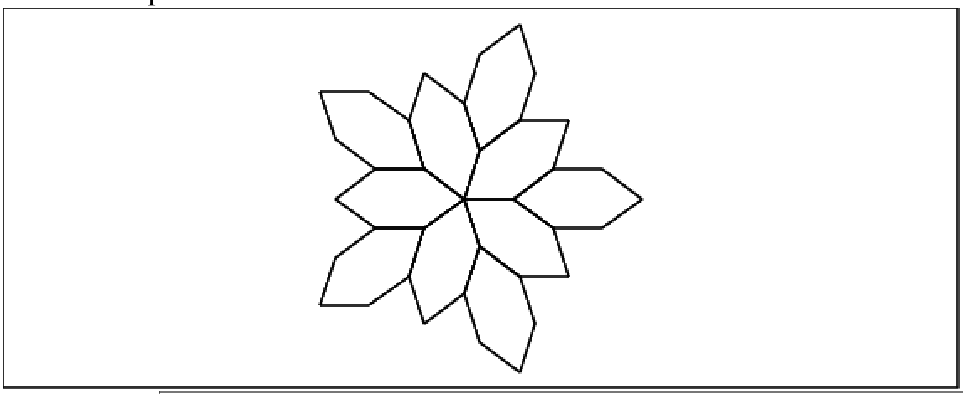
Il existe aussi des pavages plus simples !

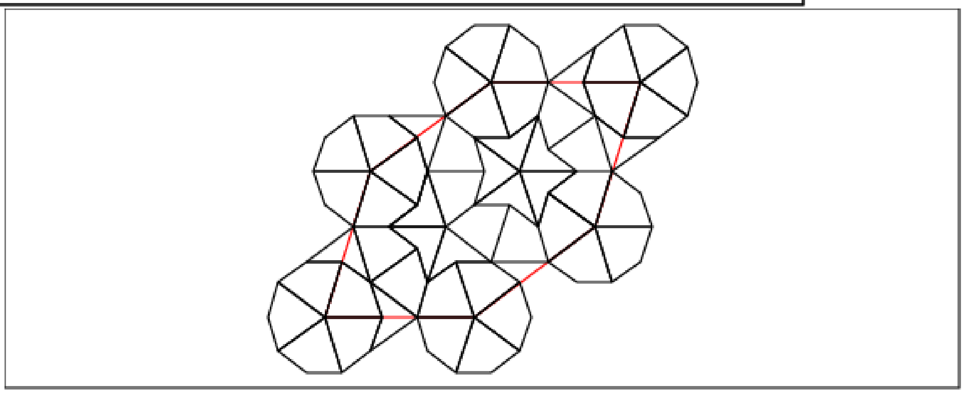




1.12 Un pavage avec 2 pavés

On veut pavé cette étoile :





Avec ce pavé
On tape :

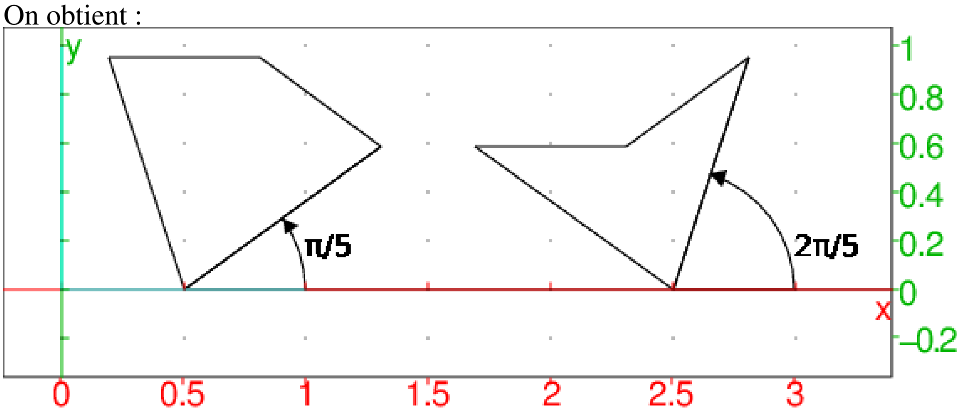
```
hexal(a,t,l):={
  local b,c,d,f,g;
  b:=a+l*exp(i*t);
  c:=b+l*exp(i*(pi/5+t));
  d:=c+l*exp(i*(2*pi/5+t));
  g:=a+l*exp(i*(2*pi/5+t));
  f:=g+l*exp(i*(pi/5+t));
  return polygone(a,b,c,d,f,g);
  }:;
pavea(a,t):={
```

```
  polygone(a,a+exp(i*t),a+exp(i*(t+pi/5)),a+exp(i*(t+2*pi/5)));
      }:;
paveb(b,t):={
  local r;
  r:=2*sin(pi/10);
  polygone(b,b+exp(i*t),b+r*exp(i*(t+pi/5)),b+exp(i*(t+2*pi/5)));
      }:;
```

On tape :

```
pavea(1/2,pi/5),angle(1/2,3,1/2+exp(i*pi/5),"pi/5"),
paveb(5/2,2*pi/5),angle(5/2,5,5/2+exp(2*i*pi/5),"2pi/5")
```

On obtient :

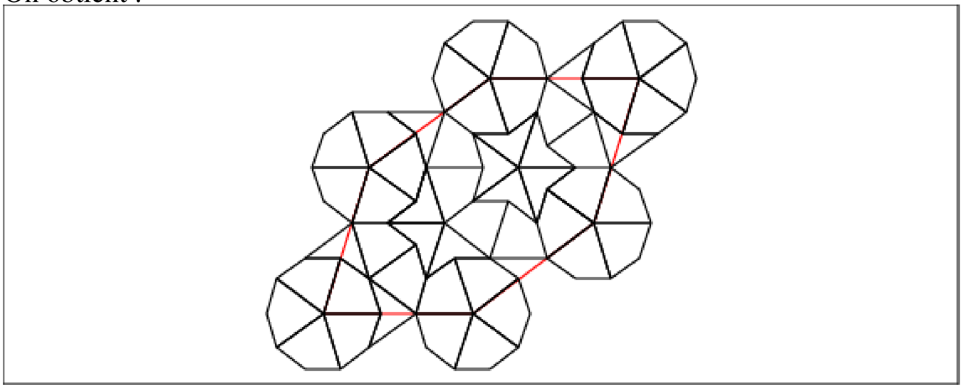


On tape pour définir un pavé hexagonal `pavehat` :

```
pavehat(a,t):={
  local L,l,r;
  r:=2*sin(pi/10);l:=2+r;
  L:=NULL;
L:=L,affichage(hexal(a,t,2+2*sin(pi/10)),1);
L:=L,pavea(a,2*k*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(2*i*pi/5+i*t),(2*k+1)*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(i*t),(2*k+1)*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(2*i*pi/5+i*t)+l*exp(i*pi/5+i*t),
           (2*k)*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(i*t)+l*exp(i*pi/5+i*t),(2*k)*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(i*t)+l*exp(i*pi/5+i*t)+l*exp(2*i*pi/5+i*t),
           (2*k+1)*pi/5+t)$(k=0..4);
L:=L,paveb(a+(3+2*r)*exp(i*pi/5+i*t),(2*k)*pi/5+t)$(k=0..4);
L:=L,paveb(a+l*exp(i*pi/5+i*t),(2*k+1)*pi/5+t)$(k=1..3);
L:=L,paveb(a+(1+r)*exp(2*i*k*pi/5+i*t),4*pi/5+2*k*pi/5+t)$(k=0..2);
L:=L,paveb(a+(2+r)*exp(2*i*pi/5+i*t)+(1+r)*exp((2*k+1)*i*pi/5+i*t),
           pi+2*k*pi/5+t)$(k=0..2);
L:=L,paveb(a+(3+2*r)*exp(i*pi/5+i*t)+(1+r)*exp((-2*k+2)*i*pi/5+i*t),
           -pi/5+2*k*pi/5+t)$(k=0..1);
L:=L,pavea(a+exp(i*pi/5+i*t),(2*k-1)*pi/5+t)$(k=0..1);
L:=L,segment(a+(3+2*r)*exp(i*pi/5+i*t)+r*exp(i*pi+2*i*k*pi/5+i*t),
            a+(3+2*r)*exp(i*pi/5+i*t)+(1+r)*exp(i*pi+2*i*k*pi/5+i*t));
L:=L,segment(a+(2+r)*exp(i*pi/5+i*t),a+(2+2*r)*exp(i*pi/5+i*t));
```

```
L:=L,segment(a+(2+r)*exp(i*t)+exp(2*i*pi/5+i*t),
                a+(3+r)*exp(i*t)+exp(2*i*pi/5+i*t));
return L;
}:;
```

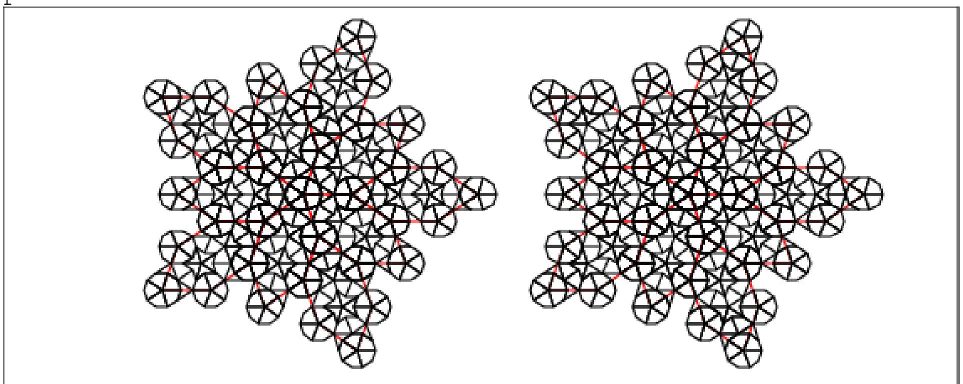
On tape :
`pavehat(0,0)`
On obtient :



On remarquera que les centres des décagone se situent aux sommets de l'hexagone non régulier précédent (ici en rouge).

1.12.1 Deux pavages de l'étoile avec ces 2 pavés

Voici, avec ces 2 pavés, 2 pavages d'une étoile faite avec l'appel de 10 fois `pavehat`.



Pour cela, on tape :

```
pavehat(a,t):={
  local L,l,r;
  r:=2*sin(pi/10);l:=2+r;
  L:=NULL;
L:=L,affichage(hexal(a,t,2+2*sin(pi/10)),1);
L:=L,pavea(a,2*k*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(2*i*pi/5+i*t),(2*k+1)*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(i*t),(2*k+1)*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(2*i*pi/5+i*t)+l*exp(i*pi/5+i*t),
             (2*k)*pi/5+t)$(k=0..4);
L:=L,pavea(a+l*exp(i*t)+l*exp(i*pi/5+i*t),(2*k)*pi/5+t)$(k=0..4);
```

```
L:=L,pavea(a+l*exp(i*t)+l*exp(i*pi/5+i*t)+l*exp(2*i*pi/5+i*t),
           (2*k+1)*pi/5+t)$(k=0..4);
L:=L,paveb(a+(3+2*r)*exp(i*pi/5+i*t),(2*k)*pi/5+t)$(k=0..4);
L:=L,paveb(a+l*exp(i*pi/5+i*t),(2*k+1)*pi/5+t)$(k=1..1);
L:=L,paveb(a+(2+r)*exp(2*i*pi/5+i*t)+(1+r)*exp((2*k+1)*i*pi/5+i*t),
           pi+2*k*pi/5+t)$(k=0..0);
L:=L,paveb(a+(2+r)*exp(i*pi/5+i*t),3*pi/5+2*k*pi/5+t)$(k=0..2);
L:=L,paveb(a+(3+2*r)*exp(i*pi/5+i*t)+(1+r)*exp((-2*k+2)*i*pi/5+i*t),
           -pi/5+2*k*pi/5+t)$(k=0..1);
L:=L,pavea(a+exp(i*pi/5+i*t),(2*k-1)*pi/5+t)$(k=0..1);
L:=L,segment(a+(3+2*r)*exp(i*pi/5+i*t)+r*exp(i*pi+2*i*k*pi/5+i*t),a+
      (3+2*r)*exp(i*pi/5+i*t)+(1+r)*exp(i*pi+2*i*k*pi/5+i*t))$(k=0..3);
L:=L,segment(a+(2+r)*exp(i*pi/5+i*t),a+(2+2*r)*exp(i*pi/5+i*t));
L:=L,segment(a+(2+r)*exp(i*t)+exp(2*i*pi/5+i*t),
             a+(3+r)*exp(i*t)+exp(2*i*pi/5+i*t));
return L;
}:;
```

Pour le premier dessin on tape :

```
pavehat(0,2*k*pi/5)$(k=0..4),
pavehat((2+r)*exp(2*i*(k+1)*pi/5),(2*k+1)*pi/5)$(k=0..4)
```

Pour le deuxième dessin on tape :

```
pavehat(21,2*k*pi/5)$(k=0..4),
pavehat(21+(7+4r)*exp(2*i*(k+1)*pi/5),-4*(2*k+1)*pi/5)$(k=0..4)
```

1.12.2 Un autre pavé

Modifions le pavé `pavehat` pour que ses côtés soient obtenus par des tanslations.

```
pavea2(a,t):={
  polygone(a,a+exp(i*t),a+exp(i*(t+pi/5)),a+exp(i*(t+2*pi/5))),
           segment(a,a+exp(i*(t+pi/5)));
      }:;
paveb(b,t):={
  local r;
  r:=2*sin(pi/10);
  polygone(b,b+exp(i*t),b+r*exp(i*(t+pi/5)),b+exp(i*(t+2*pi/5)));
      }:;
pavehat2(a,t):={
  local L,l,r,B,E;
  r:=2*sin(pi/10);l:=2+r;
  L:=NULL;
L:=L,affichage(hexal(a,t,2+2*sin(pi/10)),1);
L:=L,pavea2(a,2*k*pi/5+t)$(k=0..4);
L:=L,pavea2(a+l*exp(2*i*pi/5+i*t),(2*k+1)*pi/5+t)$(k=0..4);
L:=L,pavea2(a+l*exp(i*t),(2*k+1)*pi/5+t)$(k=0..4);
L:=L,pavea2(a+l*exp(2*i*pi/5+i*t)+l*exp(i*pi/5+i*t),
            (2*k)*pi/5+t)$(k=0..4);
```

```
L:=L,pavea2(a+l*exp(i*t)+l*exp(i*pi/5+i*t),(2*k)*pi/5+t)$(k=0..4);
L:=L,pavea2(a+l*exp(i*t)+l*exp(i*pi/5+i*t)+l*exp(2*i*pi/5+i*t),
           (2*k+1)*pi/5+t)$(k=0..4);
L:=L,paveb(a+(3+2*r)*exp(i*pi/5+i*t),(2*k)*pi/5+t)$(k=0..4);
//L:=L,paveb(a+l*exp(i*pi/5+i*t),(2*k+1)*pi/5+t)$(k=1..1);
L:=L,paveb(a+(2+r)*exp(2*i*pi/5+i*t)+(1+r)*exp((2*k+1)*i*pi/5+i*t),
          pi+2*k*pi/5+t)$(k=0..0);
L:=L,paveb(a+(2+r)*exp(i*pi/5+i*t),3*pi/5+2*k*pi/5+t)$(k=0..2);
//L:=L,paveb(a+(3+2*r)*exp(i*pi/5+i*t)+(1+r)*exp((-2*k+2)*i*pi/5+i*t),
            -pi/5+2*k*pi/5+t)$(k=0..1);
L:=L,pavea2(a+exp(i*pi/5+i*t),(2*k-1)*pi/5+t)$(k=0..1);
//L:=L,segment(a+(3+2*r)*exp(i*pi/5+i*t)+r*exp(i*pi+2*i*k*pi/5+i*t),
       a+(3+2*r)*exp(i*pi/5+i*t)+(1+r)*exp(i*pi+2*i*k*pi/5+i*t))$(k=0..3);
L:=L,segment(a+(2+r)*exp(i*pi/5+i*t),a+(2+2*r)*exp(i*pi/5+i*t));
L:=L,segment(a+(2+r)*exp(i*t)+exp(2*i*pi/5+i*t),a+(3+r)*exp(i*t)+
            exp(2*i*pi/5+i*t));
B:=point(a+l*exp(i*t)+l*exp(i*pi/5+i*t)+l*exp(2*i*pi/5+i*t)-1);
E:=point(a+l*exp(i*t)+l*exp(i*pi/5+i*t)+l*exp(2*i*pi/5+i*t)+
        exp(7*i*pi/5+i*t));
L:=L,segment(B,B+exp(6*i*pi/5+i*t));
L:=L,segment(E,E+exp(6*i*pi/5+i*t));
L:=L,pavea2(a+l*exp(i*t)+l*exp(i*pi/5+i*t)+l*exp(2*i*pi/5+i*t)+
            exp(6*i*pi/5+i*t),pi)
return L;
}:;
```

On tape :

```
pavehat2(0,0)
```

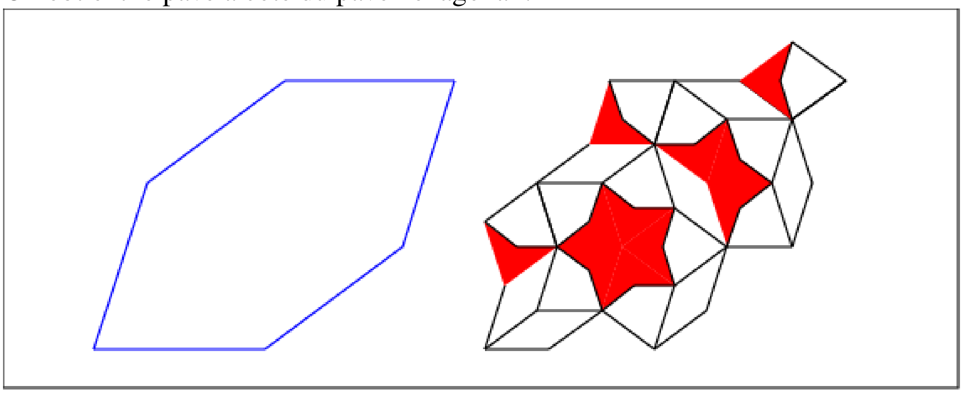
```
pavelo(a,t,c):={
  local L,r,l;
  r:=2*sin(pi/10);l:=2+r;
  L:=NULL;
  L:=L,affichage(paveb(a,t),1+rempli);
  L:=L,pavea(a+exp(i*t)+exp(i*(2*pi/5+t)),pi+t);
    return L;
    }:;
pavelo2(a,t,c):={
  local L,r,l;
  r:=2*sin(pi/10);l:=2+r;
  L:=NULL;
  L:=L,affichage(paveb(a,t),1+rempli);
  L:=L,pavea2(a+exp(i*t)+exp(i*(2*pi/5+t)),pi+t);
    return L;
    }:;
pavelat(a,t):={
   local L,r,l;
  r:=2*sin(pi/10);l:=2+r;
```

```
  L:=NULL;
  L:=L,segment(a,a+exp(i*(t+k*pi/5)))$(k=0..2);;
  L:=L,segment(a+exp(i*t),a+exp(i*t)+exp(i*(t+pi/5)));
  L:=L,pavelo(a+exp(i*(t+2*pi/5)),t+pi/5,1);
  L:=L,pavelo(a+l*exp(i*(t+pi/5)),t-pi/5+2*k*pi/5,1)$(k=0..4);
  L:=L,segment(a+l*exp(i*t),a+l*exp(i*t)+exp(i*(t+pi/5)));
  L:=L,segment(a+l*exp(i*t)+exp(i*(t+pi/5)),
               a+l*exp(i*t)+exp(i*(t+pi/5))+exp(i*(t+2*pi/5)));
  L:=L,pavelo(a+(3+2*r)*exp(i*(t+pi/5)),t+2*k*pi/5,1)$(k=-1..1);
  L:=L,segment(a+l*exp(i*(t+2*pi/5)),a+l*exp(i*(t+2*pi/5))+
               exp(i*(t+pi/5)));
  L:=L,segment(a+l*exp(i*(t+2*pi/5))+l*exp(i*(t+pi/5)),
               a+l*exp(i*(t+2*pi/5))+l*exp(i*(t+pi/5))+exp(i*t));
  L:=L,pavelo(a+l*exp(i*(t+2*pi/5))+exp(i*(t+pi/5)),t,1);
  L:=L,segment(a+(2*l+r)*exp(i*(t+pi/5)),
               a+l*exp(i*(t+2*pi/5))+l*exp(i*(t+pi/5))+exp(i*t));
   L:=L,segment(a+(2*l+r)*exp(i*(t+pi/5)),
                a+(2*l+r+1)*exp(i*(t+pi/5)));
   L:=L,segment(a+(2*l+r)*exp(i*(t+pi/5)),
                a+l*exp(i*t)+l*exp(i*(t+pi/5))+exp(i*(t+2*pi/5)));
L:=L,segment(a+l*exp(i*t)+l*exp(i*(t+pi/5)),
             a+l*exp(i*t)+l*exp(i*(t+pi/5))+exp(i*(t+2*pi/5)));
L:=L,pavelo(a+exp(i*t)+l*exp(i*(t+pi/5))+l*exp(i*(t+2*pi/5)),
             t-pi/5,1);
return L;
    }:;
```

On tape :

`affichage(hexal(-6,0,1),4),pavelat(0,0)`

On obtient le pave à còté du pavé hexagonal :



On tape :

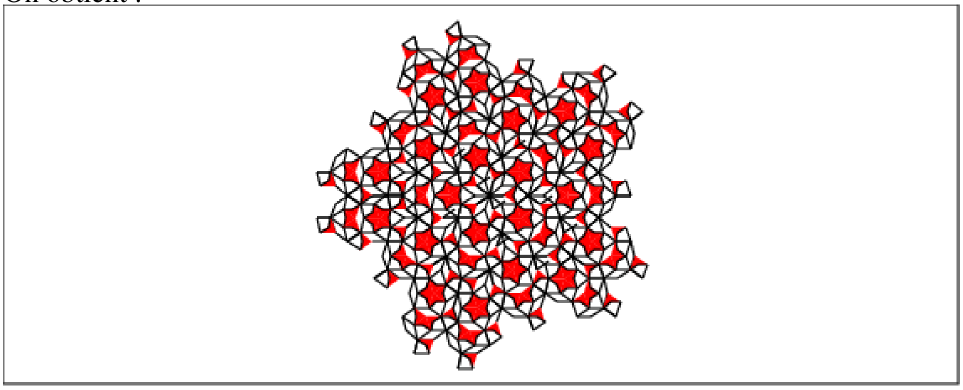
```
etoile10():={
local L;
L:=NULL;
L:=L,pavelat(0,2*k*pi/5)$(k=0..4);
```

```
L:=L,pavelat(l*exp(2*i*k*pi/5),-pi/5+2*k*pi/5)$(k=0..4);
L:=L,pavelat(l*exp(2*i*k*pi/5)+l*exp(i*(2*k+1)*pi/5),
              2*k*pi/5)$(k=0..4);
L:=L,pavelat(l*exp(2*i*(k+1)*pi/5)+l*exp(i*(2*k+1)*pi/5),
             2*k*pi/5)$(k=0..4);
return L;}:;
```

On tape :

```
etoile10()
```

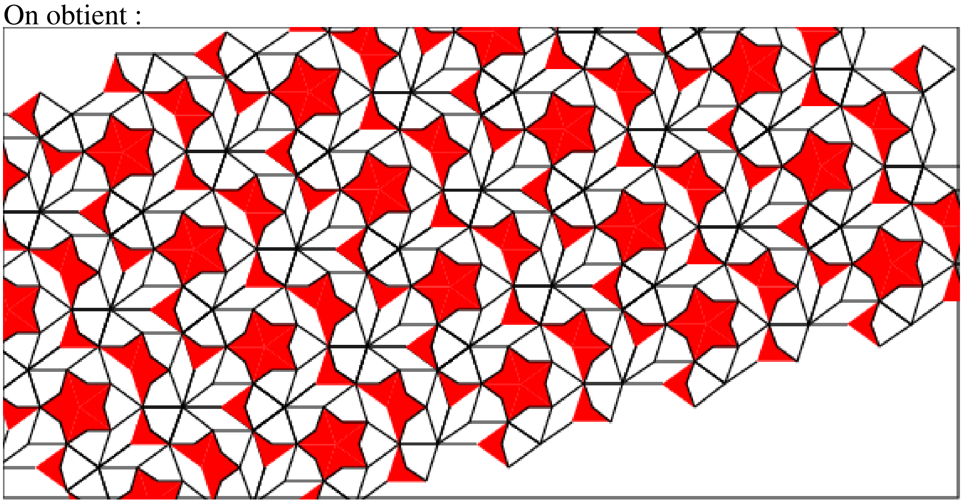
On obtient :



On tape :

```
r:=2*sin(pi/10);l:=2+r;pavelat(k*(l+l*exp(i*pi/5)),0)$(k=0..4),
pavelat(l*exp(2*i*pi/5)-l+k*(l+l*exp(i*pi/5)),0)$(k=0..4),
pavelat(2*l*exp(2*i*pi/5)-2*l+k*(l+l*exp(i*pi/5)),0)$(k=0..4),
pavelat(3*l*exp(2*i*pi/5)-3*l+k*(l+l*exp(i*pi/5)),0)$(k=0..4)
```

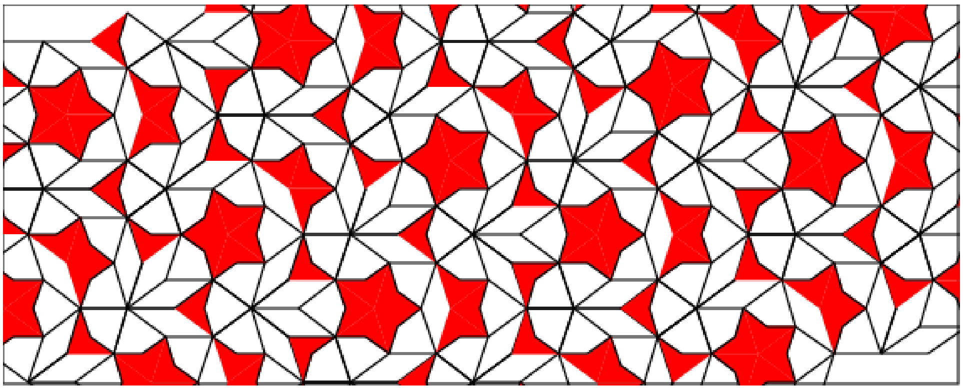
On obtient :



On tape :

```
r:=2*sin(pi/10);l:=2+r;
pavelat(k*(l+l*exp(i*pi/5)),-pi/5)$(k=0..4),
pavelat(-l+k*(l+l*exp(i*pi/5)),0)$(k=0..4),
pavelat(l*(-2+exp(4*i*pi/5)+exp(2*i*pi/5))+k*l*(1+exp(i*pi/5)),
-pi/5) $(k=0..4)
pavelat(l*(1+exp(-i*pi/5)+exp(-3*i*pi/5))+k*l(1+exp(i*pi/5)),0)
$(k=0..4)
```

On obtient (ce n'est pas tout à fait le même dessin !) :



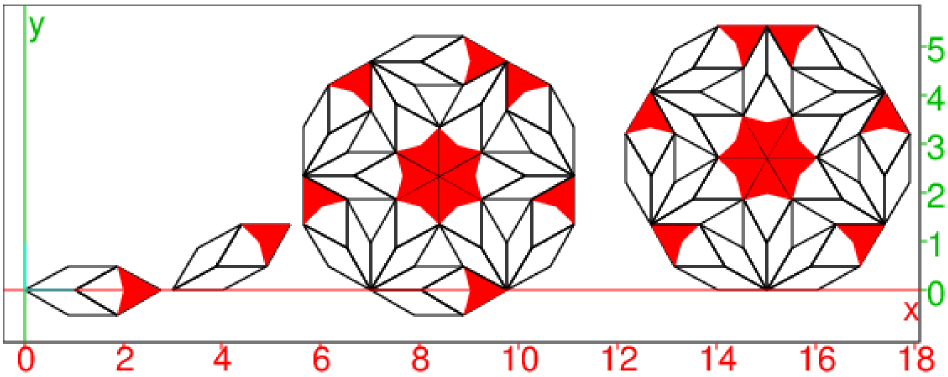
1.13 Autres pavages

1.13.1 Avec un triangle équilatèral et un dodćagone non régulier

`forme(a,t)` sert à remplir le dodécagone `pavex(a,t)`.
On tape :

```
forme(a,t):={
  local L;
  L:=NULL;
  L:=L,losange(a,a+exp(i*t),pi/6);
  L:=L,losange(a,a+exp(i*t),-pi/6);
  L:=L,losange(a+exp(i*t),a+exp(i*t)+exp(-i*(pi/6-t)),pi/3);
  L:=L,affichage(polygone(a+exp(i*t)*(1+sqrt(3)),
        a+exp(i*t)*(sqrt(3)/2+1+i*1/2),a+2*exp(i*t),
        a+exp(i*t)*(sqrt(3)/2+1-i*1/2)),1+rempli);
  return(L);
  }:;
pavehex(a,t):={
  local L;
  L:=NULL;
  L:=L,forme(a,t+k*pi/3)$(k=0..2);
  L:=L,forme(a+(1+sqrt(3))*exp(i*t),t+k*pi/3)$(k=1..2);
  L:=L,forme(a+(1+sqrt(3))*(1+exp(i*pi/3))*exp(i*t),
              t+2*pi/3+k*pi/3)$(k=0..1);
  L:=L,forme(a+(1+sqrt(3))*exp(2*i*pi/3)*exp(i*t),
              t+k*pi/3)$(k=0..1);
  L:=L,forme(a+i*(3+sqrt(3))*exp(i*t),t-k*pi/3)$(k=0..1);
  L:=L,forme(a+(1+sqrt(3)+i*(3+sqrt(3)))*exp(i*t),t-2*pi/3)
return L;
}:;
```

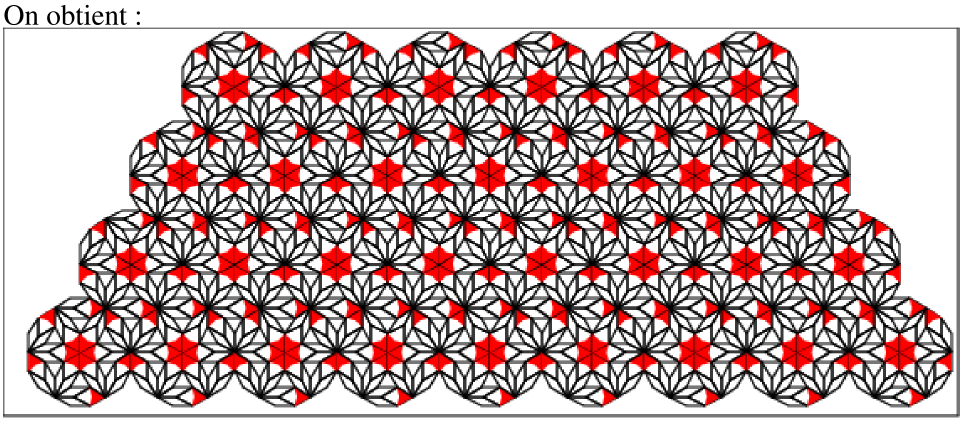
On tape :
`forme(0,0),forme(3,pi/6),pavehex(7,0),pavehex(15,pi/6)`
On obtient :



On tape :

```
pavehex(k*(2+2*sqrt(3)),0)$(k=0..8),
pavehex((2+2*sqrt(3))*(k+exp(i*pi/3)),0)$(k=0..7),
pavehex(2*(2+2*sqrt(3))*exp(i*pi/3)+k*(2+2*sqrt(3)),0)$(k=0..6),
pavehex(3*(2+2*sqrt(3))*exp(i*pi/3)+k*(2+2*sqrt(3)),0)$(k=0..5)
```

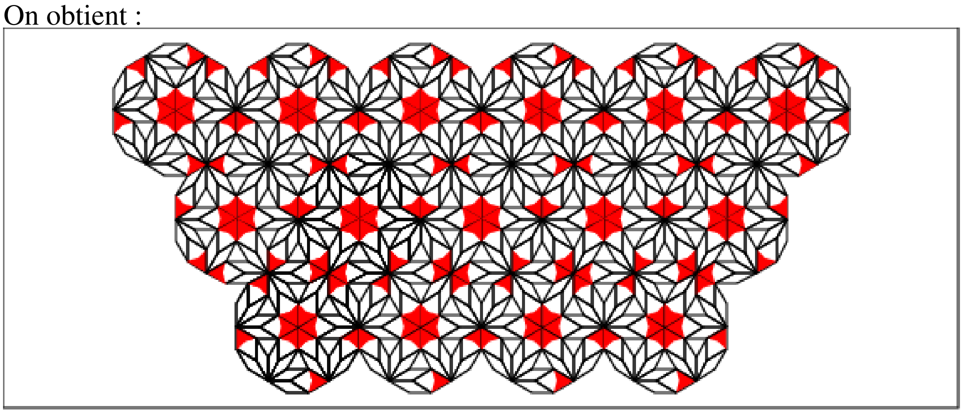
On obtient :



On tape :

```
pavehex(k*(2+2*sqrt(3)),0)$(k=0..3),
pavehex(k*(2+2*sqrt(3))+(4+4*sqrt(3))*exp(i*pi/3),pi)$(k=-1..3),
pavehex(i*(6+2*sqrt(3))+k*(2+2*sqrt(3)),0)$(k=-1..4)
```

On obtient :

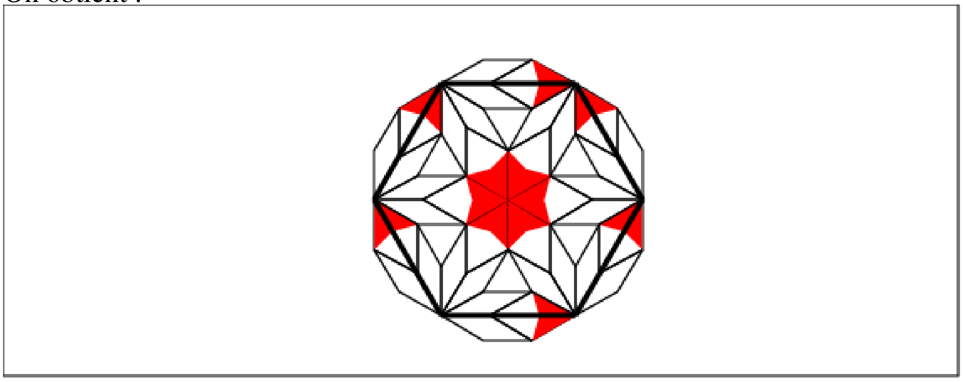


1.13.2 Avec un triangle équilatèral et un hexagone régulier

On utilise les mêmes procédures que précédement en considérant dans le dodécagone l'hexagone ci dessous.
On rajoute un choix de couleurs pour la forme (`formec(a,t,c1)`) et pour les triangles équilatéraux de l'hexagone (`T(a,t,c2)`).
Pour le pavé initial on aura `pavehexac(a,t,c1,c2)`.
Ici Le pavage se fera donc avec le pavé de droite dans lequel on a supprimé les morceaux qui vont être recouverts (c'est `pavehexagc(a,t,c1,c2)`).
On tape :

```
pavehexc(0,0,1,4), affichage(hexagone(0,1+sqrt(3)),epaisseur_ligne_3),
pavehexagc(7,0,1,4)
```

On obtient :



On met un peu de couleur en rajoutant un paramètre pour la couleur de la "forme" et en colorant les triangles équilatéraux de l'hexagone.
On tape :

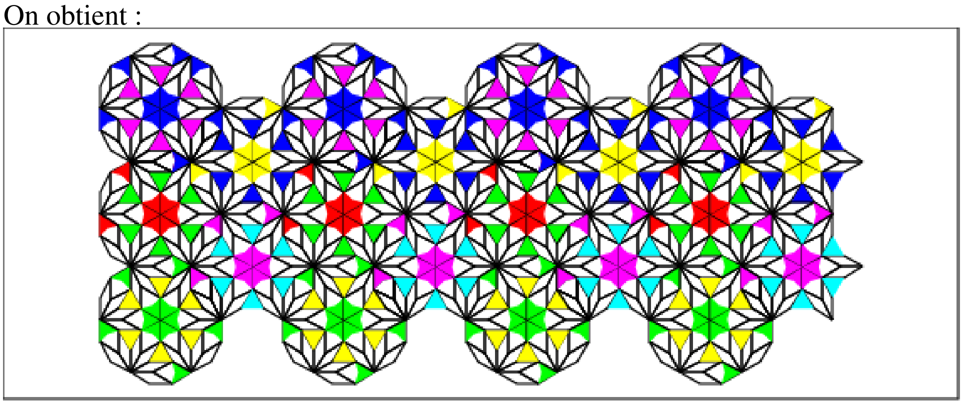
```
formec(a,t,c1):={
  local L;
  L:=NULL;
  L:=L,losange(a,a+exp(i*t),pi/6);
  L:=L,losange(a,a+exp(i*t),-pi/6);
  L:=L,losange(a+exp(i*t),a+exp(i*t)+exp(-i*(pi/6-t)),pi/3);
  L:=L,affichage(polygone(a+exp(i*t)*(1+sqrt(3)),
                 a+exp(i*t)*(sqrt(3)/2+1+i*1/2),a+2*exp(i*t),
                 a+exp(i*t)*(sqrt(3)/2+1-i*1/2)),c1+rempli)
  return(L);
  }:;
T(a,t,c2):=affichage(triangle_equilateral(a,a+exp(i*t)),rempli+c2):;
pavehexac(a,t,c1,c2):={
  local L;
  L:=NULL;
  L:=L,formec(a,t+k*pi/3,c1)$(k=0..2);
  L:=L,formec(a+(1+sqrt(3))*exp(i*t),t+k*pi/3,c1)$(k=1..2);
  L:=L,formec(a+(1+sqrt(3))*(1+exp(i*pi/3))*exp(i*t),
              t+2*pi/3+k*pi/3,c1)$(k=0..1);
  L:=L,formec(a+(1+sqrt(3))*exp(2*i*pi/3)*exp(i*t),t+k*pi/3,c1)
              $(k=0..1);
  L:=L,formec(a+i*(3+sqrt(3))*exp(i*t),t-k*pi/3,c1)$(k=0..1);
```

```
  L:=L,formec(a+(1+sqrt(3)+i*(3+sqrt(3)))*exp(i*t),t-2*pi/3,c1);
  L:=L,T(a+(1+sqrt(3))*exp(i*(t+pi/3))+exp(i*t+i*pi/6+k*i*pi/3),
         t+k*pi/3,c2)$(k=0..5);
  return L;
}:;
pavehexagc(a,t,c1,c2):={
  local L;
  L:=NULL;
  L:=L,formec(a,t+k*pi/3,c1)$(k=1..2);
  L:=L,formec(a+(1+sqrt(3))*exp(i*t),t+k*pi/3,c1)$(k=2..2);
  L:=L,formec(a+(1+sqrt(3))*(1+exp(i*pi/3))*exp(i*t),t+2*pi/3+k*pi/3,c1)$(k=1
  L:=L,formec(a+(1+sqrt(3))*exp(2*i*pi/3)*exp(i*t),t+k*pi/3,c1)$(k=0..1);
  L:=L,formec(a+i*(3+sqrt(3))*exp(i*t),t-k*pi/3,c1)$(k=0..1);
  L:=L,formec(a+(1+sqrt(3)+i*(3+sqrt(3)))*exp(i*t),t-2*pi/3,c1);
  L:=L,T(a+(1+sqrt(3))*exp(i*(t+pi/3))+exp(i*t+i*pi/6+k*i*pi/3),t+k*pi/3,c2)$
  return L;
}:;
```

On tape :

```
pavehexagc(k*(3+3*sqrt(3)),0,1,2)$(k=0..3),
pavehexac(i*(3+sqrt(3))+k*(3+3*sqrt(3)),0,4,5)$(k=0..3),
pavehexac(-i*(3+sqrt(3))+k*(3+3*sqrt(3)),0,2,3)$(k=0..3),
pavehexagc((1+sqrt(3))*(1+exp(i*pi/3))+k*(3+3*sqrt(3)),0,3,4)$(k=0..3),
pavehexagc((1+sqrt(3))*(1+exp(-i*pi/3))+k*(3+3*sqrt(3)),0,5,6)$(k=0..3)
```

On obtient :

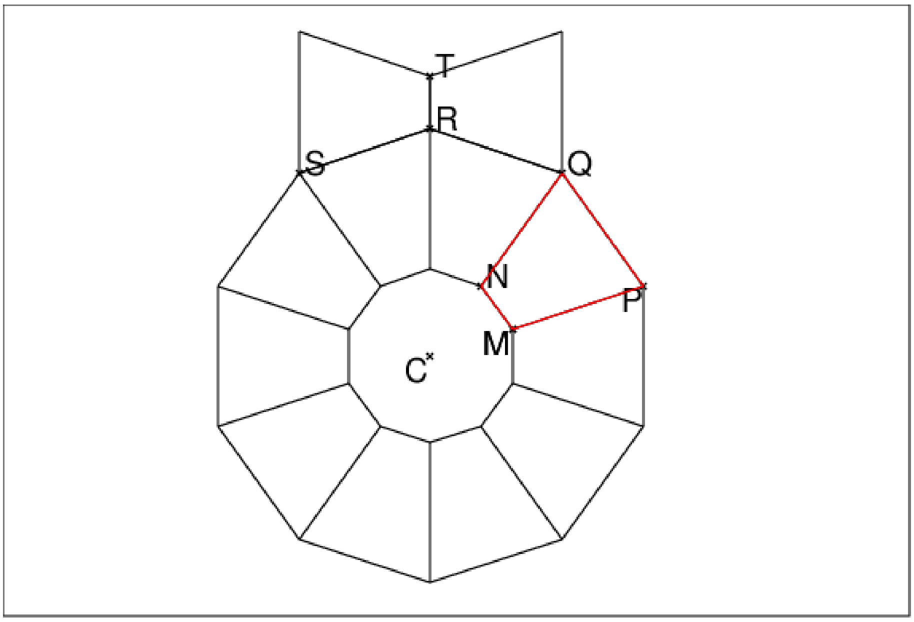


1.14 Un pavage avec un décagone et un trapèze équilatèral

1.14.1 Le pavé

Définition

On dit qu'un trapèze $ABCD$ est équilatèral d'angle α si $AB = BC = DA$ et si $\alpha = (\overrightarrow{AB}, \overrightarrow{AD}$ (DC est alors son petit côté).
Le pavé est constitué d'un décagone de centre `C` et de sommet `M`, de côté `MN` et d'un trapèze équilatèral $PQNM$ d'angle `2*pi/5`(en rouge).
On a `longueur(M,N)=2*longueur(C,A)*sin(pi/10)`.
Voici ce pavé (`C,A,M` (resp `C,B,N`) sont alignés) :



La longueur `c` du petit côté du trapèze est égal au côté du décagone est :
`c:=2*longueur(C,A)*sin(pi/10)`
et son angle $(\overrightarrow{MN}, \overrightarrow{MA})$ vaut `2*pi/5`.

Attention

Pour simplifier les programmes on va supposer que `longueur(C,A)=1`.

Soit le trapèze direct équilatèral $NMPQ$ de petit côté NM.
On a :
$l = MP = PQ$ et $PQ/2 = CP{*}\sin(pi/10)$ donc puisque on a supposé $CM = 1$,
$l = 2(1 + l)\sin(pi/10) =$ donc $l = 2\sin(pi/10)/(1 - 2\sin(pi/10))$
On tape le progromme du trapèze direct équilatèral $NMPQ$ de petit côté NM :

```
trapezequi(N,M):={
  local P,Q,l,c;
  c:=longueur(N,M);
  l:=c/(1-2*sin(pi/10));
  P:=similitude(M,l/c,-3*pi/5,N);
  Q:=similitude(N,l/c,3*pi/5,M);
  retourne polygone(N,M,P,Q);
  }:;
```

```
pave(C,M):={
  local L,N,Q,P,R,R,S,T,r,c,l;
  L:=NULL;
  r:=longueur(C,M);
  c:=2*r*sin(pi/10);
  l:=2*r*sin(pi/10)/(1-2*sin(pi/10));
  L:=L,isopolygone(C,M,-10);
  P:=homothetie(C,(1+2*sin(pi/10)/(1-2*sin(pi/10))),M);
  L:=L,isopolygone(C,P,-10);
  L:=L,segment(rotation(C,k*pi/5,M),rotation(C,k*pi/5,P))$(k=0..9);
  Q:=rotation(C,pi/5,P);
```

```
  R:=rotation(C,pi/5,Q);
  S:=rotation(C,pi/5,R);
  T:=translation((R-C)*c/(l+r),R);
  L:=L,trapezequi(T,R);
  L:=L,trapezequi(R,T);
  retourne L;
}:;
```

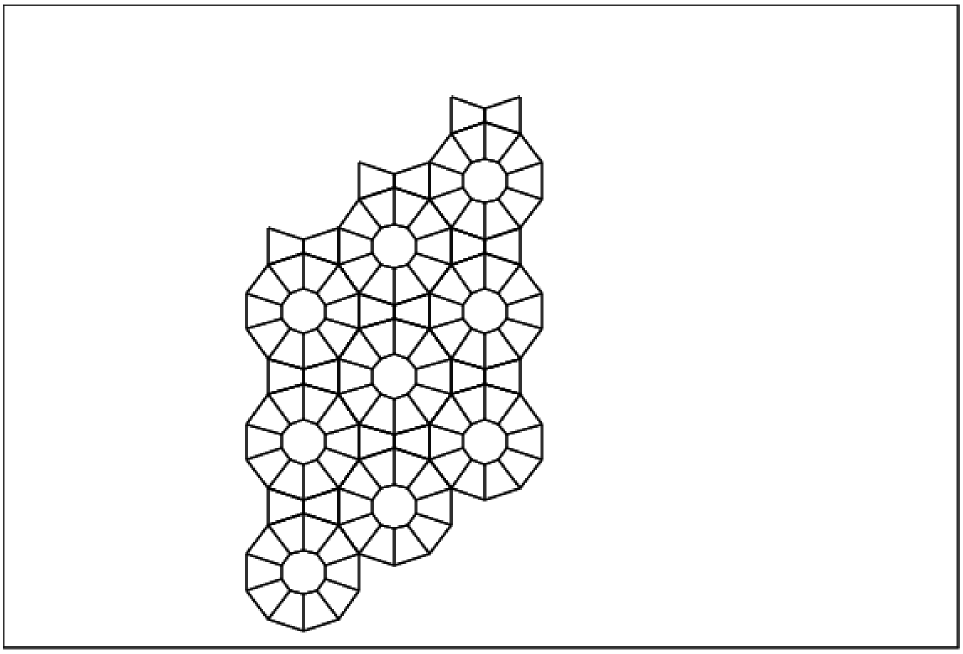
On tape :
`pave(point(0),point(exp(i*pi/10)))`
On obtient le pavé ci dessus.

1.14.2 Le pavage

Soit :
`I:=milieu(P,Q).`
Le pavage se fabrique avec 2 translations :
l'une de vecteur $2\overrightarrow{CI}$ et
l'autre de vecteur $\overrightarrow{CR}+\overrightarrow{CT}$
On tape

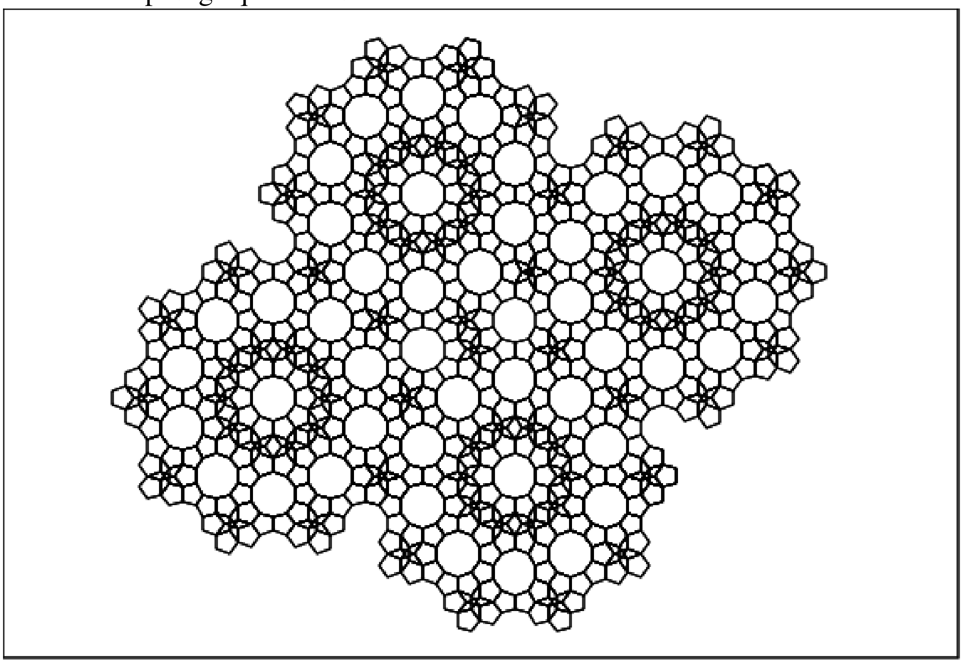
```
pavage(C,M):={
  local L,N,Q,P,R,R,S,T,r,c,l,I,C1,M1;
  L:=NULL;
  r:=longueur(C,M);
  c:=2*r*sin(pi/10);
  l:=2*r*sin(pi/10)/(1-2*sin(pi/10));
  P:=homothetie(C,(1+2*sin(pi/10)/(1-2*sin(pi/10))),M);
  Q:=rotation(C,pi/5,P);
  R:=rotation(C,pi/5,Q);
  S:=rotation(C,pi/5,R);
  T:=translation((R-C)*c/(l+r),R);
  I:=milieu(P,Q);
  L:=L,pave(translation(2*k*(I-C),C),translation(2*k*(I-C),M))$(k=0..2);
  C1:=translation(((R-C)+(T-C)),C);
  M1:=translation(((R-C)+(T-C)),M);
  L:=L,pave(translation(2*k*(I-C),C1),translation(2*k*(I-C),M1))$(k=0..2);
  C2:=translation(((R-C)+(T-C)),C1);
  M2:=translation(((R-C)+(T-C)),M1);
  L:=L,pave(translation(2*k*(I-C),C2),translation(2*k*(I-C),M2))$(k=0..2);
  retourne L;
}:;
```

On tape :
`pavage(point(0),point(exp(i*pi/10)))`
On obtient :



1.15 Pavage de quasicristaux

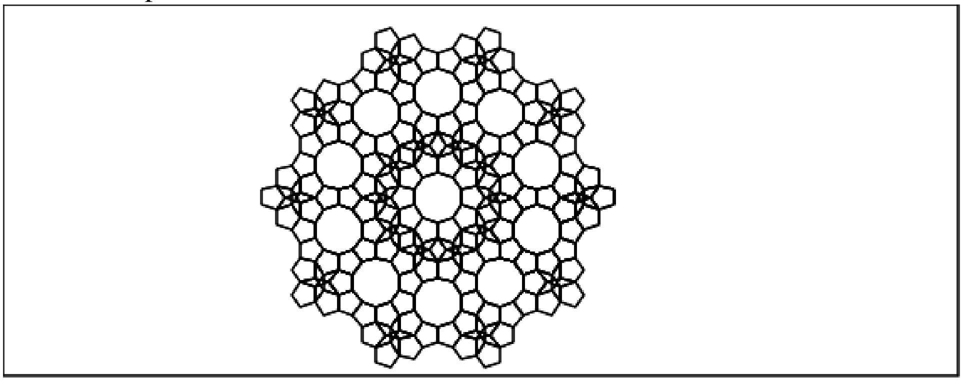
Voici le pavage que l'on veut réalisé :



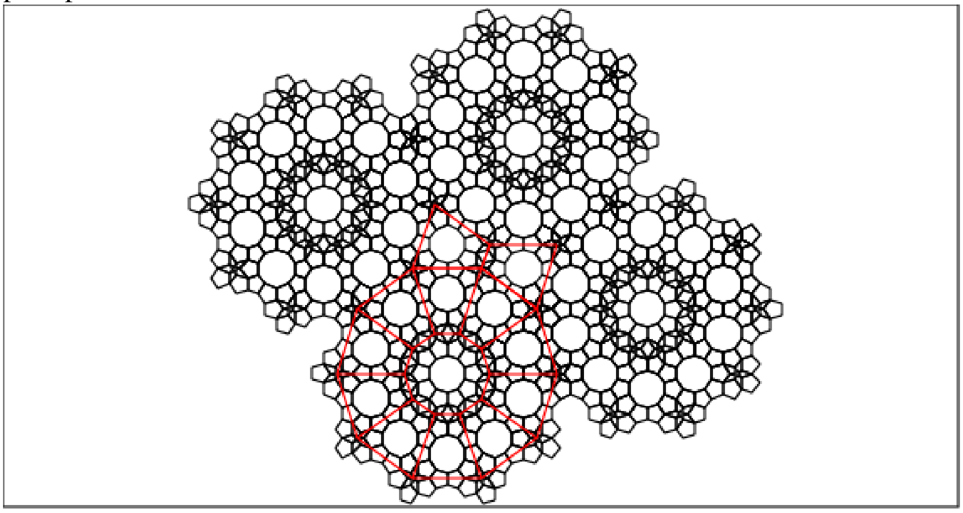
Ce pavage ressemble au pavé précédent avec en plus des étoiles et des roues polygonales.

1.15.1 Le pavé

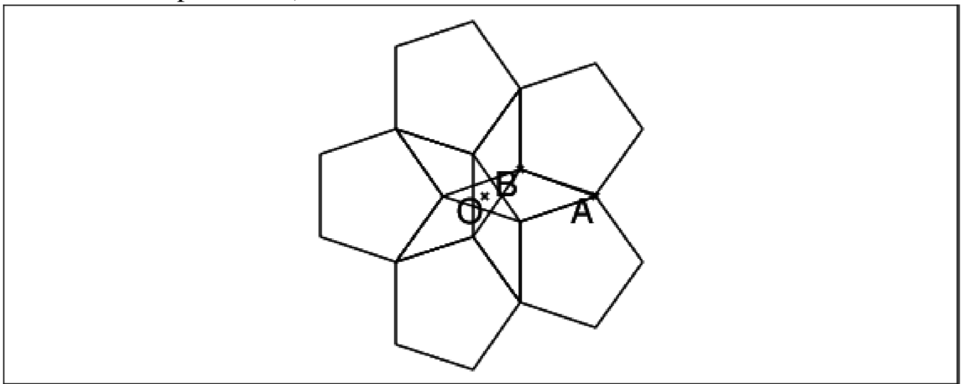
Voici le pavé :



Ce pavé ressemble au pavé précédent car les étoiles sont situées aux sommets du pavé précédent.



On écrit la procédure `etoiles(O,A)` qui dessine (le point `B` est une variable locale de cette procédure) :

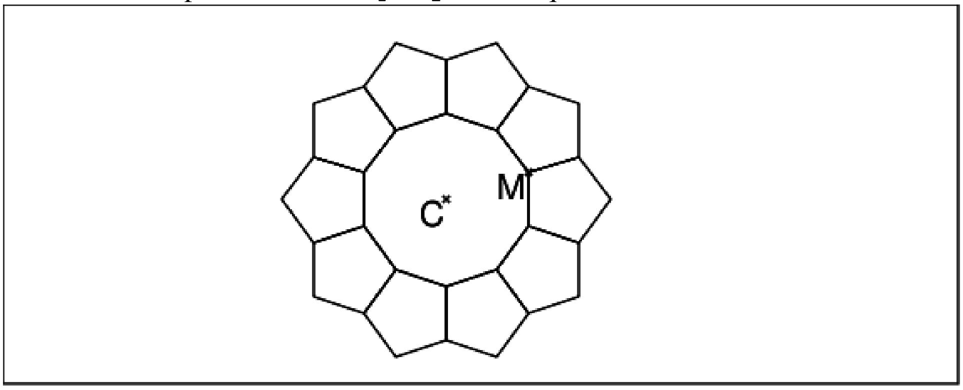


On tape :

```
etoiles(O,A):={
local P,L,B,Q,R,r,k;
   r:=evalf(longueur(O,A));
   P:=isopolygone(O,A,-5);
```

```
   B:=inter_unique(segment(A,sommets(P)[2]),
                    segment(sommets(P)[1],sommets(P)[4]));
   Q:=isopolygone(O,B,-5);
   L:=polygone(sommets(Q)[0],sommets(Q)[2],sommets(Q)[4],
               sommets(Q)[1],sommets(Q)[3]);
   R:=segment(A,B),segment(B,sommets(P)[1]);
   L:=L,rotation(O,2k*pi/5,isopolygone(B,A,5))$(k=0..4);
   L:=L,rotation(O,2k*pi/5,segment(A,B))$(k=1..5);
   L:=L,rotation(O,2k*pi/5,segment(B,sommets(P)[1]))$(k=1..5);
   retourne L;
  }:;
```

Dans la suite `C` est le centre de la roue et `M` est un sommet du décagone de centre `C`.
Puis on écrit la procédure `rouepoly(C,M)` qui dessine :

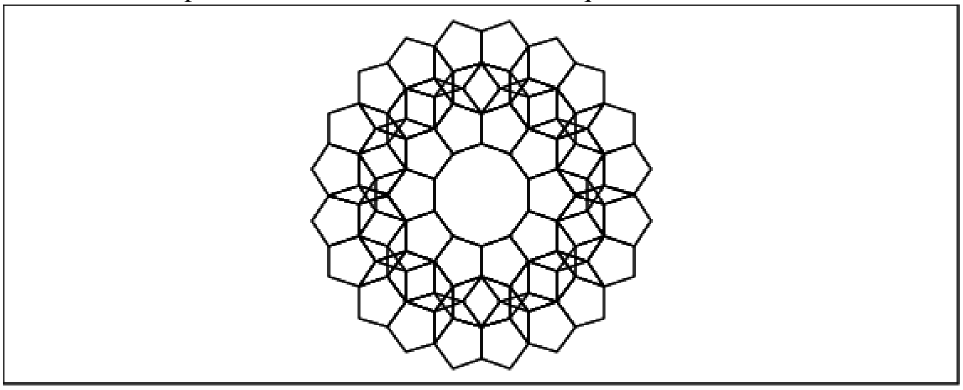


lorsque `C:=point(0);M:=point(exp(i*pi/10));`
On tape :

```
rouepoly(C,M):={
local L,D,K,Q,I,c,c1,c2,r,r1,r2,h;
L:=NULL;
L:=L,isopolygone(C,M,-10);
Q:=rotation(C,pi/5,M);
r:=evalf(longueur(C,M));
c1:=2*r*sin(pi/10);
h:=r*cos(pi/10);
r1:=c1/(2*sin(pi/5));
c2:=2*r1*sin(2*pi/5);
r2:=c2/(2*sin(pi/5));
I:=milieu(Q,M);
D:=homothetie(C,(h+r1*cos(pi/5)+r1*cos(2*pi/5)+r2*cos(pi/5))/h,I);
K:=evalf(homothetie(C,(h+c1*cos(pi/10)+r2*cos(pi/5)+r2)/h,I));
pour k de 1 jusque 10 faire
   L:=L,isopolygone(Q,M,5);
   M:=Q;
   Q:=rotation(C,pi/5,M);
fpour;
retourne L;
```

```
}:;
```

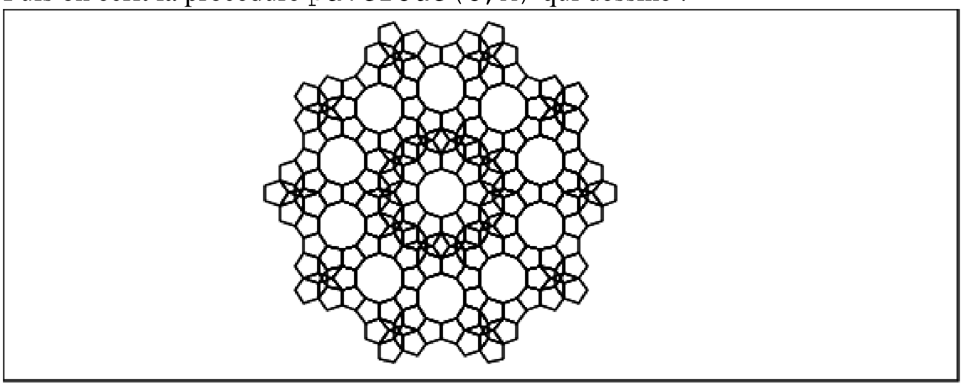
Puis on écrit la procédure `rouetoiles(C,M)` qui dessine :



lorsque `C:=point(0);M:=point(exp(i*pi/10));`
On tape :

```
rouetoiles(C,M):={
local L,D,K,Q,I,c,c1,c2,r,r1,r2;LD,LK;
L:=NULL;
L:=L,isopolygone(C,M,-10);
Q:=rotation(C,pi/5,M);
r:=evalf(longueur(C,M));
c1:=2*r*sin(pi/10);
h:=r*cos(pi/10);
r1:=c1/(2*sin(pi/5));
c2:=2*r1*sin(2*pi/5);
r2:=c2/(2*sin(pi/5));
I:=milieu(Q,M);
D:=homothetie(C,(r*cos(pi/10)+r1*cos(pi/5)+r1*cos(2*pi/5)+
               r2*cos(pi/5))/(r*cos(pi/10)),I);
K:=homothetie(C,(r*cos(pi/10)+c1*cos(pi/10)+
               r2*cos(pi/5)+r2)/(r*cos(pi/10)),I);
LD:=rotation(C,k*pi/5,D)$(k=0..9);
LK:=rotation(C,k*pi/5,K)$(k=0..9);
L:=L,etoiles(LD[k],LK[k])$(k=0..9);
retourne L;
}:;
```

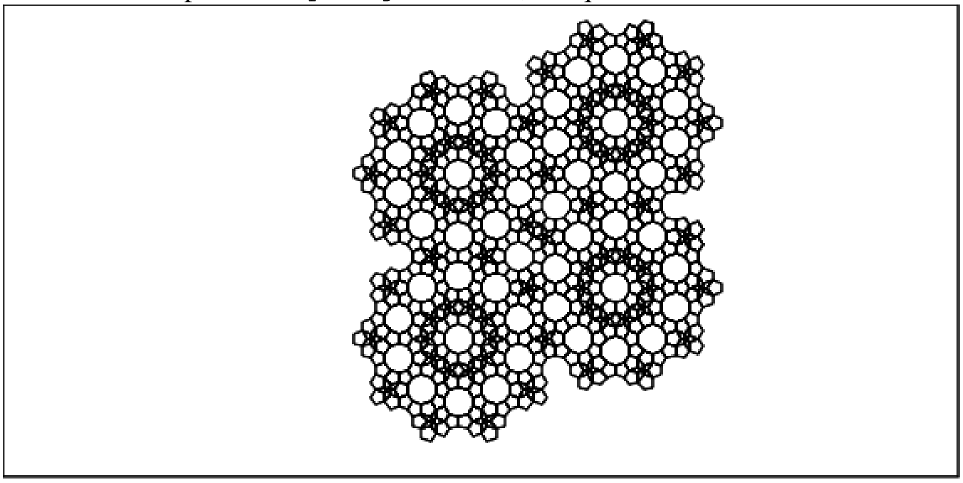
Puis on écrit la procédure `paveroue(C,M)` qui dessine :



lorsque `C:=point(0);M:=point(exp(i*pi/10));`
On tape :

```
paveroue(C,M):={
local D,D1,K,k,L,Q,C1,M1,K1,C2,r,c1,h,r1,c2,r2,I,l,LD,LK,LC1,LM1,LD1,L
L:=NULL;
L:=L,isopolygone(C,M,-10);
Q:=rotation(C,pi/5,M);
r:=evalf(longueur(C,M));
c1:=2*r*sin(pi/10);
h:=r*cos(pi/10);
r1:=c1/(2*sin(pi/5));
c2:=2*r1*sin(2*pi/5);
r2:=c2/(2*sin(pi/5));
I:=milieu(Q,M);
//l:=evalf(longueur(C,K));
l:=r*cos(pi/10)+c1*cos(pi/10)+r2*cos(pi/5)+r2;
D:=homothetie(C,(r*cos(pi/10)+r1*cos(pi/5)+r1*cos(2*pi/5)+
              r2*cos(pi/5))/(r*cos(pi/10)),I);
K:=homothetie(C,(r*cos(pi/10)+c1*cos(pi/10)+r2*cos(pi/5)+
              r2)/(r*cos(pi/10)),I);
LD:=rotation(C,k*pi/5,D)$(k=0..9);
LK:=rotation(C,k*pi/5,K)$(k=0..9);
L:=L,etoiles(LD[k],LK[k])$(k=0..9);
C1:=homothetie(C,(2*r+2*c1+c2)/r,M);
M1:=homothetie(C,(3*r+2*c1+c2)/r,M);
D1:=homothetie(C,(l+2*h+r2)/(l-r2),D);
K1:=homothetie(C,(l+2*h)/l,K);
L:=L,rouepoly(C1,M1);
L:=L,etoiles(D1,K1);
LC1:=rotation(C,k*pi/5,C1)$(k=0..9);
LM1:=rotation(C,k*pi/5,M1)$(k=0..9);
L:=L,rouepoly(LC1[k],LM1[k])$(k=0..9);
LD1:=rotation(C,k*pi/5,D1)$(k=0..9);
LK1:=rotation(C,k*pi/5,K1)$(k=0..9);
L:=L,etoiles(LD1[k],LK1[k])$(k=0..9);
retourne L;
}:;
```

Puis on écrit la procédure `pavageroue(C,M)` qui dessine :



lorsque `C:=point(0);M:=point(exp(i*pi/10));`
On tape :

```
pavageroue(C,M):={
local L,r,c1,c2,h,I,Q,TC1,TC2,M1,M2,d,d1,TC3,M3;
r:=evalf(longueur(C,M));
c1:=2*r*sin(pi/10);
h:=r*cos(pi/10);
r1:=c1/(2*sin(pi/5));
c2:=2*r1*sin(2*pi/5);
r2:=c2/(2*sin(pi/5));
Q:=rotation(C,pi/5,M);
I:=milieu(Q,M);
d:=6*r+5*c1+2*c2;
TC1:=homothetie(C,d/r,M);
M1:=homothetie(C,d/r+1,M);
L:=NULL;
L:=L,paveroue(C,M);
L:=L,paveroue(TC1,M1);
TC2:=TC1*exp(2*i*pi/5);
M2:=M1*exp(2*i*pi/5);
L:=L,paveroue(TC2,M2);
d1:=10*r+8*c1+3*c2;
TC3:=homothetie(C,d1/r,Q);
M3:=homothetie(C,d1/r+1,Q);
L:=L,paveroue(TC3,M3);
retourne L;
}:;
```

1.16 Droites remarquables du triangle

1.16.1 Définitions

Étant donné trois points `A`, `B` et `C` les commandes suivantes permettent de tracer les droites remarquables du triangle ABC :

— Les médianes
`mediane(A,B,C)` trace la médiane du triangle ABC issue de A,
`mediane(B,C,A)` trace la médiane du triangle ABC issue de B,
`mediane(C,B,A)` trace la médiane du triangle ABC issue de C.

— Les bissectrices
`bissectrice(A,B,C)` trace la bissectrice intérieure de l'angle A,
`exbissectrice(A,B,C)` trace la bissectrice extérieure de l'angle A,
`bissectrice(B,A,C)` trace la bissectrice intérieure de l'angle B,
`exbissectrice(B,A,C)` trace la bissectrice extérieure de l'angle B,
`bissectrice(C,B,A)` trace la bissectrice intérieure de l'angle C,
`exbissectrice(C,B,A)` trace la bissectrice extérieure de l'angle C.

— Les hauteurs
`hauteur(A,B,C)` trace la hauteur du triangle ABC issue de A,
`hauteur(B,A,C)` trace la hauteur du triangle ABC issue de B,
`hauteur(C,B,A)` trace la hauteur du triangle ABC issue de C.

— Les médiatrices
`mediatrice(A,B)` trace la médiatrice de AB,
`mediatrice(B,C)` trace la médiatrice de BC,
`mediatrice(A,C)` trace la médiatrice de AC

1.16.2 Exercice

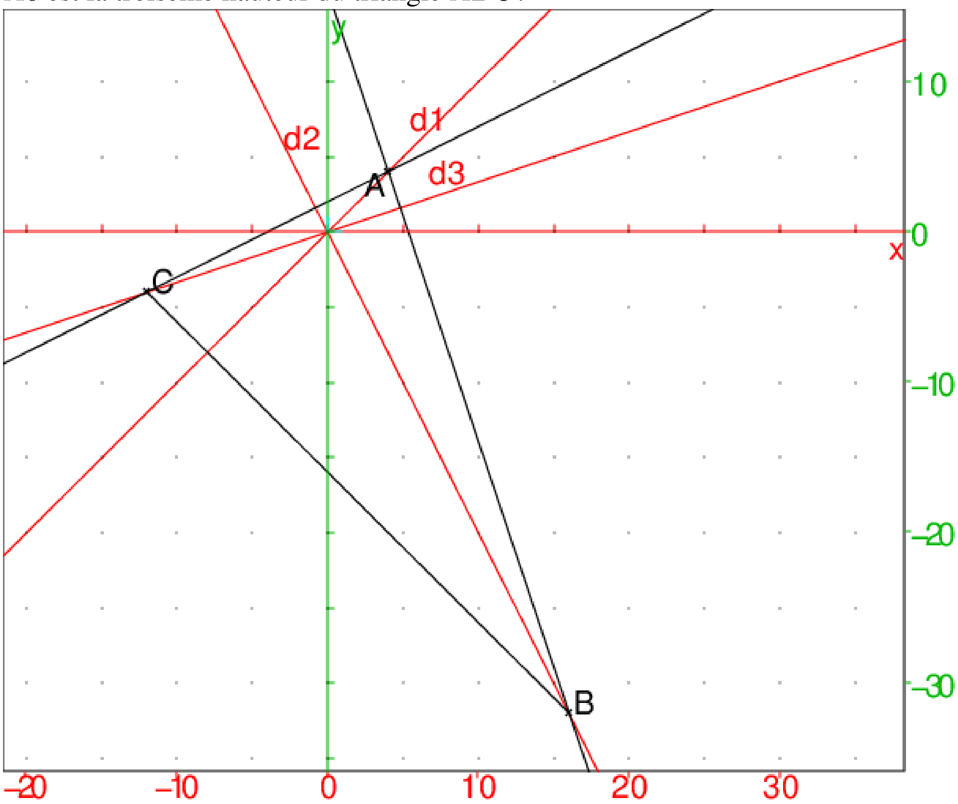
Soient d_1 d'équation $y = x$, d_2 d'équation $y = -2x$ et d_3 d'équation $y = x/3$.

1. Construire un triangle ABC qui admet d_1 comme hauteur h_A, d_2 comme hauteur h_B et d_3 comme hauteur h_C.
2. Construire un triangle ABC qui admet d_1 comme médiane m_A, d_2 comme médiane m_B et d_3 comme médiane m_C.
3. Construire un triangle ABC qui admet d_1 comme médiatrice m_{BC}, d_2 comme médiatrice m_{AC} et d_3 comme médiatrice m_{AB}.
4. Construire un triangle ABC qui admet d_1 comme bissectrice b_A, d_2 comme bissectricehauteur b_B et d_3 comme bissectrice b_C.
5. Construire un triangle ABC qui admet d_1 comme bissectrice b_A, d_2 comme bissectrice extérieure be_B et d_3 comme bissectrice extérieure be_C.

Une solution avec `Xcas`

1. Construction de ABC admettant d_1, d_2, d_3 comme hauteurs.
   On définit les 3 droites, puis on doit avoir :
   CB perpendiculaire à d_1, AC perpendiculaire à d_2 et AB perpendiculaire à d_3.
   On choisit un point A sur $d1$, puis on trace la perpendiculaire p_2 à d_2 et p_3

à d_3. p_2 coupe d_3 en C et p_3 coupe d_2 en B.
d_2 et d_3 sont 2 hauteurs du triangle ABC qui se coupent en O.
Dans un triangle les 3 hauteurs sont concourantes donc d_1 qui est la droite $A0$ est la troisème hauteur du triangle ABC.



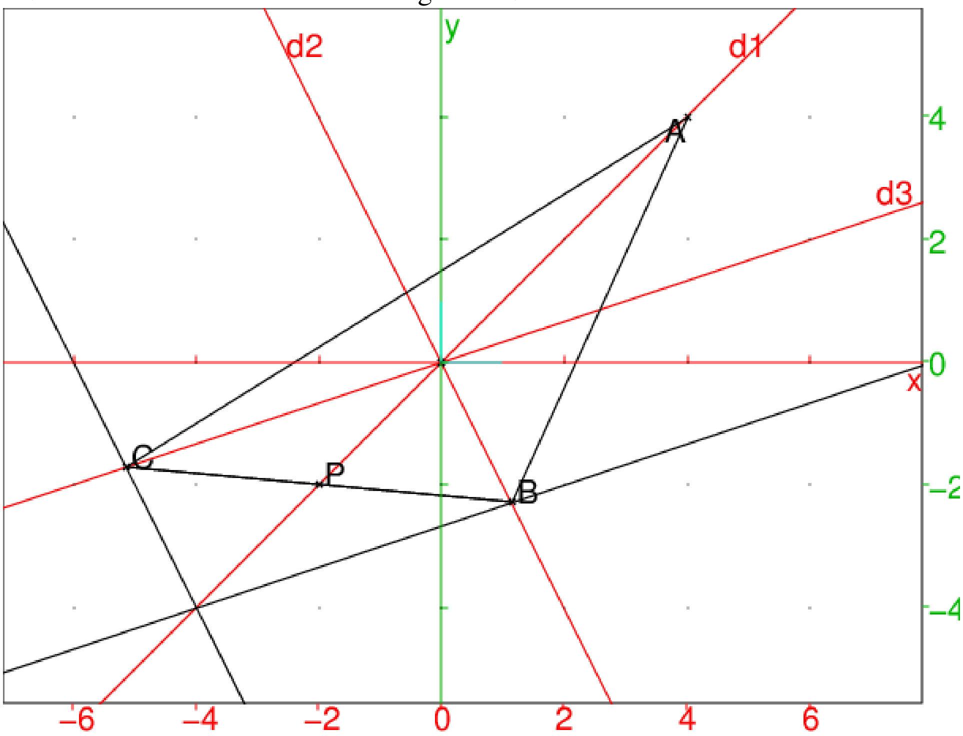
Pour obtenir cette construction, on tape :

```
d1:=droite(0,5+5*i,affichage=1);
d2:=droite(0,-2+4*i,affichage=1);
d3:=droite(0,6+2*i,affichage=1);
A:=point(4+4*i);
p2:=perpendiculaire(A,d2):;p2;
C:=inter_unique(p2,d3);
p3:=perpendiculaire(A,d3):;p3;
B:=inter_unique(p3,d2);
s:=segment(B,C);
```

On tape :
`est_perpendiculaire(s,d1);`
On obtient : `1`

2. Construction de ABC admettant d_1, d_2, d_3 comme médianes.
On doit avoir :
B est sur d_2 et le milieu M de AB doit être sur d_3, donc B se trouve aussi sur la droite d_4 image de d_3 par l'homothétie de centre A et de rapport 2.
De même, C est sur d_3 et le milieu N de AC doit être sur d_2, donc C se trouve aussi sur la droite d_5 image de d_2 par l'homothétie de centre A et de rapport 2.
On choisit un point A sur $d1$, puis on trace la droite d_4 (resp d_5) image de

d_3 (resp d_2) par l'homothétie de centre A et de rapport 2.
et d_5 coupe d_3 en C et d_4 coupe d_2 en B.
d_2 et d_3 sont 2 médianes du triangle ABC qui se coupent en O.
Dans un triangle les 3 médianes sont concourantes donc d_1 qui est la droite $A0$ est la troisème médiane du triangle ABC.



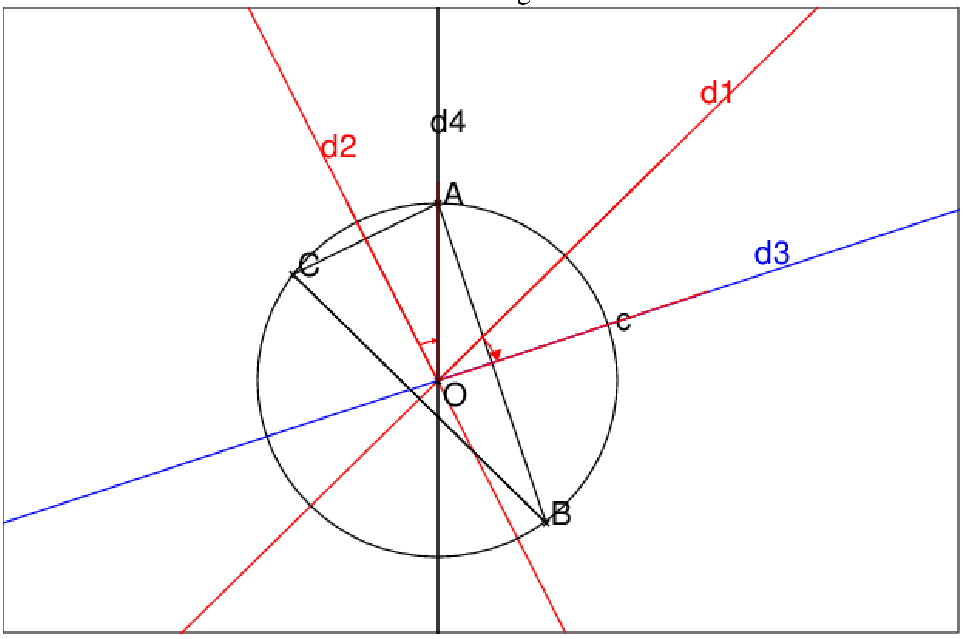
Pour obtenir cette construction, on tape :

```
d1:=droite(0,5+5*i,affichage=1);
d2:=droite(0,-2+4*i,affichage=1);
d3:=droite(0,6+2*i,affichage=1);
A:=point(4+4*i);
d4:=homothetie(A,2,d3):;d4;
d5:=homothetie(A,2,d2):;d5;
B:=inter_unique(d4,d2);
C:=inter_unique(d5,d3);
triangle(A,B,C);
s:=segment(B,C);
P:=inter_unique(s,d1);
```

On tape :
`P-(B+C)/2;`
On obtient : `point(0,0)`

3. Construction de ABC admettant d_1, d_2, d_3 comme médiatrices.
   On doit avoir :
   A, B, C sont, par exemple, sur le cercle c de centre O et de rayon 4 : qui sera le cercle circonscrit au triangle ABC.
   B doit être le symétrique de A par rapport à d_3, C doit être le symétrique de B par rapport à d_1, et A doit être le symétrique de C par rapport à d_2.
   Soit S_1 (resp S_2, S_3) la symétrie par rapport à d_1 (resp d_2, d_3). A est donc

un point fixe de $S_4 = S_2 \circ S_1 \circ S_3$. Soient d_4 la transformée de d_2 dans la rotation de centre O et d'angle (d_1, d_3) et S_4 est la symétrie par rapport à d_4.
A est donc l'intersection de c et de d_4, puis on trouve B comme symétrique de A par rapport à d_3 et C comme symétrique de A par rapport à d_2.
Dans un triangle les 3 médiatrices sont concourantes donc d_1 qui est la droite $A0$ est la troisème médiatrice du triangle ABC.



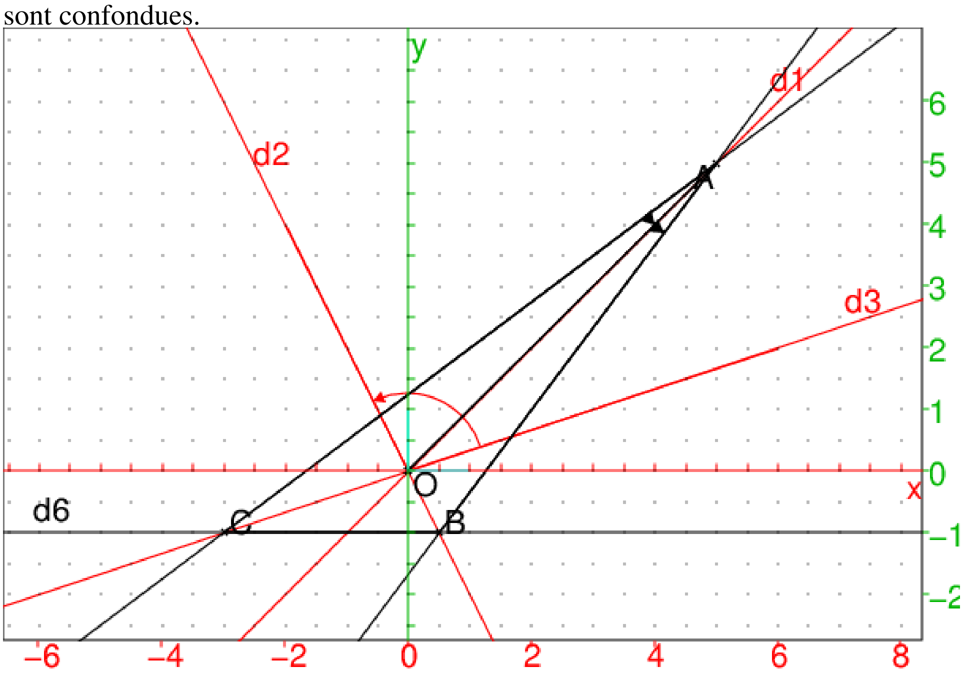
Pour obtenir cette construction, on tape :

```
d1:=droite(0,5+5*i,affichage=1);
d2:=droite(0,-2+4*i,affichage=1);
d3:=droite(0,6+2*i,affichage=1);
O:=point(0,affichage=quadrant4);
affichage(angle(d1,d3,""),1);
d4:=rotation(0,angle(d1,d3),d2):;d4;
affichage(angle(d2,d4,""),1);
c:=cercle(0,4)
A:=inter_unique(c,d4);
B:=symetrie(d3,A);
C:=symetrie(d2,A);
triangle(A,B,C);
s:=segment(B,C);
I:=inter_unique(s,d1);
```

On tape :
`simplify(equation(mediatrice(B,C)))`
On obtient : `y=x`

4. Construction de ABC admettant d_1, d_2, d_3 comme bissectrices.
   On doit avoir :
   A sur d_1, B sur d_2 et C sur d_3 et
   la somme des angles du triangle ABC égale à π donc :

$\dfrac{\widehat{A}}{2}+\dfrac{\widehat{B}}{2}+\dfrac{\widehat{C}}{2}=\dfrac{\pi}{2}$ donc
angle$(d_2,d_3)=\widehat{BOC}=\pi/2+\frac{\widehat{A}}{2}$.
Le problème est donc possible si l'angle (d_2,d_3) est obtu i.e. supérieur à $\pi/2$. On choisit un point A sur $d1$, puis on trace les droites d_4 et d_5 pour avoir :
angle(d_1,d_4)=angle(d_2,d_3)-$\pi/2$ et
angle(d_1,d_5)=-angle($\widehat{BOC}$)-$\pi/2$
Puis B est l'intersection de d_4 et de d_2.
On trace d_6 pour que d_2 soit la bissectrice de l'angle B.
Puis C est l'intersection de d_5 et de d_6.
Pourquoi C est-il sur d_3 ? Dans un triangle les 3 bissectrices sont concourantes donc CO est la troisème bissectrice du triangle ABC. donc l'angle $\widehat{COB}=\pi/2+angle(d1,d4)=angle(d_3,d_2)$ donc les droites OC et d_3 sont confondues.

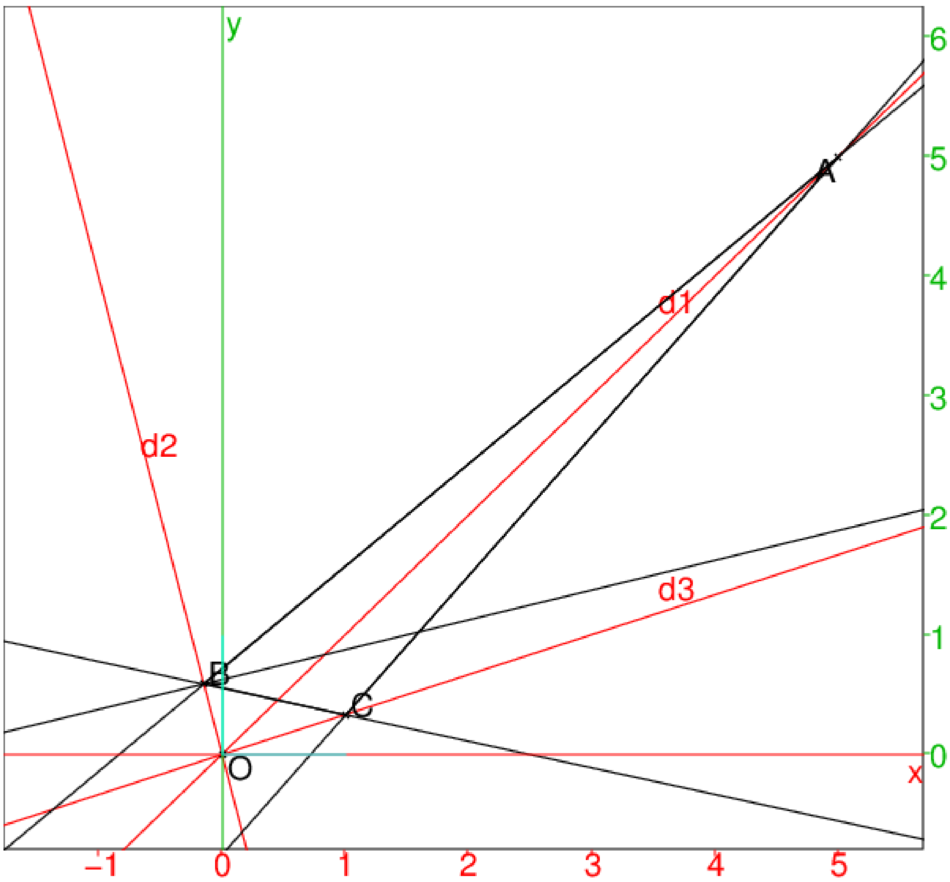


Pour obtenir cette construction, on tape :

```
d1:=droite(0,5+5*i,affichage=1);
d2:=droite(0,-2+4*i,affichage=1);
d3:=droite(0,6+2*i,affichage=1);
O:=point(0,affichage=quadrant4);
A:=point(5+5*i);
d4:=rotation(A,angle(d3,d2)-pi/2,d1):;
angle(d1,d4,"");
d5:=rotation(A,-angle(d3,d2)+pi/2,d1):;
angle(d5,d1,"");
B:=inter_unique(d4,d2);
d6:==rotation(B,angle(d4,d2),d2);
C:=inter_unique(d5,d6);
triangle(A,B,C);
est_element(C,d3)
```

On tape : `est_element(C,d3)`
On obtient : `1`
On tape : `angle(d5,d3),angle(d3,d1)`
On obtient : `1`
On tape : `angle(d1,d6)-angle(d1,d2)`
On obtient : `1`

5. Construction de ABC admettant d_1 comme bissectrice de l'angle A, et d_2 (resp d_3) comme bissectrice extérieure de l'angle B (resp C).
On doit avoir, A sur d_1, B sur d_2 et C sur d_3 et
la somme des angles du triangle ABC est égale à π donc :
$\dfrac{\widehat{A}}{2}+\dfrac{\widehat{B}}{2}+\dfrac{\widehat{C}}{2}=\dfrac{\pi}{2}$.
Soit $\beta=\pi/2$-angle($B/2$), $\gamma=\pi/2$-angle($C/2$) et
soit angle(O_1)=angle($\widehat{BOA}$), angle(O_2)=angle($\widehat{AOC}$).
Puisque la somme des angles du triangle ABO est égale à π on a :
$\pi-\beta$+angle($A/2$)+angle($O1$)=π donc
β=angle($A/2$)+angle(O_1)=$\pi/2$-angle($B/2$)=angle($A/2$)+angle($C/2$)
Donc angle(O_1)=angle($C/2$).
De même, la somme des angles du triangle ACO égale à π on a :
$\pi-\gamma$+angle($A/2$)+angle(O_2)=π donc
γ=angle($A/2$)+angle(O_2)=$\pi/2$-angle($C/2$)=angle($A/2$)+angle($B/2$)
donc angle(O_2)=angle($B/2$).
Comme angle($B/2$)+angle($C/2$)=$\pi/2$-angle($A/2$)<$\pi/2$, le problème n'est possible que si :
angle(O_1)+ angle(O_2)=angle($\widehat{BOC}$ est aigu i.e. est inférieur à $\pi/2$.
Ce qui n'est pas le cas ici ! Modifions l'équation de la droite d_2 et supposons que d_2 a pour équation $y=-4x$.
On a : angle($A/2$)=$\pi/2$-angle(O_1)-angle(0_2)=$\pi/2$-angle($\widehat{BOC}$)
Donc : $\frac{\widehat{A}}{2}=\pi/2$-angle($d_2,d_3$).
On choisit un point A sur d_1, puis on trace les droites d_4 et d_5 pour avoir :
angle(d_1,d_4)=-$\pi/2$+angle(d_3,d_2) et angle(d_1,d_5)=$\pi/2$-angle(d_3,d_2)
d_4 coupe d_2 en B .
On trace la droite d_6 pour que d_2 soit la bissectrice extérieure de l'angle B.
On a d_6 est la symétique de $d4$ par rapport à $d2$ et angle(d_2,d_6)=angle(d_4,d_2)
d_6 et d_5 se coupent en C.
Pourquoi C est-il sur d_3 ? Dans un triangle ABC, la bissectrice de l'angle A et les 2 bissectrices extérieures des angles B et C sont concourantes donc CO est la bissectrice extérieure de C.
donc l'angle $\widehat{BOC}=-\pi/2+$angle(d_1,d_5) =angle(d_2,d_3)
donc les droites OC et d_3 sont confondues.



Pour obtenir cette construction, on tape :

```
d1:=droite(0,5+5*i,affichage=1);
d2:=droite(0,-2+8*i,affichage=1);
d3:=droite(0,6+2*i,affichage=1);
O:=point(0,affichage=quadrant4);
A:=point(5+5*i);
d4:=rotation(A,angle(d3,d2)-pi/2,d1):;d4;
d5:=rotation(A,-angle(d3,d2)+pi/2,d1):;d5;
B:=inter_unique(d4,d2);
d6:=symetrie(d2,d4);
C:=inter_unique(d5,d6);
triangle(A,B,C);
```

On tape : `est_element(C,d3)`
On obtient : `1`

1.17 Cercles remarquables du triangle

Étant donné trois points `A`, `B` et `C` les commandes suivantes permettent de tracer les cercles remarquables du triangle ABC :

— Le cercle circonscrit
`circonscrit(A,B,C)` trace le cercle circonscrit au triangle ABC.

— Le cercle inscrit
`inscrit(A,B,C)` trace le cercle inscrit au triangle ABC.

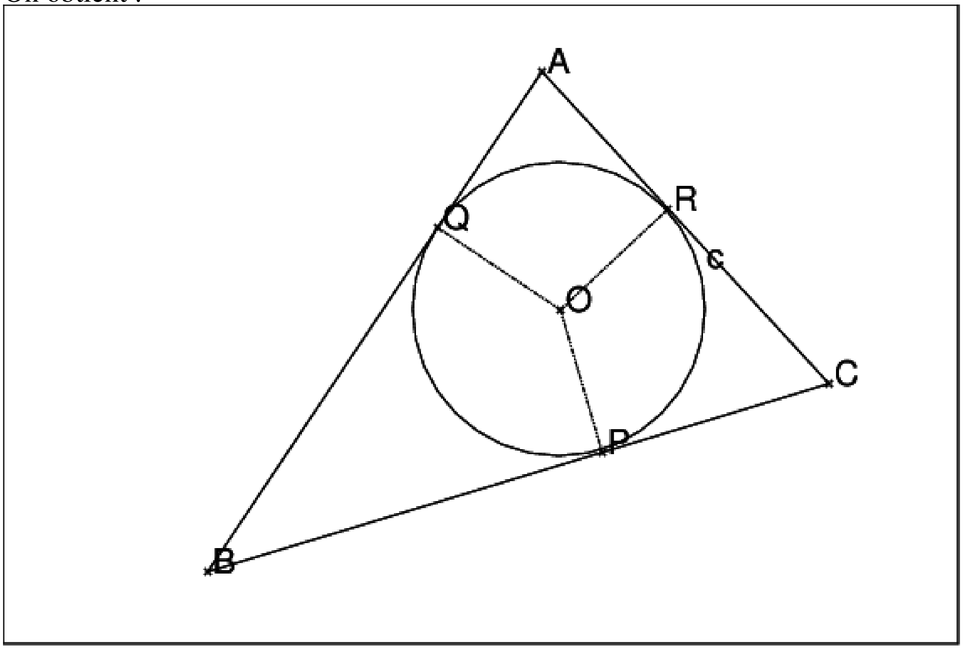
— Le cercle exinscrit
  `exinscrit(A,B,C)` trace le cercle exinscrit dans l'angle A du triangle ABC,
  `exinscrit(B,A,C)` trace le cercle exinscrit dans l'angle B du triangle ABC,
  `exinscrit(C,B,A)` trace le cercle exinscrit dans l'angle C du triangle ABC.

Exercice1
Soit un triangle ABC et c sont cercle inscrit.
Le centre O de c se projette en P (resp Q, R) sur BC (resp AB, AC).
Montrer que :
$2PB = AB + BC - AC = 2QB$,
$2QA = AB + AC - BC = 2AR$,
$2CP = BC + AC - AB = 2CR$

Solution
On fait la figure : dans un niveau de geométrie 2d, en mode point on clique sur 3 points A, B, C et on tape :

```
triangle(A,B,C);
c:=inscrit(A,B,C);
O:=centre(c);
P:=projection(droite(B,C),O);
Q:=projection(droite(B,A),O);
R:=projection(droite(A,C),O);
segment(O,P,affichage=ligne_tiret_pointpoint);
segment(O,Q,affichage=ligne_tiret_pointpoint);
segment(O,R,affichage=ligne_tiret_pointpoint);
```

On obtient :

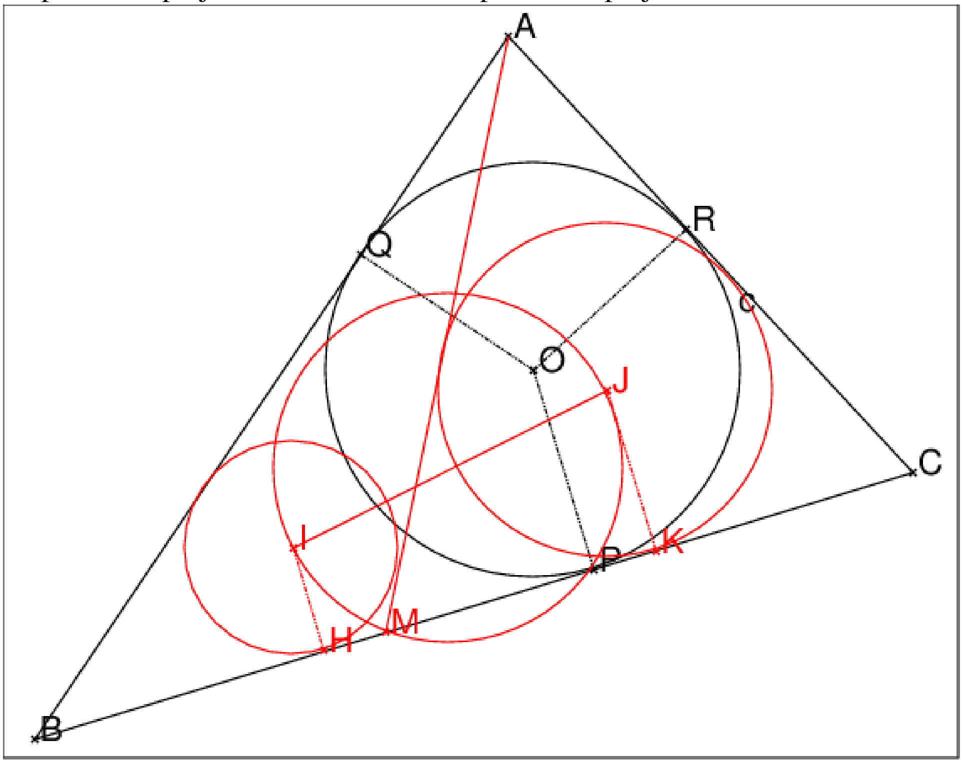


On a les égalités :
$BP = BQ$, $AQ = AR$, $CP = CR$ et

$AB = AQ + QB$, $AC = AR + RC$, $BC = BP + PC$ donc
$AB + BC = AQ + QB + BP + PC = 2BP + AQ + PC =$
$2BP + AR + CR = 2BP + AC$ donc
$2BP = AB + BC - AC = 2BQ$
de même on a :
$2QA = AB + AC - BC = 2AR$ et $2CP = BC + AC - AB = 2CR$

Prolongement de l'exercice1

Soit M un point du segment BC.
Soient I le centre du cercle inscrit du triangle ABM et J le centre du cercle inscrit du triangle AMC.
Le point I se projette en H sur BC et le point J se projette en K sur BC.



1. Montrer que les segments MP et HK ont même milieu.
2. En déduire que, lorsque M parcourt BC, le cercle de diamètre IJ passe par le point fixe P (projection sur BC du centre O du cercle inscrit du triangle ABC.
3. Faire un pliage qui met en évidence que $PB - PC = AB - AC$.

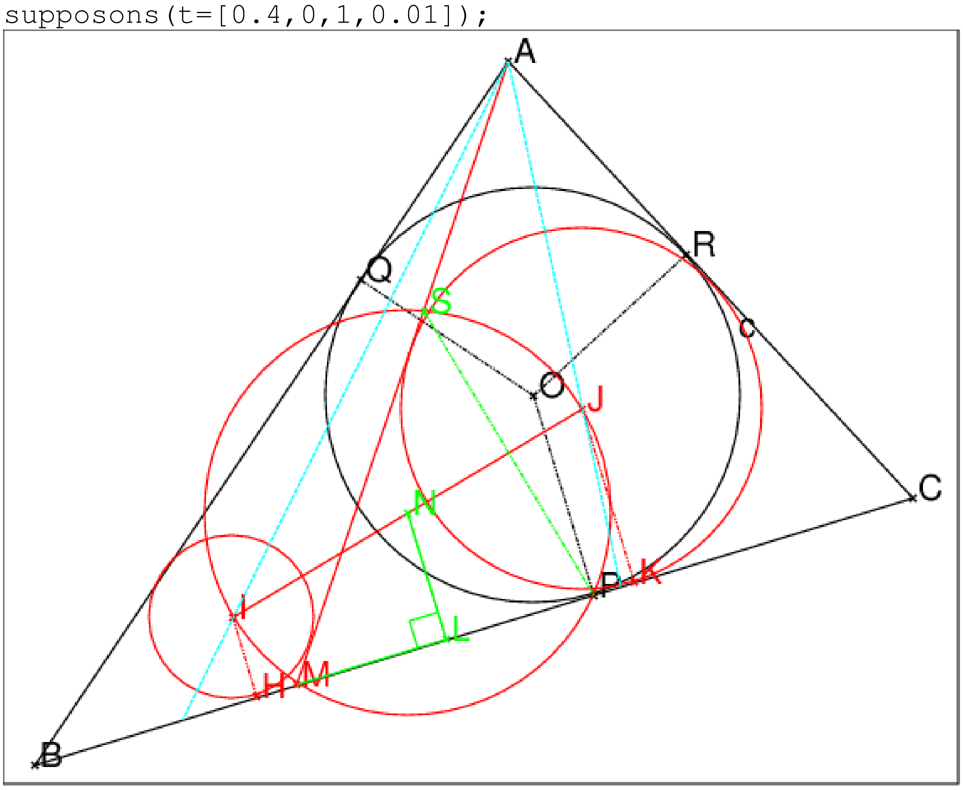
Remarque Sur la figure ci-dessus, lorsque M varie (à l'aide du paramètre t), les parties mobiles sont en rouge.
Pour faire cette figure, on a complété la figure de l'exercice1 en tapant :

```
supposons(t=[0.4,0,1,0.01]);
M:=element(segment(B,C),t,affichage=1);
segment(A,M,affichage=1)
c1:=inscrit(A,B,M,affichage=1):;
I:=affichage(centre(c1),1);c1;
c2:=inscrit(A,C,M,affichage=1):;
```

```
J:=affichage(centre(c2),1);c2;
H:=projection(segment(B,C),I,affichage=1);
K:=projection(segment(B,C),J,affichage=1);
segment(I,H,affichage=ligne_tiret_pointpoint+1);
segment(J,K,affichage=ligne_tiret_pointpoint+1);
segment(I,J,affichage=1);
cercle(I,J,affichage=1);
```

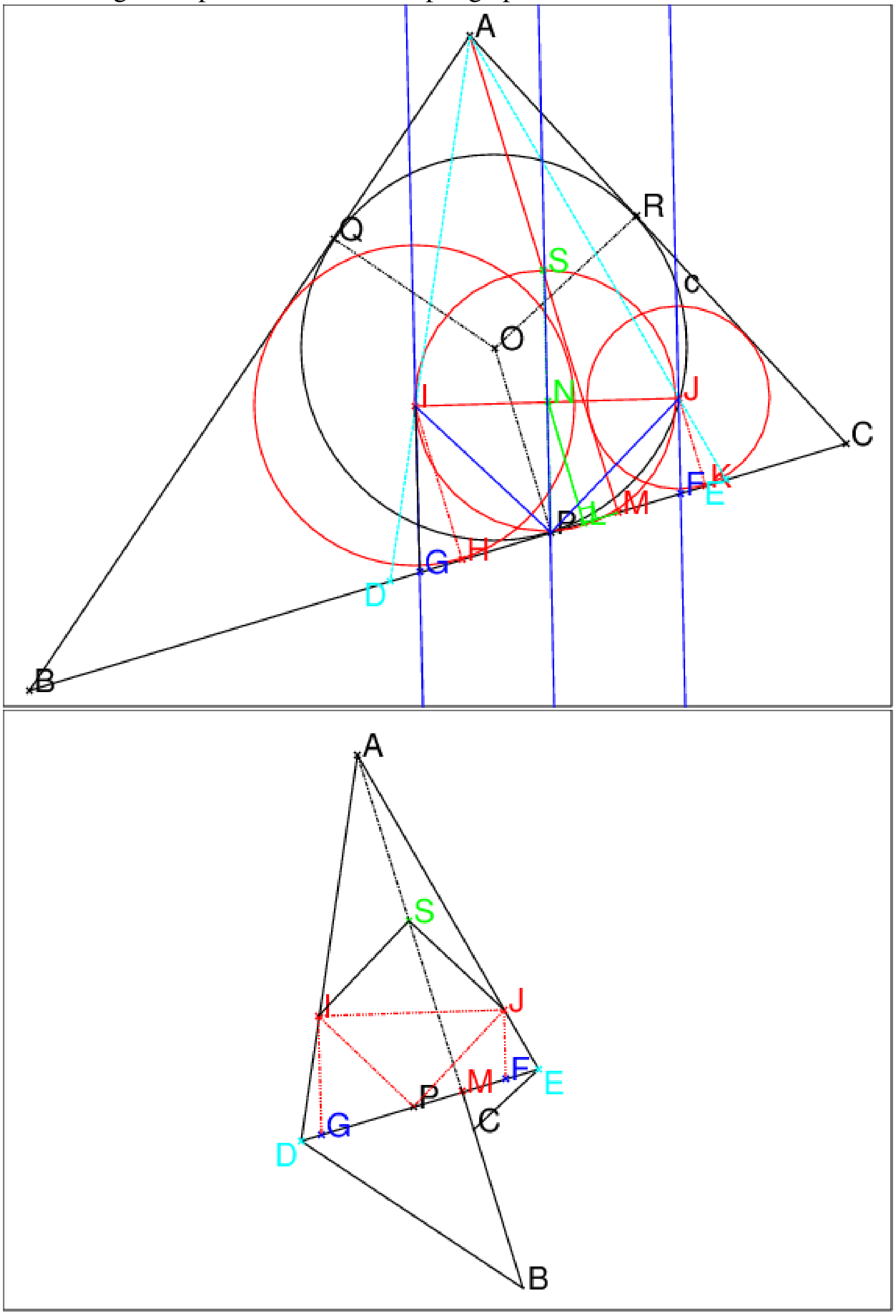
Solution

1. Montrons que les segments MP et HK ont même milieu.
   Pour cela montrons que :$(BP + BM)/2 = (BH + BK)/2$.
   On a d'après l'exercice1 :
   $2BP = BC + AB - AC$ on a :
   $2BH = BM + AB - AM$
   $2BK = 2BM + 2MK = 2BM + MC + AM - AC$
   donc :
   $2(BH + BK) = 2BM + (BM + MC) + AB - AC =$
   $2BM + BC + AB - AC = 2BM + 2BP = 2(BM + BP)$
   Donc HK et MP ont même milieu.
2. Avec `Xcas` on peut faire bouger le point M en faisant varier le paramètre t entre 0 et 1 par pas de 0.01 grâce à l'instruction :
   `supposons(t=[0.4,0,1,0.01]);`

   Le cercle de diamètre IJ passe par M puisque l'angle $\widehat{IMJ} = \pi/2$ (IM (resp MJ) est la bissectrice intérieure (resp extérieure) de l'angle $\widehat{IMJ}$).
   Le cercle de diamètre IJ passe aussi par P car : le milieu N de IJ se projette sur le milieu L de HK qui est aussi le milieu de PM, donc $NM = NP$ (égalité des triangles rectangles NML et NKL).

3. Le symétrique S de P par rapport à IJ se trouve aussi sur le cercle de diamètre IJ. Pour faire le pliage, on découpe un triangle ABC sur lequel on marque les points P, M, I, J, S et le point N milieu de IJ et on trace les droites parallèles àNP passant par I (resp J) qui coupe BC en G (resp F). Voici la figure dépliée et en cours de pliage pour $t = 0.72$:



On plie vers l'arrière selon AI et AJ et on marque les plis IP, JP et IJ et on plie IJP autour de IJ pour amener P sur S en aplatissant les plis ID et JE et en formant les plis JF et IG (on a $IG + JF = 2NP = IJ$, $\widehat{GIP} = \widehat{NPI} = \widehat{PIN}$ et $\widehat{FJP} = \widehat{JPN} = \widehat{PJN}$). Les plis JF, IG, IP, JP, amènent F et G en un point de IJ et P en S et on voit que $PC - PB = AC - AB$.

1.18 Les fonctions booléennes

— `est_element` pour savoir si un point appartient à un objet géométrique.
  `est_element` est une fonction booléenne ayant comme argument un point et un objet géométrique.
  `est_element` vaut 1 si le point appartient à l'objet géométrique, et vaut 0 sinon.
— `est_aligne` pour savoir si 3 points sont alignés.
  `est_aligne` est une fonction booléenne ayant comme argument trois points.
  `est_aligne` vaut 1 si les trois points sont alignés, et vaut 0 sinon.
— `est_cocyclique` pour savoir si 4 points sont cocycliques.
  `est_cocyclique` est une fonction booléenne ayant comme argument quatre points.
  `est_cocyclique` vaut 1 si les quatre points sont cocycliques, et 0 sinon.
— `est_parallele` pour savoir si 2 droites sont parallèles.
  `est_parallele` est une fonction booléenne ayant comme argument deux droites.
  `est_parallele` vaut 1 si les deux droites sont parallèles, et, vaut 0 sinon.
— `est_perpendiculaire` pour savoir si 2 droites sont perpendiculaires.
  `est_perpendiculaire` est une fonction booléenne ayant comme argument deux droites.
  `est_perpendiculaire` vaut 1 si les deux droites sont perpendiculaires, et, vaut 0 sinon.
— `est_orthogonal` pour savoir si 2 droites ou 2 cercles sont orthogonaux.
  `est_orthogonal` est une fonction booléenne ayant comme argument deux droites (resp deux cercles).
  `est_orthogonal` vaut 1 si les deux droites sont perpendiculaires (resp les deux cercles sont orthogonaux), et vaut 0 sinon.

1.19 La géométrie du cercle

1.19.1 Arc capable

`arc` permet de tracer un arc défini par deux points et la mesure de son angle au centre α ($-2\pi \leq \alpha \leq 2\pi$) le signe de α donne le sens de parcours.
Pour avoir l'arc capable `AB` de mesure u il faut taper :
`arc(A,B,2*(-pi+u))` si $\pi > u > 0$ ou
`arc(A,B,2*(pi+u))` si $-\pi < u < 0$.
Activité, sans se servir de la commade `arc`
Soient deux points A et B. Le lieu des points M d'où l'on voit un segment AB sous un angle u donné ($u \neq \pi+2*k*\pi$ et $u \neq 2*k*\pi$) est un arc de cercle ou deux arcs de cercle selon que ; $u = mesure(\overrightarrow{MA}, \overrightarrow{MB})$ ou que $u = mesure(\widehat{AMB})$.
On tape :

```
Arcaporient(A,B,u):={
AB:=segment(A,B);
```

```
O1:=inter(mediatrice(A,B),rotation(A,pi/2-u,droite(A,B)))[0];
if (u>0) return(cercle(O1,A-O1,arg(B-O1),arg(B-O1)+2*(pi-u)));
return(cercle(O1,A-O1,arg(B-O1),arg(B-O1)+2*(-pi-u)));
};
Arcap(A,B,u):={
local L;
AB:=segment(A,B);
O1:=inter(mediatrice(A,B),rotation(A,pi/2-u,droite(A,B)))[0];
O2:=inter(mediatrice(A,B),rotation(A,-pi/2+u,droite(A,B)))[0];
L:=[cercle(O1,A-O1,arg(B-O1),arg(B-O1)+2*(pi-u))];
L:=append(L,cercle(O2,A-O2,arg(A-O2),arg(A-O2)+2*(pi-u)));
return(L);
};
```

Exercice : La droite de Steiner
Soient un triangle quelconque ABC, H son orthocentre et c son cercle circonscrit.

— Soit Ha le symétrique de H par rapport à BC.
Montrer que $Ha \in c$.

— Soit I un point de c et Ia (resp Ib, Ic) le symétrique de I par rapport à (BC) (resp (AC), (AB)). Montrer que Ia, Ib, Ic et H sont alignés selon une droite appelée `la droite de Steiner de` I
— On montre que l'angle $\widehat{BHC} = \widehat{BHaC} = \pi - \widehat{BAC}$ En effet soient M et N les pieds des haureurs issues de B et C : le quadrilatère $AMHN$ est inscriptible donc : $\widehat{BHC} = \pi - \widehat{BAC}$
Par symétrie on a $\widehat{BHC} = \widehat{BHaC}$ Donc $\widehat{BHaC} = \pi - \widehat{BAC}$ donc le quadrilatère $AMHaN$ est inscriptible.
— On considère les cercles ca (resp cb, cc) les cercles symétriques de c par rapport à BC (resp AC, AB) et Hb (resp Hc les symétriques de H par rapport à AC (resp AB). Par des considérations d'angle on montre que l'angle $\widehat{IbHIc} = \pi$ (resp $\widehat{IbHIa} = \pi$ donc que H, Ia, Ib, Ic sont alignés

1.19.2 Puissance d'un point par rapport à un cercle

Si un point A est à une distance d du centre d'un cercle C de rayon r, la puissance de A par rapport au cercle C est égale à $d^2 - r^2$. On tape :
`puissance(cercle(0,1+i),3+i)`
On obtient :
`8`
En effet : $r = \sqrt{2}$ et $d = \sqrt{10}$ donc $d^2 - r^2 = 8$ On peut aussi faire écrire la fonction `puissancc` comme exercice de programmation.

```
Puissancc(A,C):={
local O,r,d2;
O:=centre(C);
r:=rayon(C);
d2:=longueur2(A,O);
return(d2-r^2);
```

```
};
```

1.19.3 Axe radical de deux cercles

L'axe radical de deux cercles C_1 et C_2 est le lieu des points qui ont même puissance par rapport à C_1 et à C_2 On tape :
`axe_radical(cercle(0,1+i),cercle(1,1+i)))`
On obtient :
`Le tracé de la droite x=1/2`
En effet : la droite x=1/2 est la médiatrice du segment [0 ;1] On peut aussi faire écrire la fonction `axeradical` comme exercice de programmation.

```
Axe_radical(C1,C2):={
local O1,O2,r1,r2,H,t;
O1:=centre(C1);
O2:=centre(C2);
r1:=rayon(C1);
r2:=rayon(C2);
t:=-(r2^2-r1^2)/2/longueur2(O2,O1)+1/2;
H:=O1+(O2-O1)*t;
return(perpendiculaire(H,droite(O1,O2)));
};
```

1.20 La division harmonique

Quatre points $A,\ B,\ C,\ D$ sont en division harmonique si on a :

$$\frac{\overline{CA}}{\overline{CB}} = -\frac{\overline{DA}}{\overline{DB}} = k$$

On dit aussi que C et D divisent le segment AB dans le rapport k et que le point D est le conjugué harmonique de C par rapport à A et B.
On écrit les fonctions `conj_harmonic1` qui étant donné les points A, B et C sur la droite AB détermine le point D conjugué harmonique de C par rapport à A et B. On suppose que C sur la droite AB et on écrit :

```
conj_harmonic1(A,B,C):={
local D;
D:=A+(B-A)*(C-A)/((C-B)+(C-A));
return(D);
};
```

On écrit la fonction `conj_harmonic2` qui définit les points C et D qui divise le segment AB dans le rapport k :

```
conj_harmonic2(A,B,k):={
local C,D;
C:=A+k/(1-k)*(A-B);
D:=A-k/(1+k)*(A-B);
```

```
return([C,D]);
};
```

Remarque Si C est le point qui divise le segment AB dans le rapport k, on a :
$\frac{\overline{CA}}{\overline{CB}} = k$ donc $\overline{CA} - k\overline{CB} = 0$ donc C est le barycentre de A affecté du coefficient 1 et B du coefficient $-k$. De même, si D est le point qui divise le segment AB dans le rapport $-k$, D est le barycentre de A affecté du coefficient 1 et B du coefficient k (si $1 + k \neq 0$ et si $1 - k \neq 0$).
On tape par exemple :

```
k:=2
C:=barycentre([[A,1],[B,-k]]))
D:=barycentre([[A,1],[B,k]])
```

On remarquera aussi que `C:=element(droite(A,B),t);` signifie :
$\overline{AC} = t * \overline{AB}$ donc :
`C-A:=t*(B-A)` soit `C:=(1-t)*A+t*B`.
On a donc :
$\frac{\overline{CA}}{\overline{CB}} = \frac{t}{t-1} = k$
On écrit la fonction `conj_harmonic3` en se servant de la commande `element` :

```
conj_harmonic3(A,B,k):= {
local C,D,t;
  t:=k/(k-1);
  C:=element(droite(A,B),t);
  D:=A+t/(2*t-1)*(B-A);
  return([point(C),point(D)]);
}
```

Activité
Soient trois points A, B, M alignés. Construire géométriquement le point N conjugué harmonique de M par rapport à A et B.

Réponse
On définit 3 points A, B, C non alignés.
On trace la droite MC et la parallèle $d1$ (resp $d2$) à cette droite passant par A (resp B).
La droite BC coupe $d1$ en K et la droite AC coupe $d2$ en L
N est alors l'intersection de la droite KL avec la droite AB.
En effet :

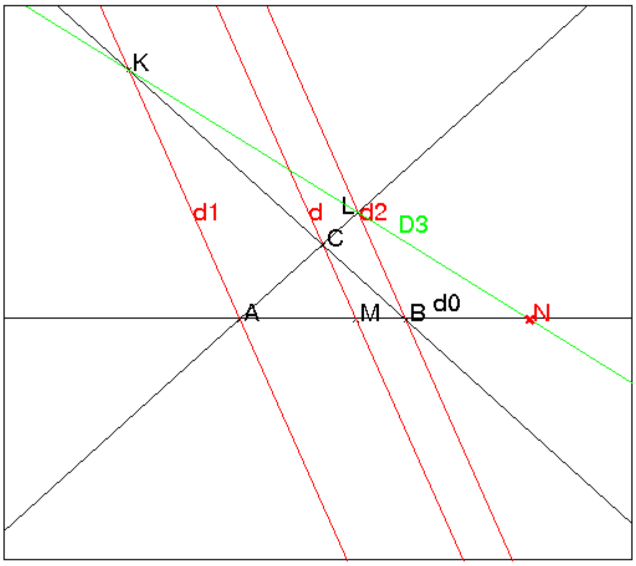
— $\frac{\overline{MA}}{\overline{MB}} = \frac{\overline{CB}}{\overline{CK}}$ car les parallèles d, d_1, d_2 coupent les droites AB et KB en M et C.
— $\frac{\overline{CB}}{\overline{CK}} = \frac{\overline{CL}}{\overline{CA}} = \frac{\overline{BL}}{\overline{KA}}$ car les triangles CBL et CKA sont homothétiques.
— $\frac{\overline{BL}}{\overline{AK}} = \frac{\overline{BN}}{\overline{AN}} = \frac{\overline{LN}}{\overline{KN}}$ car les triangles BLN et AKN sont homothétiques.

donc puisque $\frac{\overline{BL}}{\overline{KA}} = -\frac{\overline{BL}}{\overline{AK}}$ on a :
$\frac{\overline{MA}}{\overline{MB}} = -\frac{\overline{BN}}{\overline{AN}}$.
On tape :

```
A:=point(-2,-2.8);
B:=point(2,-2.8);
```

```
C:=point(0,-1);
d0:=droite(A,B):;d0;
t:=element(0..1,0.3);
M:=element(d0,t);
d:=droite(M,C,affichage=1);
d1:=parallele(A,d,affichage=1);
d2:=parallele(B,d,affichage=1);
D1:=droite(A,C):;
D2:=droite(B,C):;
K:=inter(d1,D2)[0];
L:=inter(d2,D1)[0]:;
legende(L,"L",quadrant2);
D3:=droite(K,L,affichage=2);
N:=affichage(inter(d0,D3)[0],1+epaisseur_point_2);
```

On obtient :



Autre réponse

On définit 3 points A, B, C non alignés.

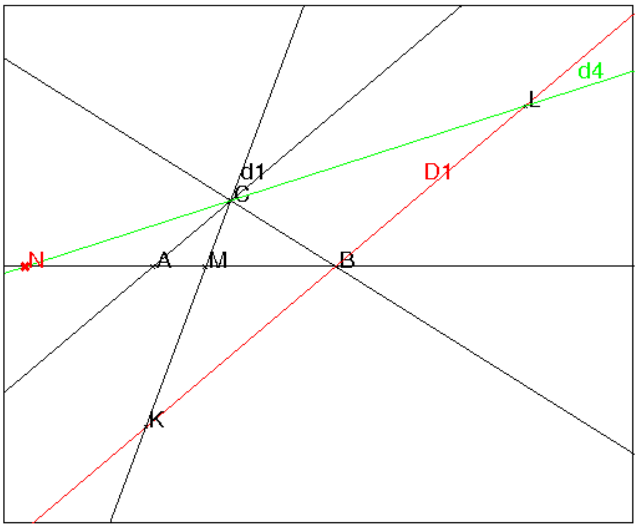
On définit les droites $d1 = CA$, $d2 = CB$, $d3 = CM$. On peut alors tracer la droite $d4$ pour que le faisceau $d1$, $d2$, $d3$, $d4$ soit harmonique.

On trace $D1$ la parallèle à la droite $d1$ passant par B qui coupe CM en K. On définit L sur $D1$ tel que $BK = BL$ et $L \neq K$.

On trace CL qui coupe AB en N.

```
A:=point(-2,-2);
B:=point(3,-2);
C:=point(0.13,-1);
d0:=droite(A,B):;d0;
```

```
t:=element(0..1,0.3);
M:=element(d0,t);
d3:=droite(C,M):;d3;
d1:=droite(C,A);
d2:=droite(C,B):;d2;
D1:=parallele(B,d1,affichage=1);
K:=inter(d3,D1)[0];
L:=inter(cercle(B,K-B),D1)[1];
d4:=droite(C,L,affichage=2);
N:=affichage(inter(d0,d4)[0],1+epaisseur_point_2);
```



1.21 Polaires par rapport à deux droites

1.21.1 Définition

Le lieu des conjugués harmoniques d'un point A par rapport à deux droites $D1, D2$ est une droite d vérifiant :

— si $D1$ et $D2$ sont concourantes en O, la droite d passe par O et le faisceau $(OA, d, D1, D2)$ est harmonique,

— si $D1$ et $D2$ sont paralléles, la droite d est paralléle à $D1$ et à $D2$, le faisceau $(d', d, D1, D2)$ est harmonique où d' est la paralléle à $D1$ passant par A.

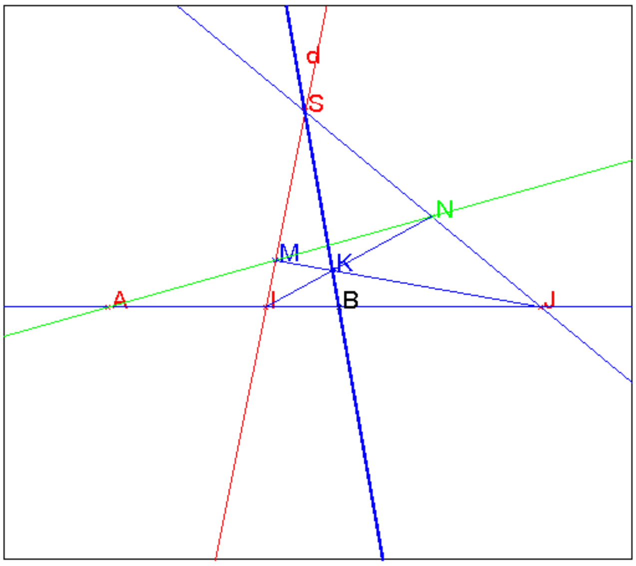
1.21.2 Un lieu

Soient un point S et trois points A, I, J alignés. Soit M un point variable de la droite SI. La droite AM coupe la droite SJ en N.
On cherche le lieu de K intersection de IN et de JM lorsque M se déplace sur la droite SI.

Les figures

On tape :

```
S:=point(1+5i,affichage=1);
A:=point(-4,affichage=1);
I:=point(0,affichage=1);
J:=point(7,affichage=1);
droite(A,J);
d:=droite(I,S,affichage=1);
droite(S,J,affichage=1);
t:=element((-2) .. 2,0.24,0.04)
M:=element(d,t);
N:=inter_droite(droite(A,M),droite(S,J),affichage=2);
droite(A,M,affichage=2);
segment(I,N);
segment(J,M);
K:=inter_droite(droite(I,N),droite(J,M),affichage=4);
L:=lieu(K,M):;
affichage(L,4+epaisseur_ligne_2);
B:=conj_harmonique(I,J,A);
droite(S,B);
```



On remarque que si B est le conjugué harmonique de A par rapport à I et J, le lieu L de K est la droite SB .

La démonstration

Le lieu de K est la polaire de A par rapport à CE et FG. En effet si CE et FG se coupent en S, si le point L est le conjugué harmonique de A par rapport à I et J et si le point P est le conjugué harmonique de A par rapport à M et N, la polaire de A par rapport à CE et FG passe par S, L et P, et la polaire de A par rapport à IN et JM passe par K, L et P donc, K se trouve sur la polaire de A par rapport à CE et FG.

1.22 Pôles et polaires par rapport à un cercle ou une conique

1.22.1 Définitions et propriétés

On dit que 2 points A et B sont conjugués par rapport à un cercle ou une conique (C) si ils forment une division harmonique avec les points E et F où la droite AB coupe (C).
Le lieu des conjugués harmoniques d'un point A par rapport à un cercle ou une conique C est une droite d appelée **polaire** de A par rapport à (C). Si la polaire d coupe (C) en T_1 et T_2 alors les droites AT_1 et AT_2 sont tangentes à (C).
Lorsque (C) est un cercle ou une conique propre à toute droite d on peut faire correspondre un point A qui admet d comme polaire par rapport à (C) : on dit que A est le **pôle** de d.
Si la conique est dégénérée en deux droites OU et OV chaque droite passant par O est la polaire d'une infinité de points alignés avec O (cf 1.21).

1.22.2 Comment trouver l'équation de la polaire

On va travailler en coordonnèes homogènes en dimension 2 :
un point $M(x, y)$ a pour coordonnées homogènes X, Y, T avec $x = X/T$ et $y = Y/T$.
Si l'équation de la conique est : $F(X, Y, T) = 0$ alors F est une forme quadratique de matrice Q et on a, si $M = [X, Y, T]$:

$$F(X, Y, T) = M * Q * M = 0$$

La polaire du point $A = [X_a, Y_a, T_a]$ a pour équation :

$$X * F'_X(X_a, Y_a, T_a) + Y * F'_Y(X_a, Y_a, T_a) + T * F'_T(X_a, Y_a, T_a) = 0$$

ou encore :

$$X_a * F'_X(X, Y, T) + Y_a * F'_Y(X, Y, T) + T_a * F'_T(X, Y, T) = 0$$

ou encore :

$$M * Q * A = 0$$

1.22.3 Comment trouver les coordonnées du pôle

Définitions
Une droite d d'équation $ux + vy + h = 0$ a pour coordonnées tangentielles u, v, h.
L'équation tangentielle d'une courbe (C) est la condition nécessaire et suffisante pour que la droite d de coordonnées tangentielles $[u, v, h]$ lui soit tangente.
Équation tangentielle de la conique (C)

L'équation tangentielle de la conique (C) d'équation $F(X,Y,T) = M{*}Q{*}M = 0$ est :

$$G(u,v,h) = [u,v,h] * inv(Q) * [u,v,h]$$

Équation tangentielle de la polaire L'équation tangentielle de la polaire de $A = [X_a, Y_a, T_a]$ par rapport à la conique (C) est : $A * Q$.
Coordonnées du pôle
Soit d une droite de coordonnées tangentielles u_d, v_d, h_d (donc d'équation $ux + vy + h = 0$), le pôle de d par rapport à (C) a pour coordonnées homogènes :

$$G'_u(u_d,v_d,h_d), G'_v(u_d,v_d,h_d)G'_h(u_d,v_d,h_d)$$

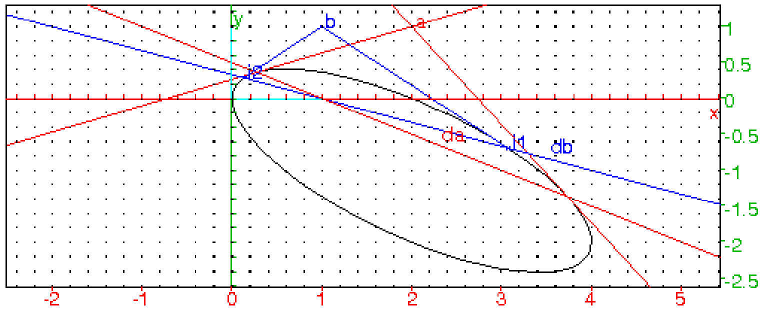
ou encore si $Q1 = det(Q) * inv(Q)$

$$[u_d, v_d, h_d] * Q1$$

1.22.4 Un exemple

Soient la conique C définie par $x^2 + 2x * y + 2 * y^2 - 2 * x = 0$, la droite d_b d'équation $x + 3y - 1 = 0$ et A le point $2 + i$.
Trouver la polaire d_a de A par rapport à C etle pôle B de d_b par rapport à C.
On tape :
```
q1:=x^2+2*x*y+2*y^2-2*x
q:=numer(subst(q1,[x,y]=[X/T,Y/T]))
```
On obtient :
```
X^2-2*X*T+2*X*Y+2*Y^2
```
On tape :
```
Q:=q2a(q,[X,Y,T])
```
On obtient :
```
[[1,1,-1],[1,2,0],[-1,0,0]]
```
On tape :
```
Q1:=det(Q)*inv(Q)
```
On obtient :
```
[[0,0,2],[0,-1,-1],[2,-1,1]]
```
On tape :
```
M:=[X,Y,T]
U:=[u,v,h]
A:=[2,1,1]
da:=A*Q*M
```
On obtient :
```
2*X+4*Y-2*T
```
La polaire d_a de A par rapport à C a donc pour équation : $x + 2y - 1 = 0$.
On tape :
```
A*Q
```
On obtient :
```
[2,4,-2]
```
La polaire d_a de A par rapport à C a donc pour équation tangentielle [2,4,-2] c'est à dire pour équation $x + 2y - 1 = 0$.
On tape :
```
db:=[1,3,-1]
```

```
B:=db*Q1
```
On obtient :
```
[-2,-2,-2]
```
Le pôle B de d_b par rapport à C a donc pour coordonnées : $(1, 1)$.
Pour avoir l'intersection i_1 et i_2 de d_b avec C, on tape :
```
solve(normal(subst(x^2+2*x*y+2*y^2-2*x,x=1-3y))=0,y)
```
On obtient :
```
[1/5*(-1-sqrt(6)),1/5*(-1+sqrt(6))]
```
Les points d'intersection i_1 et i_2 sont donc les points d'affixe z_1 et z_2 avec :
```
z1:=1+1/5*(-1-sqrt(6))*(-3+i), z2:=1+1/5*(-1+sqrt(6))*(-3+i)
```
Pour avoir l'équation tangentielle de la conique C, on tape :
```
U:=[u,v,h]
G:=normal(U*Q1*U)
```
On obtient :
```
h^2+4*h*u-2*h*v-v^2
```
On tape pour avoir l'équation tangentielle des tangentes à C passant par A (on choisit `h=1`) :
```
normal(solve(subst([G=0,U*A=0],h=1),[u,v]))
```
On obtient :
```
[[(-sqrt(3)+1)/2,sqrt(3)-2],[(sqrt(3)+1)/2,-sqrt(3)-2]]
```
Les tangentes à C passant par A ont donc pour équation :
$(-\sqrt{3}+1)x/2 + (\sqrt{3}-2)y + 1 = 0$ et $(\sqrt{3}+1)x/2 - (\sqrt{3}+2)y + 1 = 0$
On tape :
```
plotimplicit(q1,[x,y]);a:=point(2+i);b:=point(1+i);
t1:=droite((1-sqrt(3))/2*x+y*(-2+sqrt(3))+1=0,affichage=1);
t2:=droite((1+sqrt(3))/2*x-y*(2+sqrt(3))+1=0,affichage=1);
db:=droite(x+3y-1=0);da:=droite(x+2y-1=0);
segment(z1,b,affichage=4);segment(z2,b,affichage=4);
si1:=point(z1,affichage=4);i2:=point(z2,affichage=4);
```
On obtient :



1.23 Pôles et polaires par rapport à une sphère ou une quadrique

1.23.1 Définitions et propriétés

On dit que 2 points A et B sont conjugués par rapport à une sphère ou une quadrique (S) si ils forment une division harmonique avec les points E et F où la droite AB coupe (S).
Le lieu des conjugués harmoniques d'un point A par rapport à une sphère ou une quadrique (S) est un plan P appelée **plan polaire** de A par rapport à (S). Si le plan polaire P coupe (S) en la courbe T alors les droites joignant A à un point de T forment un cône tangent à (S).
Lorsque (S) est une sphère ou une quadrique propre, à tout plan P on peut faire correspondre un point A qui admet P comme polaire par rapport à (S) : on dit que A est le **pôle** de P.

1.23.2 Comment trouver l'équation de la polaire

On va travailler en coordonnèes homogènes :
en dimension 3 : X, Y, Z, T avec $x = X/T$, $y = Y/T$ et $z = Z/T$.
Si l'équation de la quadrique est : $F(X, Y, Z, T) = 0$ alors F est une forme quadratique de matrice Q et on a, si $M = [X, Y, Z, T]$:

$$F(X, Y, Z, T) = M * Q * M = 0$$

Le plan polaire du point $A = [X_a, Y_a, Z_a, T_a]$ a pour équation :

$$XF'_X(X_a, Y_a, Z_a, T_a)+YF'_Y(X_a, Y_a, Z_a, T_a)+ZF'_Z(X_a, Y_a, Z_a, T_a)+TF'_T(X_a, Y_a, Z_a, T_a) = 0$$

ou encore :

$$X_aF'_X(X, Y, Z, T)+Y_aF'_Y(X, Y, Z, T)+Z_aF'_Z(X, Y, Z, T)+T_aF'_T(X, Y, Z, T) = 0$$

ou encore :

$$M * Q * A = 0$$

1.23.3 Comment trouver les coordonnées du pôle

Définitions
Un plan P d'équation $ux + vy + wz + h = 0$ a pour coordonnées tangentielles u, v, w, h.
L'équation tangentielle d'une surface (S) est la condition nécessaire et suffisante pour que le plan P de coordonnées tangentielles $[u, v, w, h]$ lui soit tangente.
Équation tangentielle de la quadrique (S)
L'équation tangentielle de la conique (S) d'équation $F(X, Y, Z, T) = M * Q * M = 0$ est :

$$G(u, v, w, h) = [u, v, w, h] * inv(Q) * [u, v, w, h]$$

Équation tangentielle du plan polaire L'équation tangentielle du plan polaire de $A = [X_a, Y_a, Z_a, T_a]$ par rapport à la conique (C) est : $A * Q$.

Coordonnées du pôle
Soit d une droite de coordonnées tangentielles u_d, v_d, w_d, h_d (donc d'équation $ux+vy+wz+h=0$), le pôle de d par rapport à (C) a pour coordonnées homogènes :

$$G'_u(u_d, v_d, h_d), G'_v(u_d, v_d, h_d), G'_h(u_d, v_d, h_d)$$

ou encore si $Q1 = det(Q) * inv(Q)$

$$[u_d, v_d, w_d h_d] * Q1$$

1.23.4 Un exemple

Soient la quadrique S définie par $x^2+2x*y+2*y^2-2*x-2y+z^2=0$, A le point $[2,-1,1]$, le plan P_b d'équation $x+2y+z-2=0$ et le plan P_c d'équation $x+3y+z-1=0$.
Trouver la polaire d_a de A par rapport à S, le pôle B de d_b par rapport à S et le pôle C de d_c par rapport à S.
On tape :
```
q1:=x^2+2x*y+2*y^2-2*x-2y+z^2
q:=numer(subst(q1,[x,y,z]=[X/T,Y/T,Z/T]))
```
On obtient :
```
X^2-2*X*T+2*X*Y-2*Y*T+2*Y^2+Z^2
```
On tape :
```
Q:=q2a(q,[X,Y,Z,T])
```
On obtient :
```
[[1,1,0,-1],[1,2,0,-1],[0,0,1,0],[-1,-1,0,0]]
```
On tape :
```
Q1:=det(Q)*inv(Q)
```
On obtient :
```
[[-1,1,0,1],[1,-1,0,0],[0,0,-1,0],[1,0,0,1]]]
```
On tape :
```
M:=[X,Y,T]
U:=[u,v,h]
A:=[2,-1,1,1]
Pa:=A*Q*M
```
On obtient :
```
-Y+Z-T
```
Le plan polaire d_a de A par rapport à S a donc pour équation : $-y+z-1=0$
On tape :
```
A*Q
```
On obtient :
```
[0,-1,1,-1]
```
Le plan polaire P_a de A par rapport à S a pour équation tangentielle [0,-1,1,-1] c'est à dire pour équation $-y+z-1=0$
On tape :
```
Pb:=[1,2,1,-2]
B:=Pb*Q1
```
On obtient :
```
[-1,-1,-1,-1]
```

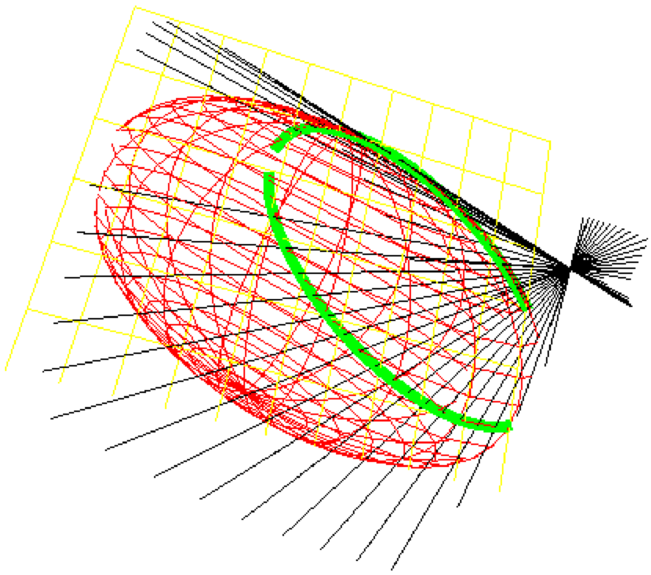
Le pôle B de d_b par rapport à S est donc le point coordonnées [1,1,1].
On tape :
```
Pc:=[1,3,1,-1]
C:=Pb*Q1
```
On obtient :
```
[1,-2,-1,0]
```
Le pôle C de d_c par rapport à S se trouve à l'infini dans la direction [1,-2,-1] car le plan P_c passe par le centre de l'ellipsoide S On tape :
```
normal(solve([Pa=0,qq=0,T=1],[X,Y,Z,T]))
```
On obtient :
```
[[X,(-X+sqrt(-2*X^2+6*X-3))/3,(-X+sqrt(-2*X^2+6*X-3)+3)/3,1],
[X,(-X-sqrt(-2*X^2+6*X-3))/3,(-X-sqrt(-2*X^2+6*X-3)+3)/3,1]]
```
On tape :
```
CC:=plotparam([X,(-X+sqrt(-2*X^2+6*X-3))/3,
(-X+sqrt(-2*X^2+6*X-3)+3)/3],X)
CC2:=plotparam([X,(-X-sqrt(-2*X^2+6*X-3))/3,
(-X-sqrt(-2*X^2+6*X-3)+3)/3],X)
```
Pour avoir le cône des tangentes de sommet `a`, on tape :

```
conetan():={
local L,M,N,X;
L:=NULL;
a:=point([2,-1,1]);
pour X de 0.63 jusque 2.37 pas 0.1 faire
M:=point(X,(-X+sqrt(-2*X^2+6*X-3))/3,
   (-X+sqrt(-2*X^2+6*X-3)+3)/3);
N:=point(X,(-X-sqrt(-2*X^2+6*X-3))/3,
   (-X-sqrt(-2*X^2+6*X-3)+3)/3);
L:=L,droite(a,M),droite(a,N);
fpour;
retourne L;
}:;
```

Dans un écran de géométrie 3d on tape :

```
plotimplicit(q1,[x,y,z],affichage=rouge);
plan(-y+z-1=0,affichage=jaune);
affichage([CC,CC2],vert+epaisseur_ligne_7);
a:=point([2,-1,1]);
conetan();
```

On obtient :



1.24 Tracé des coniques avec `Xcas`

— L'ellipse
`ellipse(A,B,C)` trace l'ellipse de foyers `A` et `B` passant par `C`.
`ellipse(A,B,a)` où `a` est un réel, trace l'ellipse de foyers `A` et `B` ayant `a` comme demi-grand axe.

— L'hyperbole
`hyperbole(A,B,C)` trace l'hyperbole de foyers `A` et `B` passant par `C`.
`hyperbole(A,B,a)` où `a` est un réel, trace l'hyperbole de foyers `A` et `B` ayant `a` comme demi-grand axe.

— La parabole
`parabole(A,B)` trace la parabole de foyer `A` et de sommet `B`.

Activité
Tracer un cercle tangent à une droite et passant par deux points donnés.
Pour cela, créer quatre points `A`, `B`, `C` et `D`.
Tracer la droite `E` passant par `A` et `B`.
Construire un cercle `C1` passant par `C` et `D` tangent à la droite `E`.

Réponse
On sait que le centre du cercle cherché est sur la médiatrice de CD et sur la parabole P de foyer C et de directrice la doite AB.
On clique avec la souris pour avoir quatre points `A`, `B`, `C` et `D` puis on exécute la

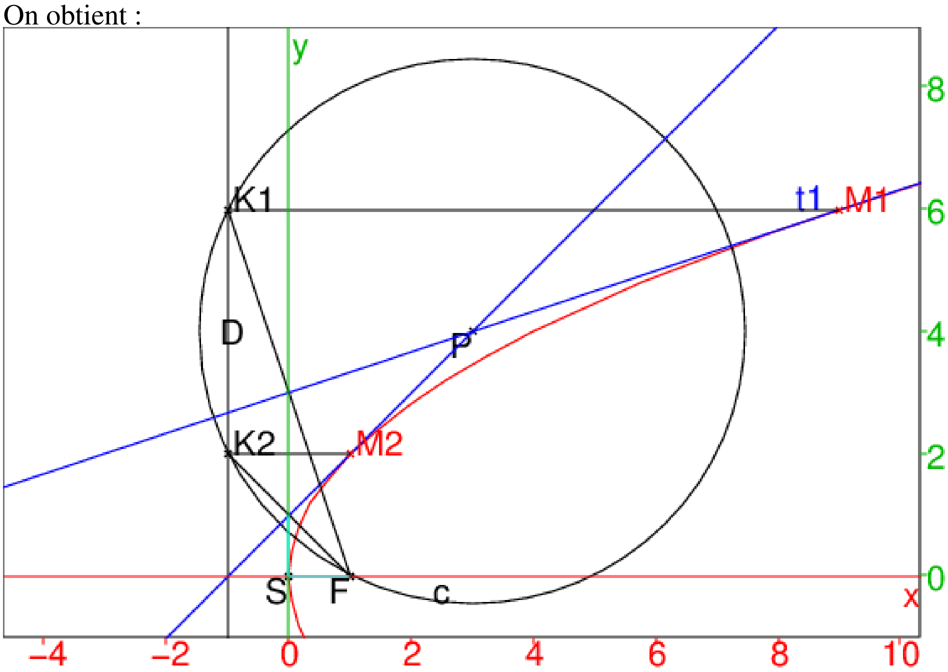
liste des instructions qui se trouve dans `geo11` (faire `Charger session` du menu `Fich` de `Xcas` et selectionner `geo11` du répértoire `examples/geo` pour exécuter ce fichier).
Voici le détail de `geo11` :
`E:=droite(A,B)`
trace la droite E passant par A et B,
`M:=mediatrice(C,D)`
trace la médiatrice de CD,
`H:=projection(E,C)`
définit la projection orthogonale de C sur E,
`P:=parabole(C,milieu(C,H)))`
trace la parabole de foyer C et de directrice E,
`N:=inter(P,M)`
N est la liste des points d'intersection de la droite M et de la parabole P,
`C1:=cercle(N[0],C-N[0])`
trace le cercle de centre `N[0]` passant par `C` : il repond à la question.
`C2:=cercle(N[1],C-N[1])` trace le cercle de centre `N[1]` passant par `C` : il repond à la question.
`Q:=parabole(D,milieu(D,projection(E,D)))` trace la parabole de foyer D et de directrice E, elle passe par N. Cela prouve qu'il n'y a pas d'autres solutions.

1.25 Tangente à une conique donnée passant par un point P donné

— **La parabole Rappel**
La parabole est le lieu géométrique du centre M d'un cercle tangent à une droite fixe D (la directrice) et passant par un point fixe F (le foyer). MF est le rayon vecteur du point M.
La distance $FH = p$ du foyer à la directrice est le paramètre de la parabole.Le milieu O de FH est le sommet de la parabole.
l'équation de la parabole dans le repère Oxy (Ox dirigé selon F) est : $y^2 - 2px = 0$.
En tout point d'une parabole il y a une tangente.
La tangente en un point M d'une parabole est la médiatrice du segment MK où K est la projection orthogonale de M sur la directrice D. C'est aussi la bissectrice intérieure $\widehat{KMF}$.
Tangente à une parabole passant par un point P donné
Puisque les tangentes sont les médiatrices des segments FK pour $K \in D$:
- le symétrique K du foyer F par rapport à une tangente se trouve sur la directrice D et
- la médiatrice de FK est une tangente ; la tangente à construire passe par P donc on a $PF = PK$.
Avec `Xcas`, `parabole(F,S)` trace la parabole de foyer `F` et de sommet `S`.
On tape :
`F:=point(1);S:=point(0);D:=droite(x=-1);`

```
affichage(parabole(F,S),1);P:=point(3+4*i);
c:=cercle(P,F-P);K1,K2:=inter(C,D);
segment(K2,F);segment(K1,F);
t1:=affichage(mediatrice(F,K1),4);
t2:=affichage(mediatrice(K2,F),4);
M1:=affichage(inter_unique(t1, parallele(K1,droite(F,S))),1);
M2:=affichage(inter_unique(t2, parallele(K2,droite(F,S))),1);
segment(K2,M2);segment(K1,M1);
```

On obtient :



Donc K est l'intersection de la directrice D avec le cercle c de centre P et de rayon PF. Si le cercle de centre P et de rayon PF coupe la directrice en 2 points K_1 et K_2, il existe 2 tangentes à la parabole qui passent par P : ce sont les médiatrices de FK_1 et de FK_2. Les points de contact M_1 et M_2 sont sur les parallèles à l'axe, issues de K_1 et K_2.

Condition d'existence

Soit d la distance de P à la directrice D :

si $d < PF$ i.e. (P est extérieur à la parabole), le cercle c de centre P et de rayon PF coupe la directrice en 2 points : on peut donc mener par P 2 tangentes distinctes,

si $d = PF$ i.e. (P est sur la parabole), le cercle c de centre P et de rayon PF coupe la directrice en 1 point : on peut donc mener 1 tangente qui est la tangente en P,

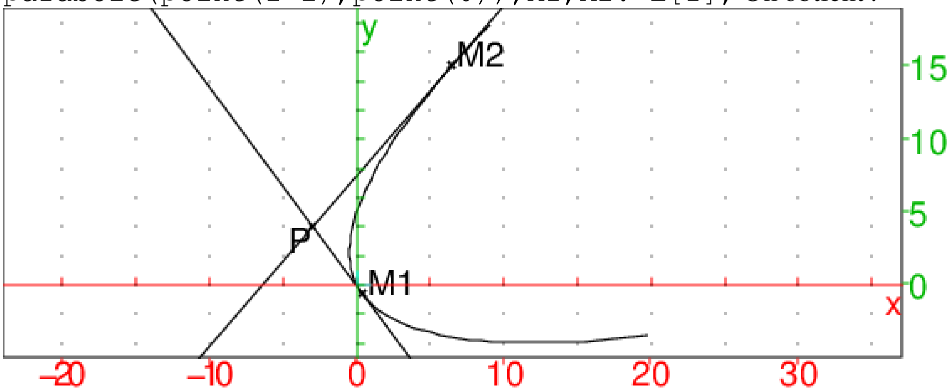
si $d > PF$ i.e. (P est intérieur à la parabole), le problème est impossible.

De façon plus générale, on tape :

```
tanparabole(P,F,S):={
local D,H,I,K1,K2,d,l,t1,t2,M1,M2;
H:=symetrie(S,F);
D:=perpendiculaire(H,droite(F,S));
I:=projection(D,P);
d:=longueur(P,I);
```

```
l:=longueur(P,F);
si d> l alors retourne []; fsi;
K1,K2:=inter(cercle(P,l),D);
t1:=mediatrice(F,K1);
t2:=mediatrice(F,K2);
M1:=inter_unique(t1, parallele(K1,droite(F,S)))
M2:=inter_unique(t2, parallele(K2,droite(F,S)));
retourne [mediatrice(F,K1), mediatrice(F,K2)],[M1,M2];
}:;
```

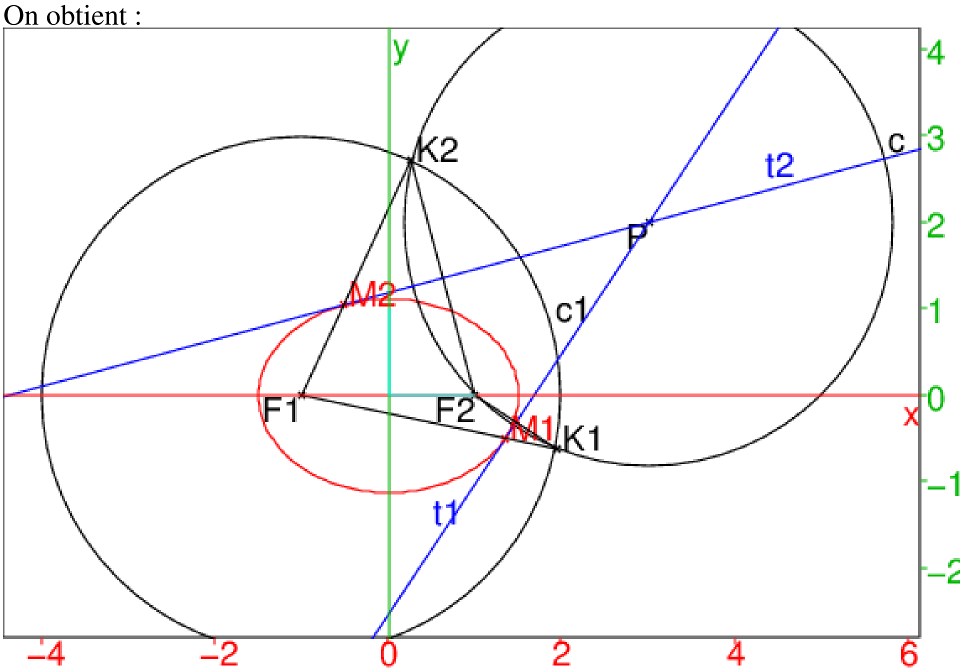
Puis on tape :
`P:=point(-3+4*i);L:=tanparabole(P,point(2+i),point(0));`
`parabole(point(2+i),point(0));M1,M2:=L[1];` On obtient :



— **L'ellipse**
Rappel
L'ellipse est le lieu géométrique des points M dont la somme des distances à deux points fixes F_1 et F_2 est constante et égale à $2a$.
F_1 et F_2 sont les foyers de l'ellipse et ils jouent le même rôle.
$F_1F_2 = 2c$ est la distance focale, on a $c < a$; F_1F_2 est l'axe focal.
Les segments MF_1 et MF_2 sont les rayons vecteurs du point M.
Le cercle c_1 (resp c_2) de centre F_1 (resp F_2) et de rayon $2a$ est le cercle directeur associé à F_1 (resp F_2).
En tout point d'une ellipse il y a une tangente.
La tangente en un point M d'une ellipse est la médiatrice du segment F_2K_1 où K_1 est l'intersection de la droite F_1M avec le cercle directeur de F_1.
C'est aussi la bissectrice intérieure $\widehat{F_2MK_1}$ i.e. la bissectrice extérieure de l'angle des rayons vecteurs.
Tangente à une ellipse passant par un point P donné
Puisque les tangentes de l'ellipse sont les médiatrices des segments F_2K pour $K \in c_1$ où c_1 est le cercle directeur associé à F_1 :
- le symétrique K du foyer F par rapport à une tangente se trouve sur le cercle directeur c_1 associé à F_1 et
- la médiatrice de FK est une tangente est, la tangente à construire passe par P donc on a $PF = PK$.
On tape : `P:=point(3+2*i);F1:=point(-1);F2:=point(1);`
```
affichage(ellipse(F1,F2,3/2),1);
c1:=cercle(F1,3);c:=cercle(P,longueur(P,F2));
K1,K2:=inter(c1,c);
```

```
t1:=affichage(mediatrice(F2,K1),4);
t2:=affichage(mediatrice(F2,K2),4);
M1:=affichage(inter_unique( droite(F1,K1),t1),1);
M2:=affichage(inter_unique(t2, droite(F1,K2)),1);
segment(F2,K1);segment(F2,K2);
segment(F1,K1);segment(F1,K2);
```

On obtient :



Donc K est l'intersection du cercle directeur c_1 associé à F_1 avec le cercle c de centre P et de rayon PF_2. Si le cercle c coupe c_1 en 2 points K_1 et K_2, il existe 2 tangentes à l'ellipse qui passent par P : ce sont les médiatrices de F_2K_1 et de F_2K_2. Les points de contact M_1 et M_2 sont sur les rayons vecteurs F_1K_1 et F_1K_2.

Condition d'existence

Pour que K_1 et K_2 existent, il faut que le cercle c de centre P et de rayon PF coupe c_1 en 2 points ou 1 point : si $2a < PF_1 + PF_2$ (P est à l'extérieur de l'ellipse) les 2 cercles se coupent en 2 points, on peut donc mener par P 2 tangentes distinctes,
si $2a = PF_1 + PF_2$, (P est sur l'ellipse), les 2 cercles sont tangents : on peut donc mener 1 tangente qui est la tangente en P,
si $2a > PF_1 + PF_2$ (P est à l'intérieur de l'ellipse) : le problème est impossible.

On tape

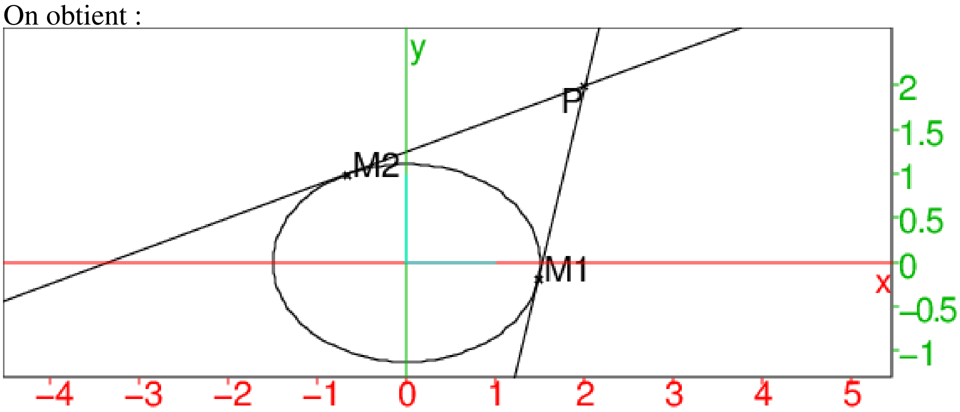
```
tanellipse(P,F1,F2,a):={
local c1,c,K1,K2,l1,l2,t1,t2,M1,M2;
l1:=longueur(P,F1);
l2:=longueur(P,F2);
si l1+l2<2a alors retourne [] fsi;
c1:=cercle(F1,2a);
c:=cercle(P,l2);
[K1,K2]:=inter(c1,c);
```

```
t1:=mediatrice(F2,K1);
t2:=mediatrice(F2,K2);
M1:=inter_unique(t1, droite(F1,K1));
M2:=inter_unique(t2, droite(F1,K2));
retourne [t1,t2],[M1,M2];
}:;
```

On tape :

```
P:=point(2+2*i);L:=tanellipse(P,point(-1),point(1),3/2);
M1,M2:=L[1];ellipse(-1,1,3/2);
```

On obtient :



— **L'hyperbole**

Rappel

L'hyperbole est le lieu géométrique des points M dont la différence des distances à deux points fixes F_1 et F_2 est constante et égale à $2a$ ($a \neq 0$).
F_1 et F_2 sont les foyers de l'hyperbole et ils jouent le même rôle.
$F_1F_2 = 2c$ est la distance focale, on a $c > a$; F_1F_2 est l'axe focal.
Les segments MF_1 et MF_2 sont les rayons vecteurs du point M.
Le cercle c_1 (resp c_2) de centre F_1 (resp F_2) et de rayon $2a$ est le cercle directeur associé à F_1 (resp F_2).
En tout point d'une hyperbole il y a une tangente.
La tangente en un point M d'une hyperbole est la médiatrice du segment F_2K_1 où K_1 est l'intersection de la droite F_1M avec le cercle directeur de F_1.
C'est aussi la bissectrice intérieure $\widehat{F_2MK_1}$ i.e. la bissectrice extérieure de l'angle des rayons vecteurs.

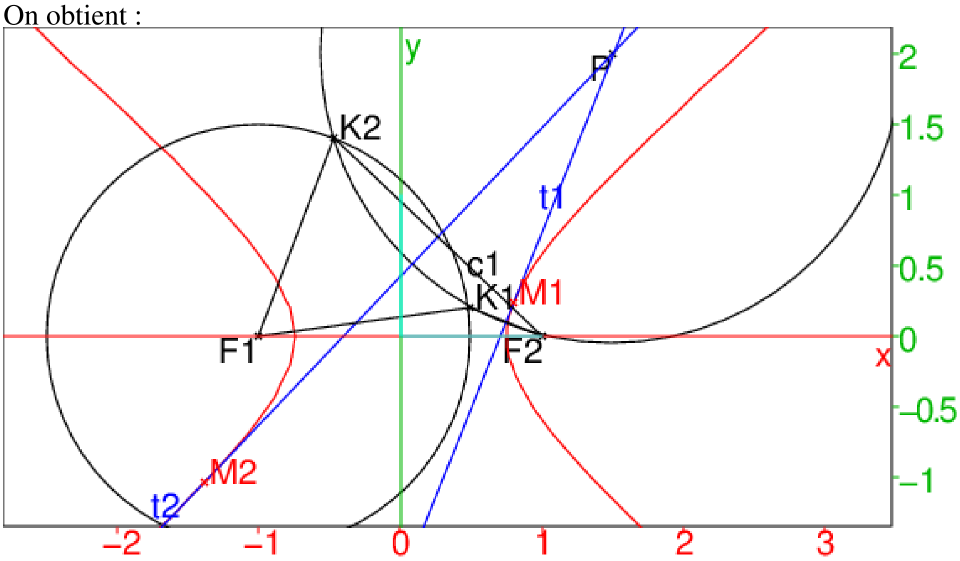
Tangente à une hyperbole passant par un point P donné

Puisque les tangentes de l' hyperbole sont les médiatrices des segments F_2K pour $K \in c_1$ où c_1 est le cercle directeur associé à F_1 :
- le symétrique K du foyer F_2 par rapport à une tangente se trouve sur le cercle directeur c_1 associé à F_1 et
- la médiatrice de F_2K est une tangente et la tangente à construire passe par P donc on a $PF = PK$.

```
P:=point(1+2*i);F1:=point(-1);F2:=point(1);
affichage(hyperbole(F1,F2,3/4),1);
c1:=cercle(F1,3/2);c:=cercle(P,longueur(P,F2));
K1,K2:=inter(c1,c);
t1:=affichage(mediatrice(F2,K1),4);
```

```
t2:=affichage(mediatrice(F2,K2),4);
M1:=affichage(inter_unique( droite(F1,K1),t1),1);
M2:=affichage(inter_unique(t2, droite(F1,K2)),1);
segment(F2,K1);segment(F2,K2);
segment(F1,K1);segment(F1,K2);
```

On obtient :



Donc K est l'intersection du cercle directeur c_1 associé à F_1 avec le cercle c de centre P et de rayon PF_2. Si le cercle c de centre P et de rayon PF_2 coupe c_1 en 2 points K_1 et K_2, il existent 2 tangentes à l' hyperbole qui passent par P : ce sont les médiatrices de F_2K_1 et de F_2K_2. Les points de contact M_1 et m_2 sont sur les rayons vecteurs F_1K_1 et F_1K_2.

Condition d'existence

Pour que K_1 et K_2 existent, il faut que le cercle c de centre P et de rayon PF coupe c_1 en 2 points ou 1 point : si $|PF_1 - PF_2| < 2a$ (P est à l'extérieur de l'hyperbole) les 2 cercles se coupent en 2 points, on peut donc mener par P 2 tangentes distinctes,

si $2a = |PF_1 - PF_2|$, (P est sur l'hyperbole), les 2 cercles sont tangents : on peut donc mener 1 tangente qui est la tangente en P,

si $2a < |PF_1 - PF_2|$ (P est à l'intérieur de l'hyperbole) : le problème est impossible.

On tape

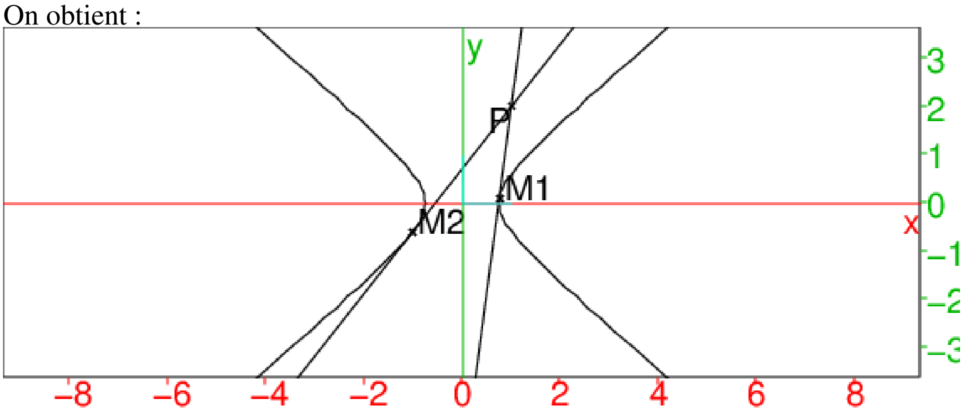
```
tanhyperbole(P,F1,F2,a):={
local c1,c,K1,K2,l1,l2,t1,t2,M1,M2;
l1:=longueur(P,F1);
l2:=longueur(P,F2);
si abs(l1-l2)>2a alors retourne [] fsi;
c1:=cercle(F1,2a);
c:=cercle(P,l2);
[K1,K2]:=inter(c1,c);
t1:=mediatrice(F2,K1);
t2:=mediatrice(F2,K2);
M1:=inter_unique(t1, droite(F1,K1))
M2:=inter_unique(t2, droite(F1,K2));
```

```
retourne [t1,t2],[M1,M2];
}:;
```

On tape :

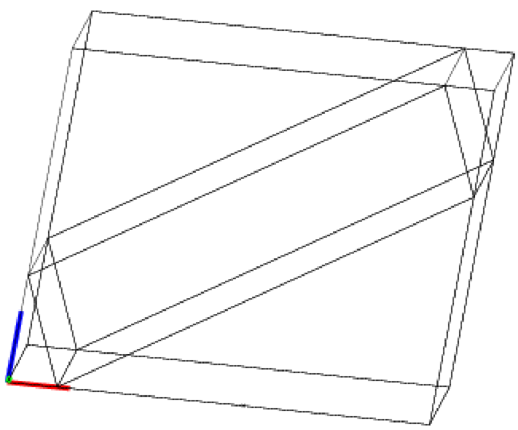
```
P:=point(1+2*i);L:=tanhyperbole(P,point(-1),point(1),3/4);
hyperbole(point(-1),point(1),3/4);M1,M2:=L[1];
```

On obtient :

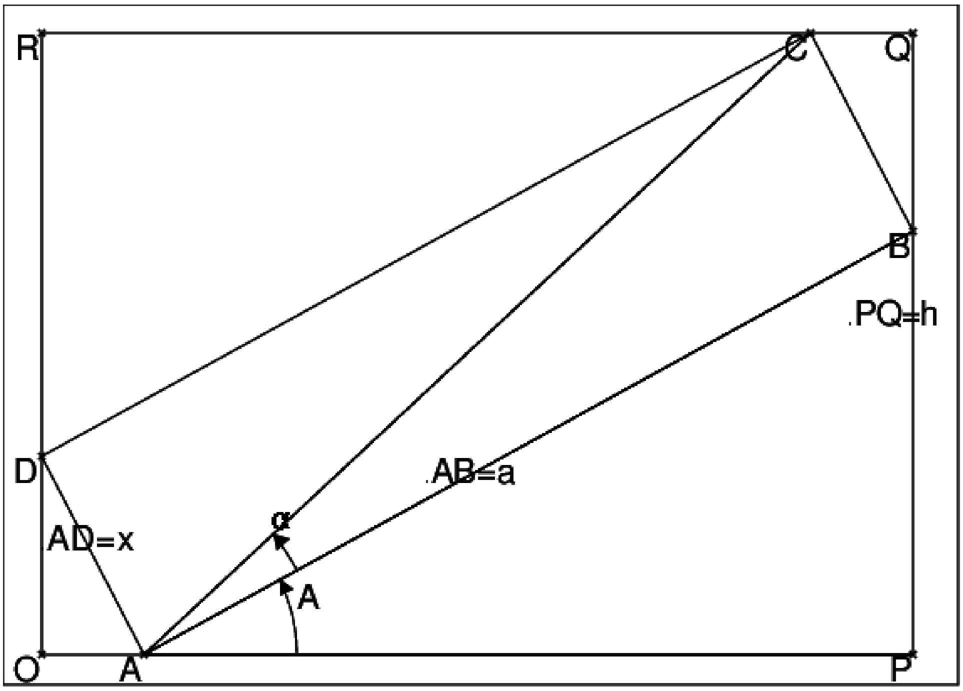


1.26 Exercice : le paquet

Dans une boîte aux lettres ayant la forme d'un parallélépipède rectangle (de hauteur h et de base carrée de côté a), le facteur dépose un colis ayant la forme d'un parallélépipède rectangle (de hauteur x et de base carrée de côté a). Il se trouve que par maladresse le facteur a coincé le colis en position inclinée entre les parois de la boîte aux lettres :



Voici la vue de face lorsqu'on ouvre la boîte aux lettres :



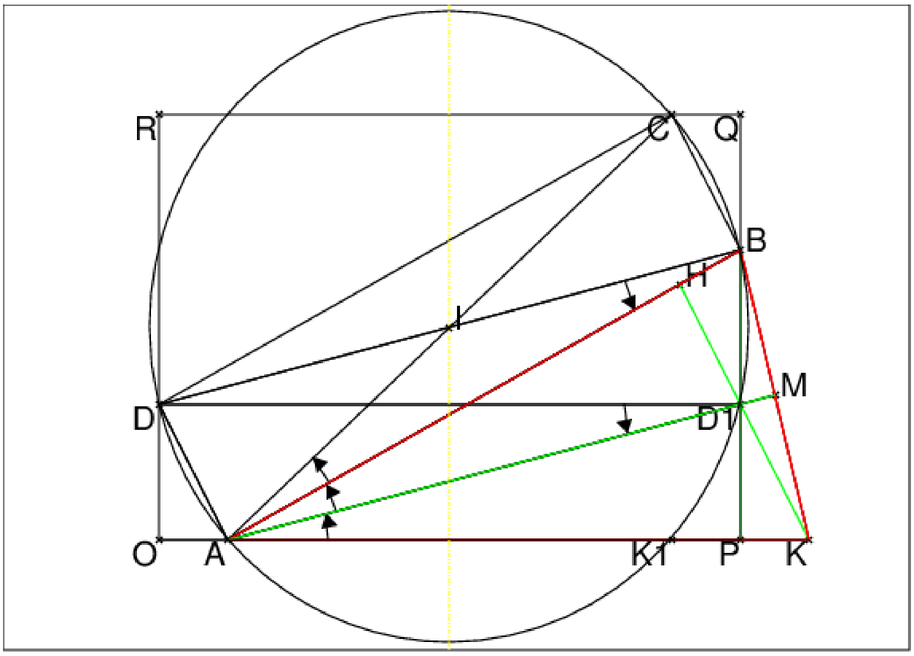
On note :
$OPQR$ l'ouverture de la boîte aux lettres,
$ABCD$ la face rectangulaire visible du paquet lorsqu'on ouvre la boîte aux lettres.
$\widehat{A}$ l'angle d'inclinaison du paquet i.e $\widehat{PAB}$ et α l'angle $\widehat{BAC}$.
Montrer que :
$\widehat{A} = 2\alpha,\ x = a\tan(\widehat{A}/2),\ \ \sin(\alpha) = \dfrac{x}{\sqrt{a^2+x^2}}$
$h^2 = (x^2+a^2)\sin(3\alpha)^2$.
Trouver une relation entre a, h, x.

Application numérique

Déterminer x lorsque $a = 68$, $h = 68$ puis lorsque $a = 68$, $h = 47$.

Solution

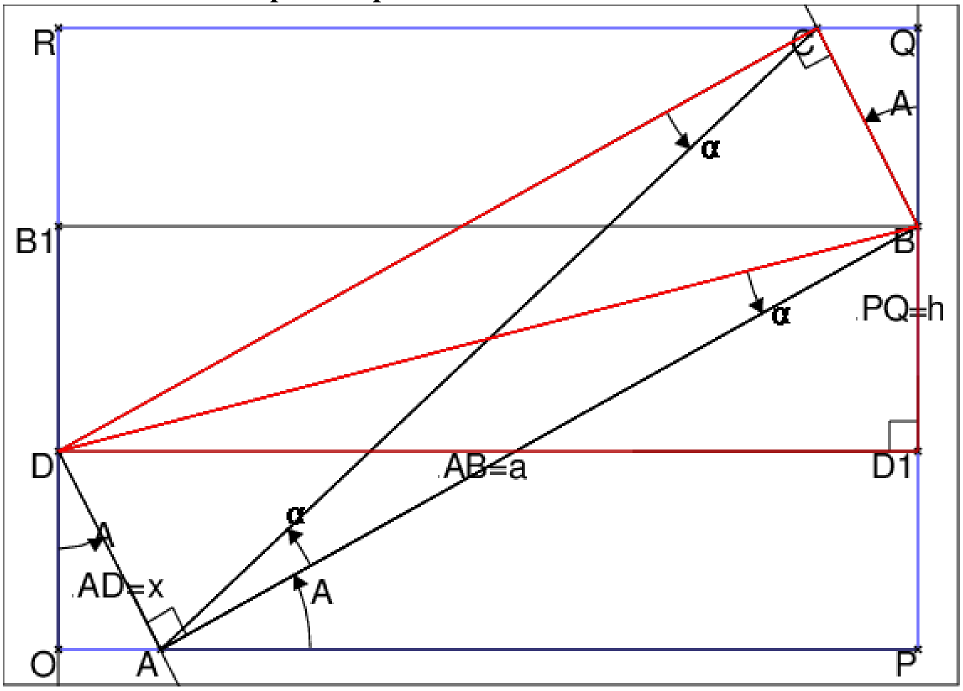
Pour montre que $\widehat{A} = 2\alpha$, on va mettre en évidence le fait que $AB = a$ en formant le triangle isocèle AKB pour cela on définit K par :
$\overrightarrow{AK} = \overrightarrow{OP} = a$.
BP est la hauteur issue de B du triangle ABK.
Soit le parallélogramme AKD_1D.



On a :
$\overrightarrow{DD_1} = \overrightarrow{AK = a}$
donc D_1 se trouve sur PQ et donc sur BP AD est perpendiculaire à AB (puisque $ABCD$ est un rectangle) donc KD_1 est perpendiculaire à AB donc KD_1 est la hauteur issue de K du triangle ABK.
BP et KD_1 se coupent en D_1 donc AD_1 est la hauteur issue de A du triangle isocèle ABK.
Le triangle ABK est isocèle de sommet A donc AD_1 est donc aussi la bissectrice de l'angle $\widehat{A}$ et la médiatrice de BK.
Soit M le milieu de BK Soit I le centre du rectangle $ABCD$.
Soit K_1 le symétrique de K par rapport à P.
La médiatrice de AK_1 est aussi la médiatrice de DD_1 et elle passe par I.
Le cercle de diamètre AC (centre I et rayon =AI) passe donc par A, B, C, D, D_1 et K_1.
Donc $alpha = \widehat{BDA} = \widehat{AK_1D}$ car ce sont des angles inscrits qui intercéptent la même corde puisque $BC = AD = K_1D_1 = KD_1$.
On a donc $\widehat{A} = 2\alpha$
Le triangle AK_1C est rectangle en K_1 (les triangles OAD et QCB sont égaux donc $CQ = OA = PK = PK_1$) donc :
$\sin(3\alpha) = \frac{h}{\sqrt{a^2+x^2}}$.
On sait que $\sin(3\alpha) = 3\sin(\alpha) - 4\sin(\alpha)^3$ et que $x\sqrt{x^2+a^2}\sin(\alpha)$ donc :
$h = \sqrt{a^2+x^2}\sin(3\alpha) = 3 * x - 4x^3/(x^2+a^2)$
$h = 3 * x - 4x^3/(a^2+x^2)$
$h(x^2+a^2) - 3x(x^2+a^2) + 4x^3 = 0$ D'où la relation :

$$x^3 + hx^2 - 3a^2x + a^2h = 0$$

Autre démonstration plus simple



Soient D_1 la projection de D sur BC et B_1 la projection de B sur OR.
Les triangles rectangles BCD et BD_1D sont égaux car ils ont 2 côtés egaux (BD en commun et $DC = DD_1 = a$) donc d'après Pythagore $BC = BD_1$.
Cela entraine que :
$\widehat{D_1DB} = \widehat{BDC} = \alpha$ et on a :
$\cos(alpha)^2 = a^2/(x^2 + a^2)$.
$\widehat{PAB} = \widehat{D_1DC\widehat{A}} = 2\alpha$ (angles à côtés parallèles)
$\widehat{ODA} = \widehat{PAB} = \widehat{QBC}$ (angles à côtés perpendiculaires)
Puisque les triangles rectangles BCD et BD_1D sont égaux, les rectangles $ABCD$ et DD_1BB_1 sont égaux donc :
$BD_1 = x$ Cherchons une équation reliant x, h,et a.
On a :
$OD = B_1R = AD\sin(\widehat{A}) = x\cos(\widehat{A}) = x\cos(2\alpha)$
Donc :
$h = 2 * OD + DB_1 = 2x\cos(2\alpha) + x =$
$h = 2x\cos(2\alpha) + x = 4x\cos(\alpha)^2 - 2x + x = x(4a^2/(x^2 + a^2) - 1)$
$h(x^2 + a^2) = 4a^2x - x^3 - a^2x$.
D'où la relation :

$$x^3 + hx^2 - 3a^2x + a^2h = 0$$

Application numérique
$a = h = 68$
On tape :
```
solve(68=3*x-4x^3/(68^2+x^2),x)
```
On obtient :
```
[-68-sqrt(2)*68,-68+sqrt(2)*68,68]
```
$a = 68$, $h = 47$
Donc $x = 68(\sqrt{2} - 1)$ si $\widehat{A} < \pi/2$ et $x = 68$ si $\widehat{A} = \pi/2$ On tape :

```
solve(x^3+47*x^2-3*68^2*x+47*68^2=0,x)
```

On obtient :

```
[-32-4*sqrt(863),17,-32+4*sqrt(863)]
```

Donc $x = 17$

Chapitre 2

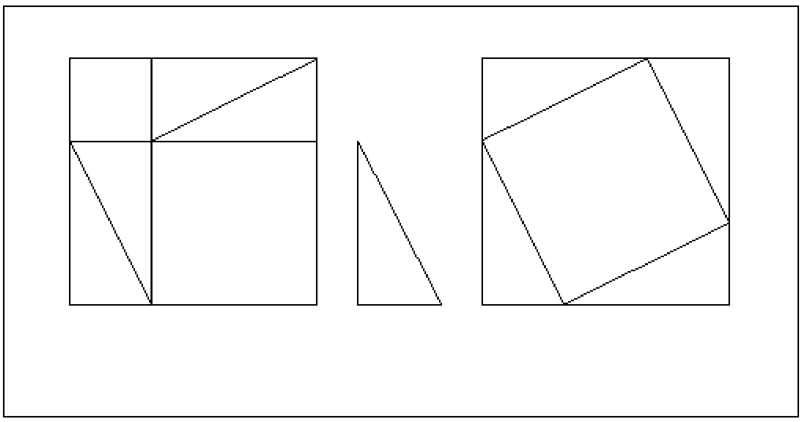
Le théorème de Pythagore

2.1 Le théorème et sa réciproque

Dans un triangle rectangle, le carré de l'hypoténuse est la somme des carrés des deux autres côtés.
Si dans un triangle ABC on a $BC^2 = AB^2 + AC^2$ alors le trianle est rectangle en A

2.2 Une démonstration qui saute aux yeux

On considère un triangle rectangle T de coés a, b, c (c est l'hypoténuse et on suppose $a \geq b$), on veut donc montrer que $a^2 + b^2 = c^2$.
On fait quatre copies de ce triangle T et deux copies C_1 et C_2 d'un carré de cotés $a + b$. On dispose les quatres copies du triangle dans les carrés C_1 et C_2 selon la figure ci-dessous :

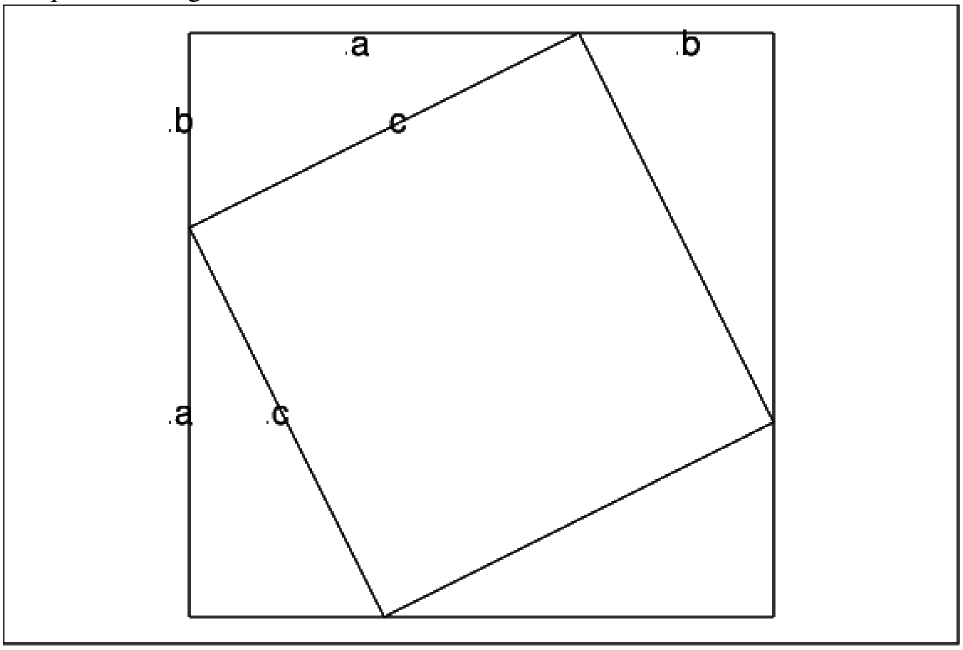


Dans le carré C_1 on a 4 triangles T et deux carrés l'un de cotés a (de surface a^2) et l'autre de cotés b (de surface b^2).
Dans le carré C_2 on a 4 triangles T et un carré de cotés c (de surface c^2).
Comme les surfaces de C_1 et de C_2 sont les mêmes on en déduit que :
$a^2 + b^2 = c^2$.

```
carre(-4-i,-1-i);
carre(1-i,4-i);
triangle_rectangle(-0.5-i,0.5-i,2);
segment(-3-i,-3+2*i);
segment(-4+i,-1+i);
segment(-3-i,-4+i);
segment(-3+i,-1+2*i);
segment(1+i,2-i);
segment(1+i,3+2*i);
segment(4,3+2*i);
segment(4,2-i);
```

2.3 Le théorème et la formule $(a+b)^2 = a^2 + 2ab + b^2$

On ne dessine que le carré C_2 et on a :
C_2 a comme surface $(a+b)^2$
les quatres triangles T ont comme surface $2ab$



Le carré central a comme surface c^2 et aussi $(a+b)^2 - 2ab$
Donc :
$(a+b)^2 - 2ab = c^2$
On sait que $(a+b)^2 = a^2 + 2ab + b^2$ (identité remarquable) donc
$c^2 = a^2 + b^2$

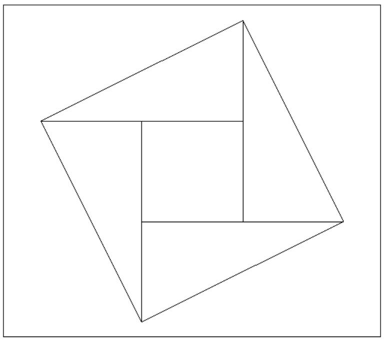
```
carre(1-i,4-i)
segment(1+i,2-i);
segment(1+i,3+2*i);
segment(4,3+2*i);
segment(4,2-i);
legende(0.9+i*1.5,"a");
legende(3.5+i*1.9,"a");
```

```
legende(0.9,"b");
legende(1.8+i*1.9,"b");
legende(2+i*1.5,"c");
legende(1.4,"c");
```

2.4 Le théorème et la formule $(a-b)^2 = a^2 - 2ab + b^2$

On considère un carré C de coté c et on place les quatres copies du triangles rectangle T selon la figure ci-dessous :

```
triangle_rectangle(0,1,2);
triangle_rectangle(i,2*i,2);
triangle_rectangle(-1+i,-2+i,2);
triangle_rectangle(-1,-1-i,2);
```

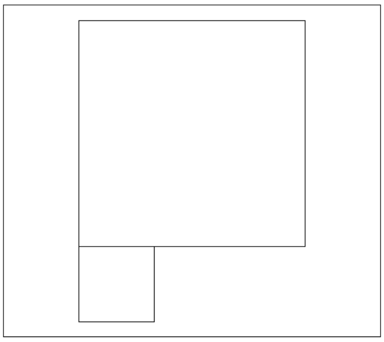


Le carré de coté c est composé d'un carré de cotés $a-b$ et de 4 triangles T donc on a : $(a-b)^2 + 2ab = c^2$
On sait que $(a-b)^2 = a^2 - 2ab + b^2$ (identité remarquable) donc
$c^2 = a^2 + b^2$.

2.5 Les puzzles et le théorème

2.5.1 Le théorème et un premier puzzle

Soit une forme constituée de deux carrés de cotés a et b (avec $a \geq b$) disposés selon la figure :

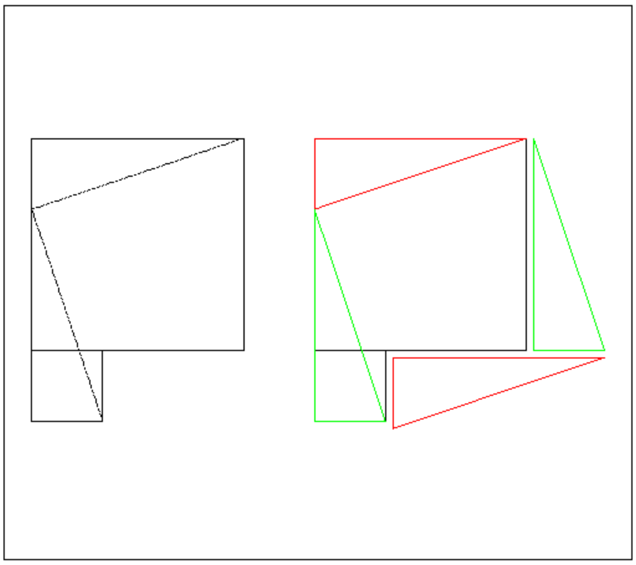


Comment découper cette forme pour faire un puzzle de trois pièces permettant de reconstituer un carré ?

Solution

Si le carré de côtés a a pour sommets $(0, a, a(1+i), ia)$, on joint les points $i(a-b)$ et $a(1+i)$ ainsi que les points $i(a-b)$ et $b(1-i)$.

On obtient :



On tape pour avoir cette figure :

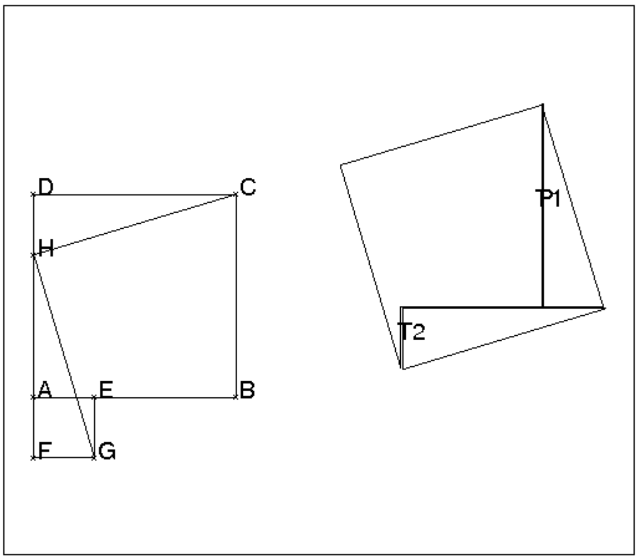
```
carre(-4-i,-1-i);
carre(-4-2*i,-3-2*i);
segment(-3-2*i,-4+i,affichage=ligne_tiret);
segment(-1+2*i,-4+i,affichage=ligne_tiret);
carre(-i,3-i);
carre(-2*i,1-2*i);
couleur(triangle_rectangle(2*i,i,3),rouge);
couleur(triangle_rectangle(1.1-1.1*i,1.1-2.1*i,3),rouge)
```

```
couleur(triangle_rectangle(-2*i,1-2*i,3),vert);
couleur(triangle_rectangle(3.1-i,4.1-i,3),vert)
```

On peut simuler un vrai puzzle en déplaçant avec la souris les trois pièces.
On tape :

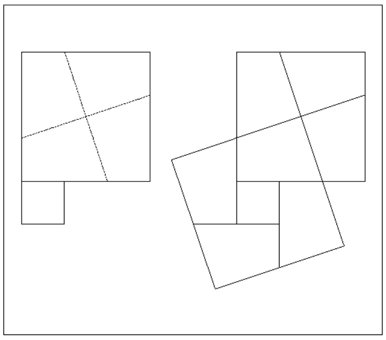
```
A:=point(-4,-1);
B:=point(-1,-1);
carre(A,B,C,D);
E:=element(droite(A,B),0.3);
carre(E,A,F,G);
H:=F+C-B;
segment(H,C);
segment(H,G);
P:=polygone(C,H,G,E,B);
T1:=triangle(H,G,F);
T2:=triangle(H,C,D);
```

Puis on bouge le polygône et les 2 triangles pour faire 1 grand carré : pour cela on se met en mode `Pointeur` et on clique sur le côté EB du polygône et on le déplace. Puis on clique sur le côté HC du triangle(H, C, D) et on le déplace pour amener D en E, puis on clique sur le côté HG du triangle(H, G, F) et on le déplace pour amener F en B.
Remarque : on peut tracer les segments HC et HG seulement apres avoir déplacé le polygône et les 2 triangles.
On obtient :



2.5.2 Le théorème et un autre puzzle

On dispose de deux carrés de côtés a et b ($a \geq b$).
Comment découper le carré de côtés a en quatre morceaux pour pouvoir faire, avec ces 4 morceaux et le carré de coté b, un puzzle de cinq pièces permettant de reconstituer un carré ?
Une solution
On pose $a - b = 2d$ et si le carré de côté a a pour sommets $(0, a, a(1 + i), ia)$, on joint les points id et $a + i(d + b)$ ainsi que les points $d + b$ et $d + ia$. Ces deux segments ont pour longueur $c = \sqrt{a^2 + b^2}$, et se coupent selon quatre angles droits qui deviendront les sommets du carré solution de côtés $c = \sqrt{a^2 + b^2}$.
On obtient le découpage du carré de côté a et le carré constitué des 5 pièces dans la figure ci-dessous :



Pour faire cette figure on tape :

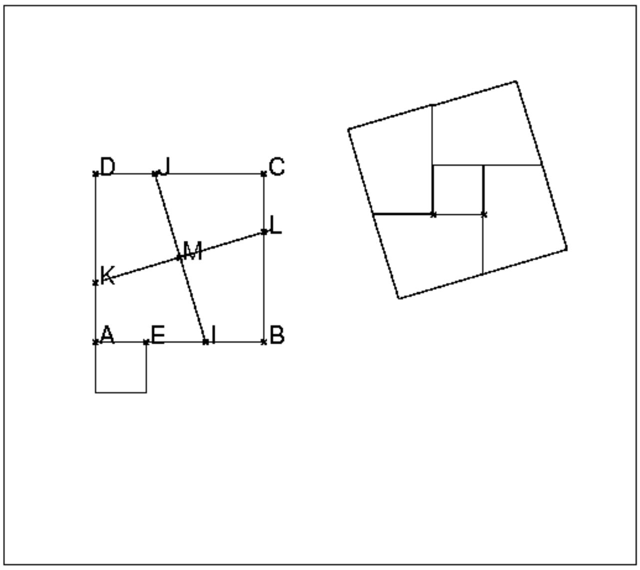
```
carre(-5-2i,-4-2i);
carre(-5-i,-2-i);
segment(-5,-2+i,affichage=ligne_tiret);
segment(-3-i,-4+2i,affichage=ligne_tiret);
carre(-i,3-i);
segment(1+2*i,2-i)
segment(0,3+i)
carre(-2*i,1-2*i);
segment(2-i,2.5-2.5*(i));
segment(1-2*i,1-3*(i));
segment(2.5-2.5*(i),-0.5-3.5*(i));
segment(1-2*(i),-1-2*(i));
segment(-1.5-0.5*i,0);
segment(-1.5-0.5*i,-1-2*i);
segment(-0.5-3.5*(i),-1-2*(i));
```

On peut simuler un vrai puzzle en déplaçant avec la souris les cinq pièces.
On tape :

```
A:=point(-4,-1);
B:=point(-1,-1);
carre(A,B,C,D):;
```

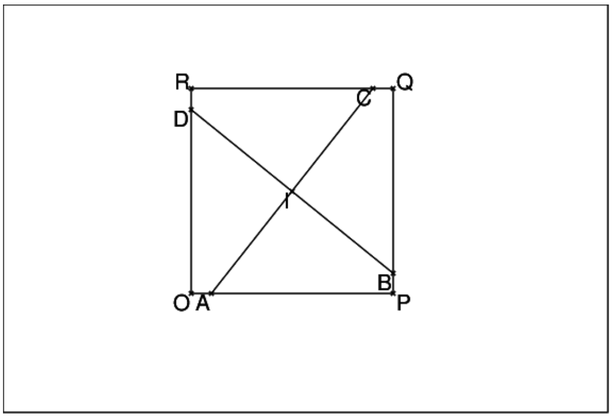
```
E:=element(droite(A,B),0.3);
carre(E,A,F,G):;
I:=milieu(E,B);
J:=D+B-I;
K:=rotation(A,pi/2,A+B-I);
L:=C+A-K;
segment(I,J);
segment(K,L);
M:=inter(droite(I,J),droite(K,L))[0];
quadrilatere(D,J,M,K);
quadrilatere(C,J,M,L);
quadrilatere(A,I,M,K);
quadrilatere(B,I,M,L);
quadrilatere(E,A,F,G);
```

Puis on bouge les 5 quadrilatères dans un grand carre : pour cela, on se met en mode pointeur et pour faciliter la sélection des quadrilatères on prend soin de ne pas dessiner le `carre(A,B,C,D):;` grâce au `:;`. On clique successivement sur un côté de chaque quadrilatère et on les déplace. Puis on dessine le `carre(A,B,C,D)` en enlevant le `:`
On obtient :



2.5.3 Ce puzzle comme exercice

Soient T un triangle rectangle dont les côtés de l'angle droit ont pour longueur a et b avec $a < b$. On pose $d = (b - a)/2)$.
On considére un carré $OPQR$ de côté b. Soient sur les côtés de ce carré les points A, B, C, D tels que : $OA = PB = QC = RD = d$ (cf figure) :



— Montrer que les segments AC et BD sont égaux et perpendiculaires et qu'ils se coupent au centre I du carré $OPQR$.
— Calculer la longueur de AC en fonction de a et b.
— Montrer que l'angle $\widehat{PAC}$ est égal à l'un des angles aigus du triangle T
— Montrer qu'avec les 4 morceaux $IAPB$, $IBQC$, $ICRD$, $IDOA$ et un carré de côté a, on peut faire un carré ayant un côte de la longueur de l'hypothénuse de T.

Solution

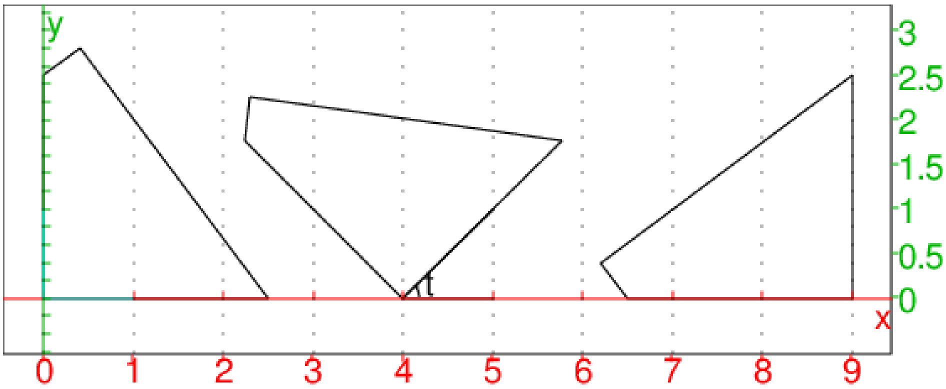
On tape :

```
morceau(a,b,o,t):={
local c,beta,d;
c:=sqrt(a^2+b^2);
d:=(b-a)/2;
beta:=atan(b/a);
return polygone(o,o+c/2*exp(i*t),o+c/2*exp(i*t)+(b-d)*exp(i*(t+pi-beta
  }:;
pythagore(a,b):={
local d,L;
L:=NULL;
si a>b alors d:=a;a:=b;b:=d; fsi;
d:=(b-a)/2;
L:=L,triangle(0,a,i*b);
L:=L,carre(a,0),carre(0,i*b),carre(i*b,a);
L:=L,segment(-b+i*(b-d),i*d),segment(-b+d,-d+i*b);
return L;
}:;
```

On tape :

```
morceau(3,4,0,0),morceau(3,4,4,pi/4),
angle(4,5,5+i,"t"),morceau(3,4,9,pi/2)
```

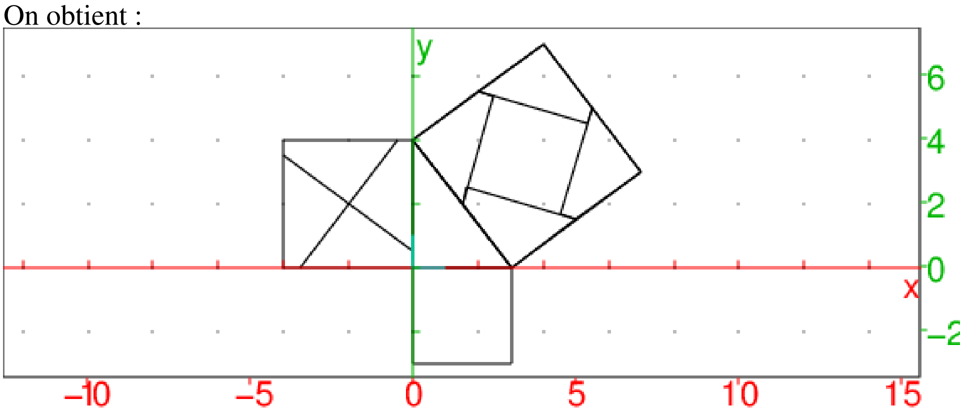
On obtient :



On tape :

```
pythagore(3,4),morceau(3,4,3,atan(3/4)),morceau(3,4,4*i,-atan(4/3)),
morceau(3,4,3+5*exp(i*atan(3/4)),atan(3/4)+pi/2),
morceau(3,4,4*i+5*exp(i*atan(3/4)),pi+atan(3/4))
```

On obtient :



2.5.4 Exercices

Exercice1 : octogone étoilé et Pythagore

Soit $ABCD$ un carré de centre I et de côté a et $A_1B_1C_1D1$ le carré obtenu on faisant tourner le carré $ABCD$ avec une rotation de centre O et d'angle $\pi/4$.

Soit M (resp N...T) l'intersection des segments AB et D_1A_1 (resp AB et A_1B_1,...,AD et A_1D_1).

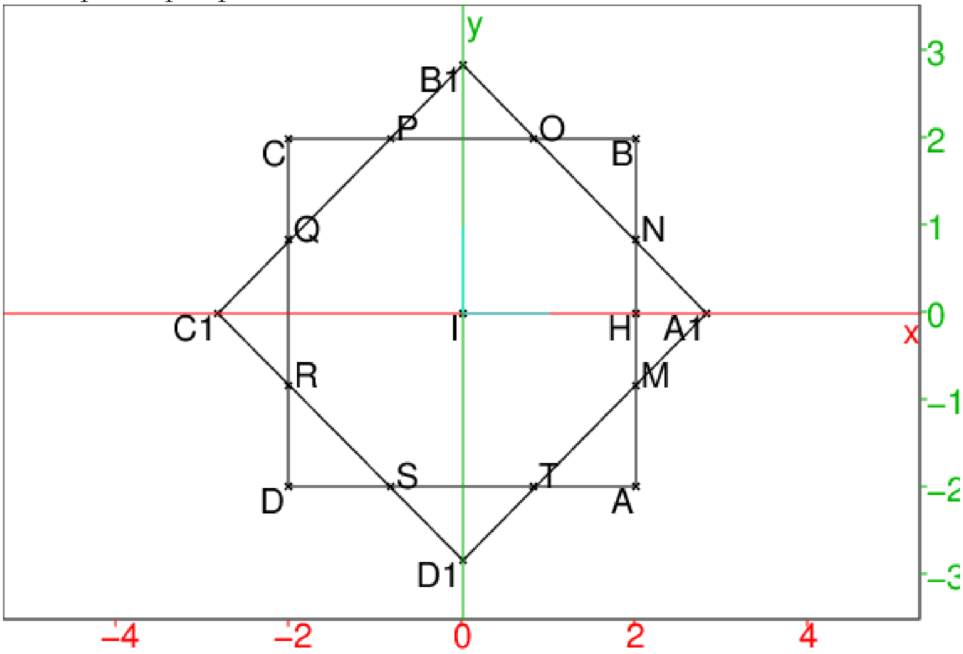
Soit H le point d'affixe $a/2$.

On tape :

```
assume(a=[4,0,10,0.1]); c0:=carre(a/2*(1-i),a/2*(1+i)):;c0;
A,B,C,D:=sommets(c0);I:=point(0);
c1:=rotation(I,pi/4,c0):;c1 A1,B1,C1,D1:=sommets(c1)
H:=point(a/2);
M:=inter_unique(segment(A,B),segment(A1,D1))
N:=inter_unique(segment(A,B),segment(A1,B1))
O:=inter_unique(segment(B,C),segment(A1,B1))
P:=inter_unique(segment(B,C),segment(B1,C1))
Q:=inter_unique(segment(D,C),segment(B1,C1))
R:=inter_unique(segment(D,C),segment(D1,C1))
```

```
S:=inter_unique(segment(A,D),segment(D1,C1))
T:=inter_unique(segment(A,D),segment(A1,D1)) affichage(polygone(A,M,A1
```

On obtient les sommets d'un octogone $AA_1BB_1CC_1DD_1$ et un octogone étoilé $AMA_1NBB_1..D_1TA$:



1. Calculer la longueur IA_1,
2. Calculer la longueur de HA_1 en fonction de a,
3. Calculer la longueur MN en fonction de a,
4. Calculer l'aire st du triangle A_1NM,
5. Calculer l'aire so de l'octogne étoilé,
6. Montrer que les segments RN et TP dont égaux et perpendicilaires. Calculer leur longueur c en fonction de a et en dédure que $c^2 = so + st$.
7. Comment découper l'octogone étoilé en 8 morceaux pour reconstituer un carré avec ces 8 morceaux ?

Solution

1. $IA_1 = IA = a/(2\sqrt{2})$
2. Le triangle A_1NM est rectangle isocèle donc I est le milieu de MN et $HM = HN = HA_1 = IA_1 - IH = a/(2\sqrt{2}) - a/2 = a/2(\sqrt{2} - 1)$
3. $MN = 2HM = 2HA_1 = a(\sqrt{2} - 1)$
4. l'aire du triangle A_1NM vaut donc : $st = (a/2(\sqrt{2} - 1))^2 = (a^2(\sqrt{2} - 1)^2/4 = a^2(3 - 2\sqrt{(2)})/4$
5. l'aire de l'octogone étoilé vaut donc : $so = a^2 + 4 * st = a^2 + a^2(3 - 2\sqrt{(2)}) = 2a^2(2 - \sqrt{(2)})$

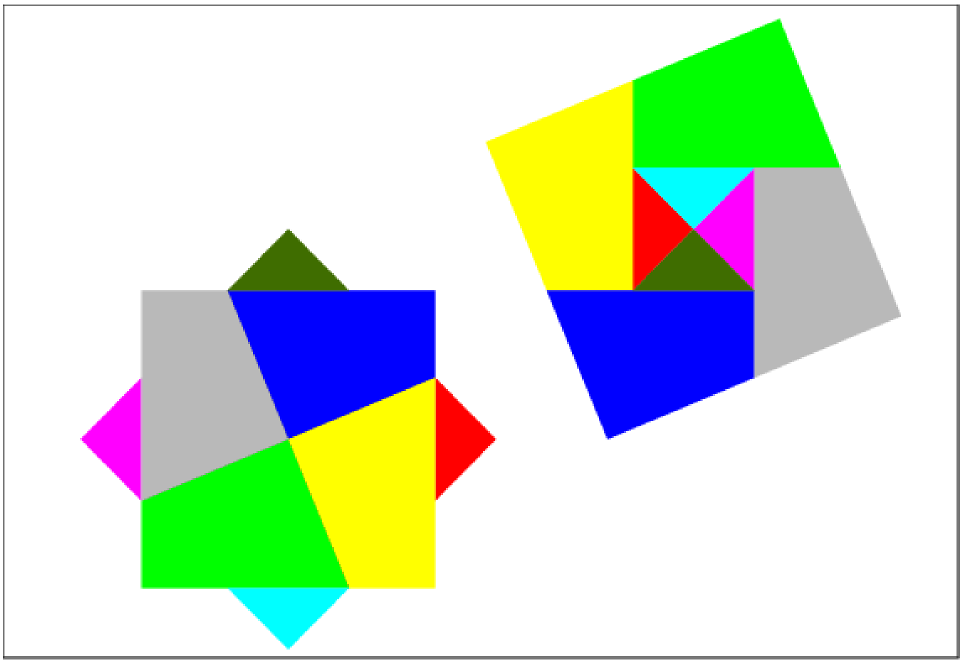
6. RN (resp PT) est les hypothénuse du triangle rectangle MNR (resp OPT).
   Le triangle OPT se déduit de MNR par une rotation de centre I et d''angle $\pi/2$ donc $RN = TP = a^2 + MN^2 = a^2 + a^2(3 - 2\sqrt{(2)})/4$

7. on a $4 * st = a^2(\sqrt{2} - 1))^2 = (a(\sqrt{2} - 1))^2$ donc $b = a(sqrt(2) - 1)$
   $so = a^2 + b^2$. Donc c est l'hypothénuse d'un triangle rectangle de côtés a,b.

8. on découpe l'octogone étoilén 8 morceaux qui sont :
   les 4 triangles rectangles $A1NM$, $B1PO$, $C1RQ$,$D1TS$ et les 4 quadrilatères $ITAN$, $INBP$, $IPCR$, $IRDT$.
   On tape :

```
supposons(a=[4.6,0,10,0.1]);
c0:=carre(a/2*(1-i),a/2*(1+i)):;
A,B,C,D:=sommets(c0):;I:=point(0):;
c1:=rotation(I,pi/4,c0):;;
A1,B1,C1,D1:=sommets(c1):;;
M:=inter_unique(segment(A,B),segment(A1,D1)):;
N:=inter_unique(segment(A,B),segment(A1,B1)):;
O:=inter_unique(segment(B,C),segment(A1,B1)):;
P:=inter_unique(segment(B,C),segment(B1,C1)):;
Q:=inter_unique(segment(D,C),segment(B1,C1)):;
R:=inter_unique(segment(D,C),segment(D1,C1)):;
S:=inter_unique(segment(A,D),segment(D1,C1)):;
T:=inter_unique(segment(A,D),segment(A1,D1)):;
H:=point(a/2):;;
//affichage(polygone(A,M,A1,N,B,O,B1,P,C,Q,C1,R,D,S,D1,T),1+epaisseur_li
segment(R,N);
segment(T,P);
affichage(polygone(M,A1,N),1+rempli);
affichage(polygone(I,N,B,P),4+rempli);
P1:=translation((P-C),P):;
affichage(translation(5+(N-R)+(P-T),polygone(I,R,D,T)),2+rempli);
affichage(translation(5+(P1-M),polygone(A1,M,N)),1+rempli);
affichage(polygone(I,R,D,T),2+rempli);
affichage(polygone(I,T,A,N),3+rempli);
affichage(translation(5+(P-T),polygone(I,T,A,N)),3+rempli);
affichage(translation(5+(B-R),polygone(C1,R,Q)),5+rempli);
affichage(polygone(C1,R,Q),5+rempli);
affichage(translation(5+(N-R)+(P-T)-(P1-P),polygone(D1,T,S)),6+rempli);
affichage(polygone(D1,T,S),6+rempli);
affichage(translation(5,polygone(I,N,B,P)),4+rempli);
affichage(polygone(I,P,C,R),47+rempli);
affichage(translation(5+(N-R),polygone(I,P,C,R)),47+rempli);
affichage(polygone(P,O,B1),67+rempli);
affichage(translation(5+(P1-P),polygone(P,O,B1)),67+rempli);
```

   On obtient :

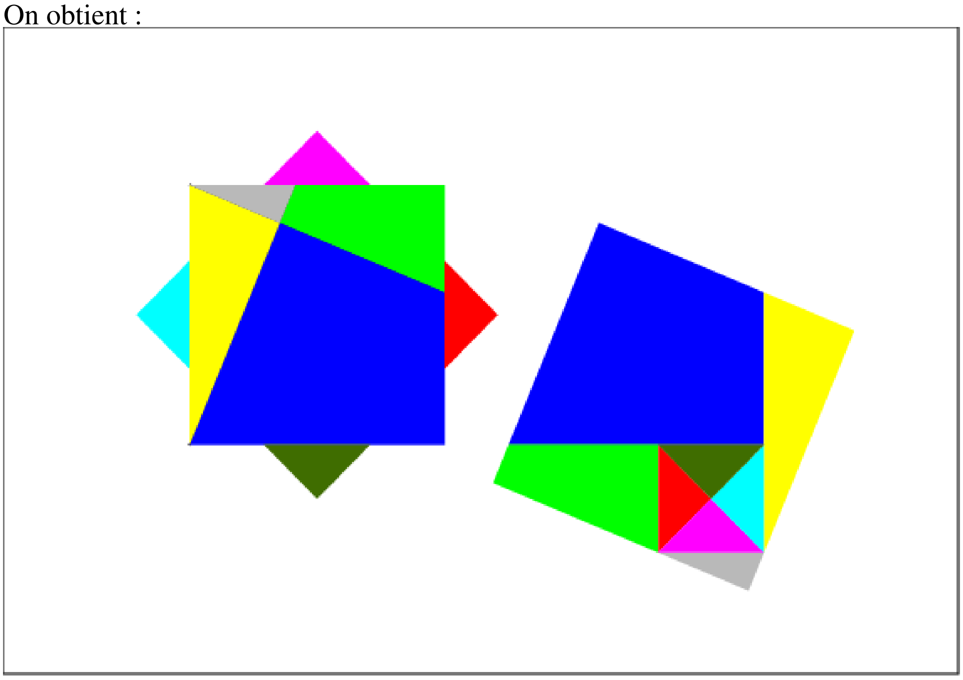


Exercice2 : autre découpage
Trouver d'autres solutions.
On peut en effet découper le carre $ABCD$ selon n'importe quelle parallèle à IJ et KL...à vous de le montrer !
On tape par exemple :

```
c:=carre(2-2i,+2+2i):; c:;A,B,C,D:=sommets(c):;I:=point(0);;
c1:=rotation(I,pi/4,c):;c1:;;
A1,B1,C1,D1:=sommets(c1):;;
M:=inter_unique(segment(A,B),segment(A1,D1)):;;
N:=inter_unique(segment(A,B),segment(A1,B1)):;;
T:=inter_unique(segment(A,D),segment(A1,D1)):;;
O:=inter_unique(segment(B,C),segment(A1,B1)):;;
P:=inter_unique(segment(B,C),segment(B1,C1)):;;
Q:=inter_unique(segment(D,C),segment(B1,C1)):;;
R:=inter_unique(segment(D,C),segment(D1,C1)):;;
S:=inter_unique(segment(A,D),segment(D1,C1)):;;
H:=point(2):;;
M1:=B+(M-N):;;
P1:=C+(O-P):;;
segment(C,M1);
segment(D,P1);
J:=inter_unique(droite(C,M1),droite(D,P1));
affichage(polygone(J,M1,B,P1),2+rempli);
affichage(polygone(J,D,A,M1),4+rempli);
affichage(polygone(J,D,C),3+rempli);
affichage(polygone(J,P1,C),47+rempli);
affichage(polygone(M,N,A1),1+rempli);
affichage(polygone(O,P,B1),5+rempli);
affichage(polygone(Q,R,C1),6+rempli);
```

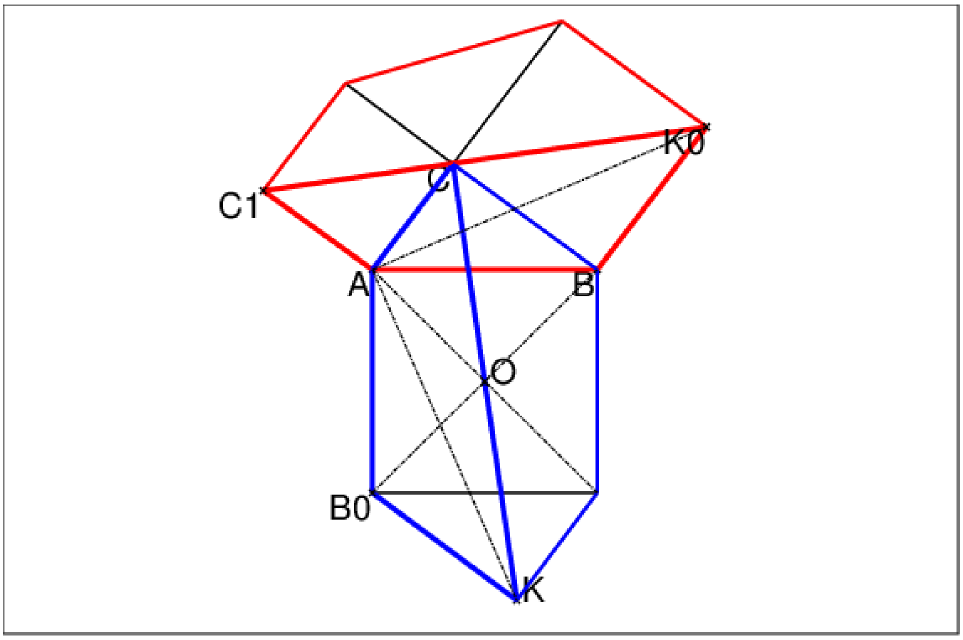
```
affichage(polygone(S,T,D1),67+rempli);
affichage(translation(5,polygone(J,D,A,M1)),4+rempli);
affichage(translation(5+(D-P1),polygone(J,M1,B,P1)),2+rempli);
affichage(translation(5+(M1-C),polygone(J,D,C)),3+rempli);
affichage(translation(5+(M1-C)+(D-P1),polygone(J,P1,C)),47+rempli);
affichage(translation(5+(D-P1)+(B-N),polygone(M,A1,N)),1+rempli);
affichage(translation(5+(B-P)+(M1-B)+(D-P1),polygone(O,B1,P)),5+rempli);
affichage(translation((A-N)+5+(A-D),polygone(Q,C1,R)),6+rempli);
affichage(translation(5+(B-O),polygone(S,D1,T)),67+rempli);
```

On obtient :



2.5.5 Le puzzle de Léonard de Vinci

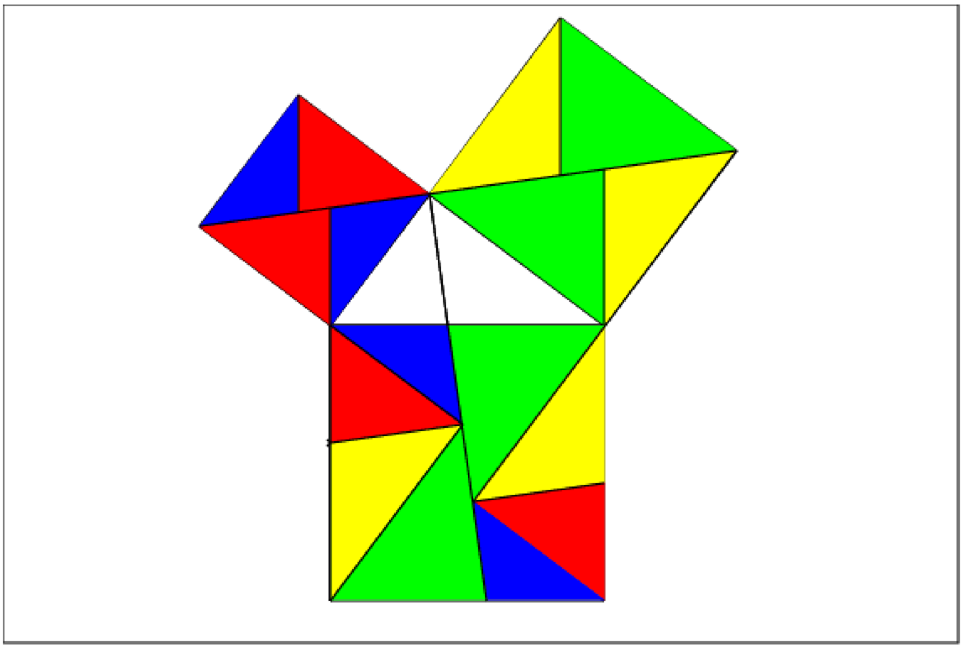
On trace la figure habituelle qui illustre le théorème de Pythagore :
le triangle rectangle en C T=CAB et à l'extérieur de T, les 3 carrés de côtés respectifs $AB = c$, $BC = a$ et $CA = b$.
Leonard de Vinci a l'idée de compléter cette figure en traçant 2 triangles égaux à T selon la figure ci-dessous.



On trace alors les 2 quadrilatères rouges et les 2 quadrilatères bleus. Les 2 quadrilatères rouges (resp bleus) sont égaux : les rouges sont symétriques par rapport à la droite C_1K_0 et les bleus sont symétriques par rapport au milieu O de BB_0.
La rotation de centre A et d'angle $\pi/2$ transforme le quadilatère bleu AB_0KC en le quadrilatère rouge ABK_0C_1 : les 2 quadrilatères bleu ont donc la même aire que les 2 quadrilatères rouge.
On a donc : 2*aire$(T) + a^2 + b^2$=2*aire$(T) + c^2$ donc $a^2 + b^2 = c^2$.

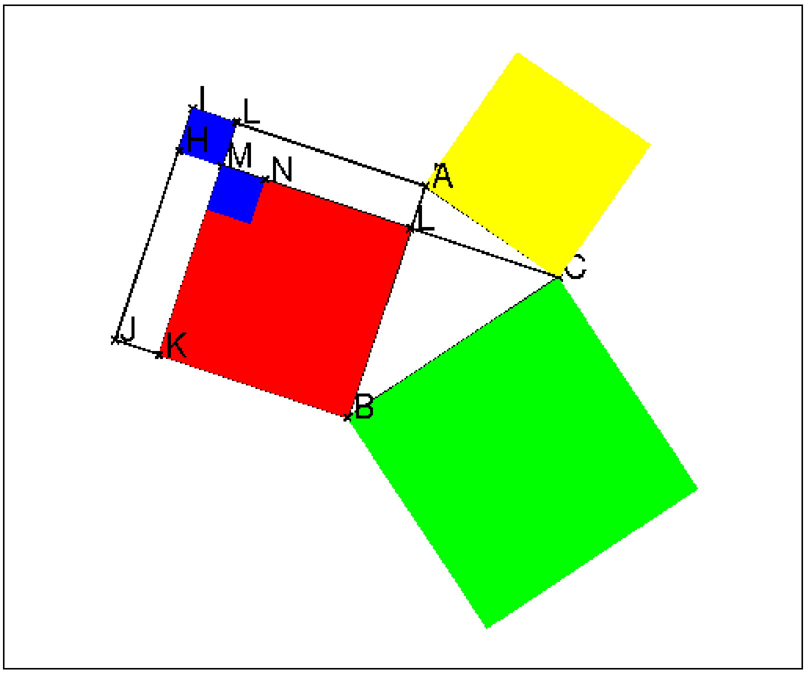
Le puzzle selon la démonstration de Léonard de Vinci

A l'intérieur du carré de côté c on trace deux triangles directement égaux à T. Ces deux triangles sont symétriques par rapport à O centre du carré de côté c.Puis on trace la bissectrice intérieure de l'angle C de T cette bissectrice passe par les sommets $S1$ et $S2$ de l'angle droit de ces 2 triangles (on remarquera que les triangles $CAS1$ et $CBS2$ sont des triangles rectangles isocèles).
A partir des sommets $S1$ et $S2$ trace des ségments parallèles à C_1C : cela détermine les triangles rouge et jaune. Puis on colore en vert et bleu les 2x2 triangles non colorés du carré de côté c.
On obtient :



2.6 La formule $(a-b)^2 = a^2 - 2ab + b^2$ comme exercice

Soit ABC est un triangle quelconque et L le pied de la hauteur issue de C. On pose $AB = c$, $AC = b$, $BC = a$, $CH = h$, $HB = d_1$, $HA = d_2$. Montrer que :
$h^2 = b^2 - d_2^2$,
$h^2 = a^2 - d_1^2$,
$d_2 = b\cos(A)$,
en déduire que : $a^2 = d_1^2 + h^2 = d_1^2 - d_2^2 + b^2$.
En observant la figure, montrer que :
la surface verte V = a^2
la surface jaune J= b^2
la surface rouge R=$d_1^2 - d_2^2$

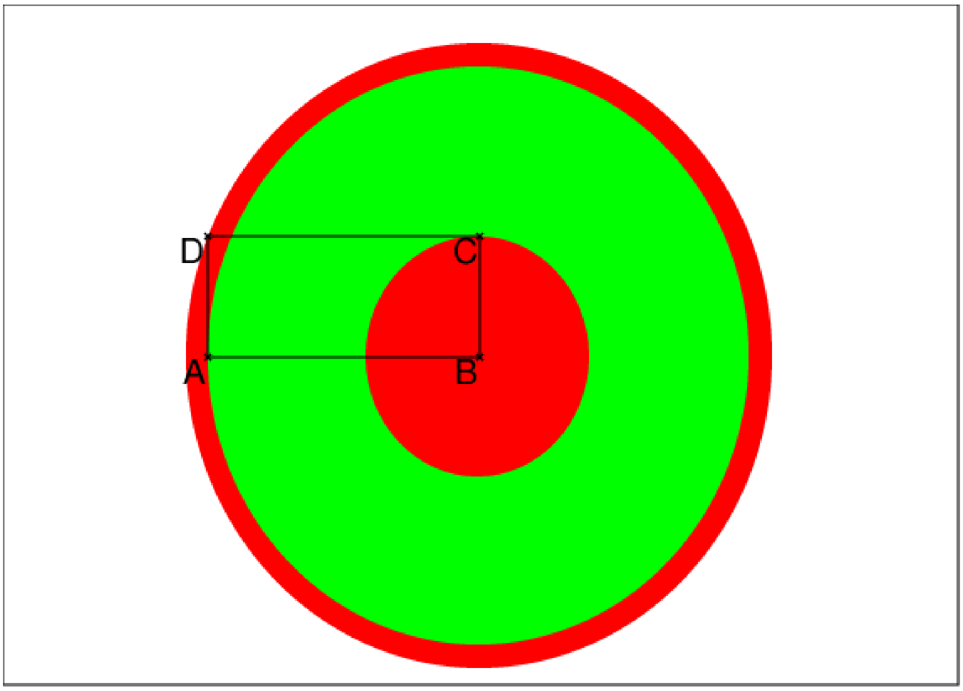


Que vaut la surface rouge R par rapport à c et d_2, par rapport à $b, c, \cos(A)$? En déduire la généralistion du théorème de Pythagore.
Solution On applique le théorème de Pythagore aux triangles rectangles ALC et BLC donc :
$h^2 = b^2 - d_2^2$ et
$h^2 = a^2 - d_1^2$,
On a donc : $V = R + J = a^2 = d_1^2 - d_2^2 + b^2$
Puisque $c = d_1 + d_2$ on a $d_1 - d_2 = c - 2d_2 = c - 2b\cos(A)$:
$R = d_1^2 - d_2^2 = c(c - 2d_2) = c(c - 2b\cos(A)) = c^2 - 2bc\cos(A)$
$c^2 = R + 2d_2^2 + 2d_1d_2 = R + 2c * d_2 = a^2 - b^2 + 2bc\cos(A)$
Donc
$a^2 = b^2 + c^2 - 2bc\cos(A)$

2.7 Exercice

Soit un rectangle $ABCD$.
On trace les trois cercles :

1. C_1 : cercle de centre B et de rayon BC,
2. C_2 : cercle de centre B et de rayon BA,
3. C_3 : cercle de centre B et de rayon BD.



Laquelle des deux zones en rouge a l'aire la plus grande ?

Solution

On pose $AB = a$, $BC = b$ $BD = d$.

L'aire de C_1 vaut : πb^2

L'aire de C_2 vaut : πa^2

L'aire de C_3 vaut : πd^2

On a d'après le théorème de Pythagore $d^2 = BD^2 = AB^2 + BC^2 = a^2 + b^2$

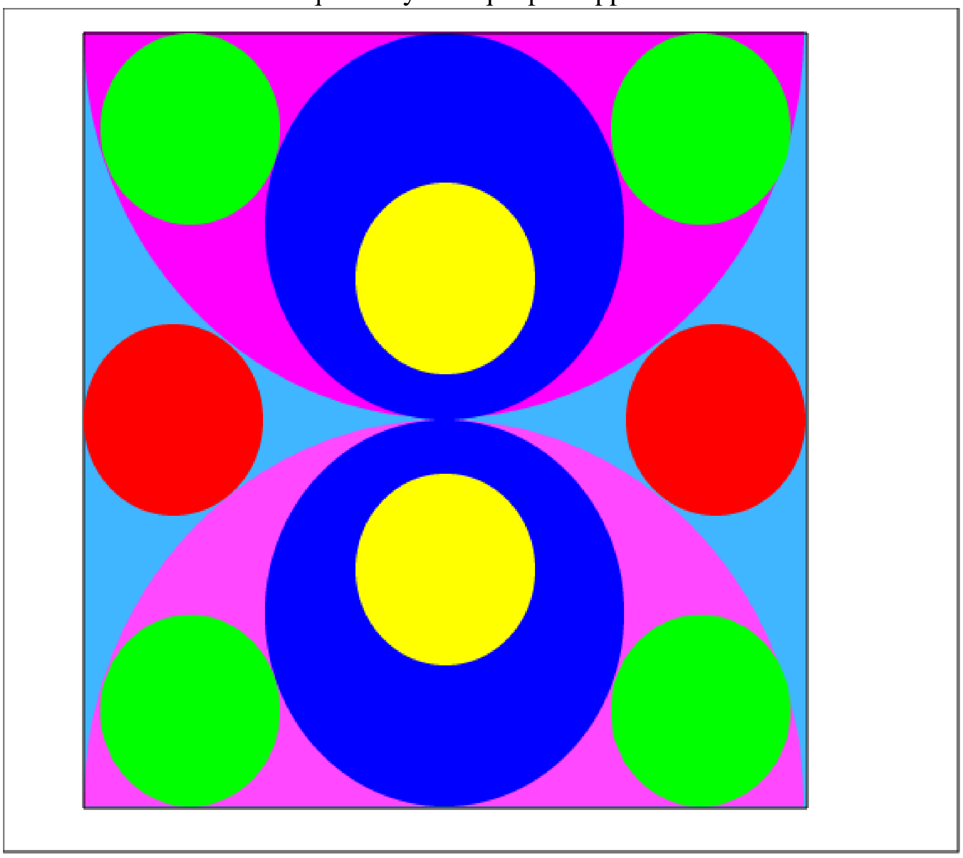
Donc l'aire de C_3 vaut : $\pi(a^2 + b^2)$

L'aire de l'anneau rouge vaut donc : $\pi(a^2 + b^2) - \pi a^2 = \pi b^2$

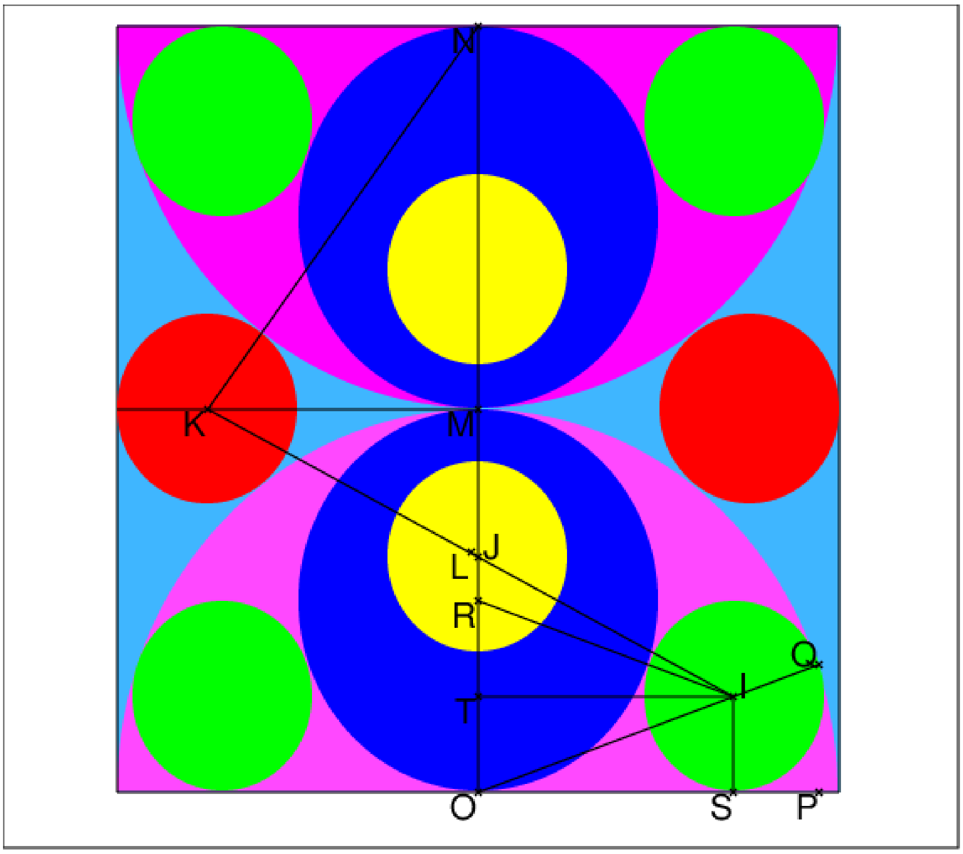
Donc les 2 zones en rouge ont même aire !

2.8 Exercice le carré de soie

Voici un carré de soie qui est symétrique par rapport à ses médianes :



Les cercles rouges et verts sont-ils egaux ?
Sachant que le cercle jaune est égale au cercle vert et que les centres I, J, K des cercles vert, jaune, rouges sont alignés, est ce que J est-il le milieu de IK ?
Solution



On note R le rayon des cercles bleus.
Les cercles magenta ont alors $2R$ comme rayon.
Cherchons le rayon r des cercles rouges.
On a :
$MN = 2R$, $NK = 2R + r$, $MK = 2R + r$ et d'après Pythagore $(2R + r)^2 = (2R - r)^2 + 4R^2$ donc $8Rr = 4R^2$ et $r = \frac{R}{2}$ Cherchons le rayon r_v des cercles verts.
Le cercle jaune est centré en J situé sur une médiane du carré Les cercles bleus ont comme rayon R On trace le rectangle $OSIT$, on a :
$RI = R + r_v$, $OQ = 2R$, $IS := r_v$, $OS = IT$ et $RT = R0 - 0T = R - r_v$ donc d'après Pythagore, $OS^2 = (2R - rv)^2 - r_v^2 = 4R^2 - 4Rr_v$ et
$OS^2 = IT^2 = IR^2 - RT^2 = (R + r_v)^2 - (R - r_v)^2 = 4Rr_v$
Donc :
$4R^2 - 4Rr_v = 4Rr_v$
$r_v = \frac{R}{2}$
On en déduit que $OS^2 = 2R$.
Cherchons les coordonnées de I de K et de J en prenant les bords du carré comme axe de coordonnées.
On a :
K est le point de coordonnées $R/2, 2R$
I est le point de coordonnées $R(2 + \sqrt{2}), R/2$
J est l'intersection de IK avec la droite $x = R$.
On tape :

```
I:=point(R*(2+sqrt(2) +i/2))
```

```
K:=point(R*(1/2+i*2))
equation(droite(I,K))
```

On obtient :

```
y=((6*sqrt(2)-9)*x+(13-6*sqrt(2))/2*R)
```

Donc les coordonnées de J sont : $2R, (-23+18*\sqrt{2})*R/2$ Les coordonnées du milieu L de IK sont :
On tape pour avoir les coordonnées du milieu L de IK :

```
evalc(factor((I+J)/2))
```

On obtient :
`(10+4*sqrt(2))*R/8+i*5*R/4` Les coordonnées du milieu L de IK sont : $R(10+4\sqrt{2})/8, 5*R/4$
On tape :

```
evalf((10+4*sqrt(2))/8)
```

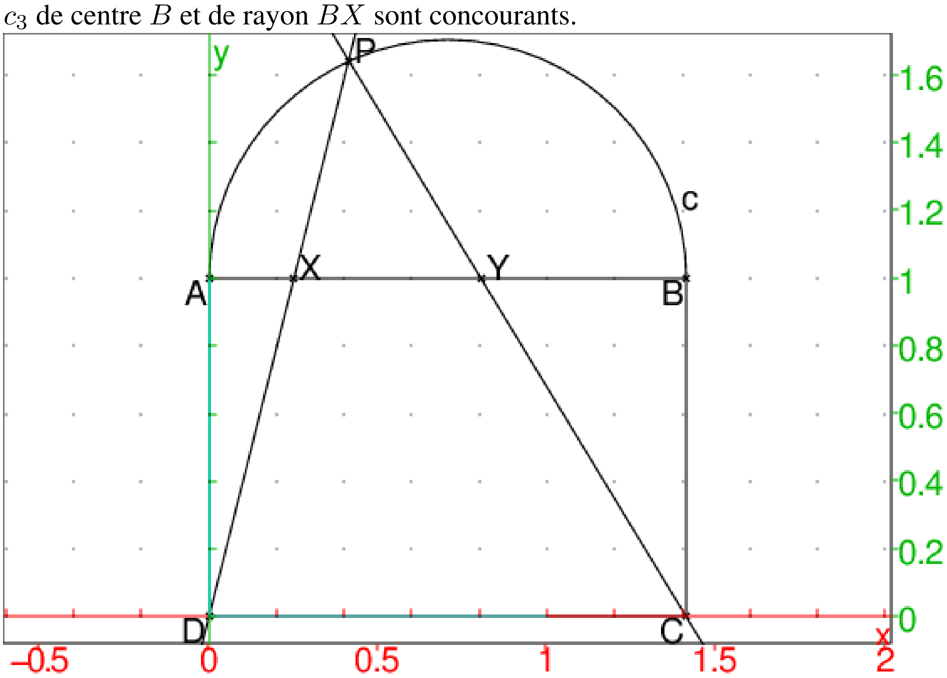
On obtient :

```
1.9571068
```

ce qui est proche de 2 donc L est proche de J
Donc J n'est pas le milieu de IK.

2.9 Exercice un problème posé par Fermat en 1658

Soit un rectangle indirect $ABCD$ tel que $AB=\sqrt{2}$ et $AD=1$.
On construit à l'extérieur du rectangle un demi-cercle c de diamètre AB.
Soit P un point sur c, et X (resp Y) l'intersection de PD (resp de PC) avec AB.
Monter que $AY^2+BX^2=2=AB^2$.
En déduire que les 3 cercles, c_1 de diamètre AB, c_2 de centre A et de rayon AY et c_3 de centre B et de rayon BX sont concourants.



Avec Xcas
On tape dans un niveau de géométrie :

```
A:=point(i);B:=point(sqrt(2)+i);
```

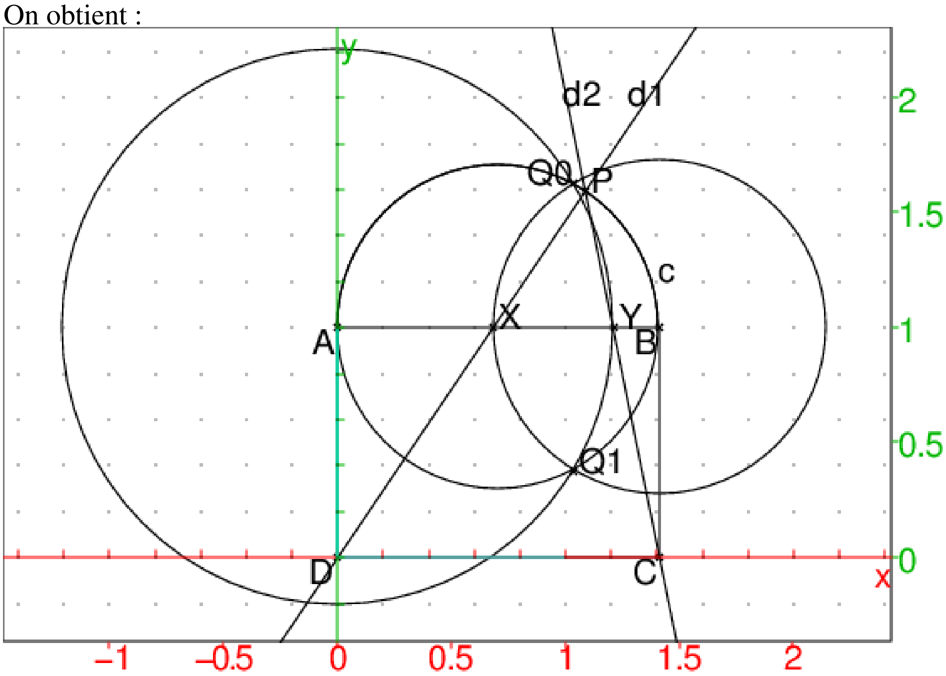
```
C:=point(sqrt(2));D:=point(0);
polygone(A,B,C,D);
c:=cercle(sqrt(2)/2+i,sqrt(2)/2,0,pi);
supposons(t=[2.0,0,0,0.1]);
P:=element(c,t);
d1:=droite(D,P);
d2:=droite(C,P);
X:=inter_unique(d1,droite(y=1));
Y:=inter_unique(d2,droite(y=1));
```

On tape :
`simplify(longueur2(A,Y)+longueur2(B,X))`
On obtient :
`2`
Soit Q_0, Q_1 les points d'intersection des cercles c_2 et c_3.
On a donc :
$AQ_0 = AQ_1 = AY$ (car Q_0, Q_1 sont des points de c_2).
$BQ_0 = BQ_1 = BX$ (car Q_0, Q_1 sont des points de c_3).
Donc :
$AQ_0^2 + BQ_0^2 = 2 = AB^2$ et $AQ_1^2 + BQ_1^2 = 2 = AB^2$

D'après la réciproque du théorème de Pythagore on en déduit que :
$\widehat{AQ_0B} = \widehat{AQ_1B} = pi/2$ donc
Q_0 et Q_1 sont sur le cercle de diamètre AB.
On tape dans le même niveau de géométrie :

```
c1:=cercle(A,B):;c1
Q:=inter(c1,cercle(A,longueur(A,Y))):;
Q0:=affichage(Q[0],quadrant2);Q1:=Q[1];
```

On obtient :



Sans Xcas

Soient $t = \widehat{BAP}$ et (x_1, y_1) les coordonnées de P.
On a $AP = \sqrt{2}\cos(t)$ donc :
$x_1 = AP\cos(t) = \sqrt{2}\cos(t)^2$
$y_1 = 1 + AP\sin(t) = 1 + \sqrt{2}\cos(t)\sin(t)$
Soient $u = \widehat{CDP} = \widehat{AXD}$ donc :
$\tan(u) = y_1/x_1 = 1/AX$ donc

$$AX = \frac{x_1}{y_1}$$

Soient $v = \widehat{PCD} = \widehat{BYC}$ donc :
$\tan(v) = y_1/(\sqrt{2} - x_1) = 1/BY$ donc

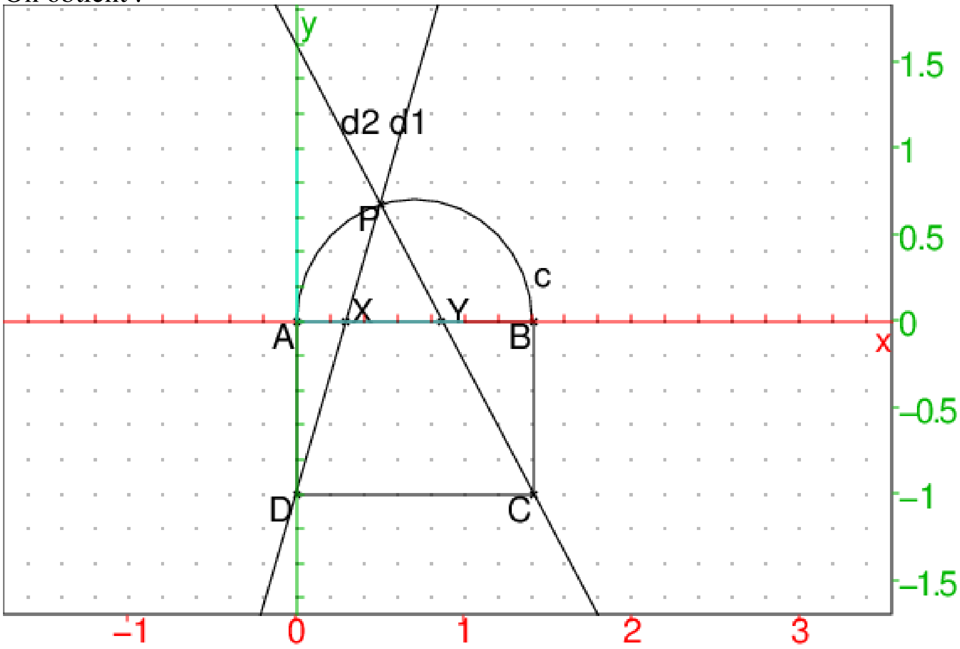
$$BY = \frac{\sqrt{2} - x_1}{y_1}$$

Donc :
$AY^2 = (\sqrt{2} - BX)^2 = (\sqrt{2} + x_1/y_1 - \sqrt{2}/y_1)^2$ et
$BX^2 = (\sqrt{2} - AX)^2 = (\sqrt{2} - x_1/y_1)^2$
$AY^2 + BX^2 = (\sqrt{2} - x_1/y_1)^2 + (\sqrt{2} + x_1/y_1)^2 + 2/y_1^2 - 2\sqrt{2}(\sqrt{2} + x_1/y_1)/y_1$
$AY^2 + BX^2 = 4 + 2(x_1^2 + 1 - 2y_1 - \sqrt{2}x_1)/y_1^2 = 4 + 2(N/D)$
On a en remplacant x_1 et y_1 en fonction de t :
$N = (x_1^2 - \sqrt{2}x_1 + 1 - 2y_1) =$
$N = 2\cos(t)^4 - 2\cos(t)^2 + 1 - 2 - 2\sqrt{2}\cos(t)\sin(t)$
$N = -2\cos(t)^2\sin(t)^2 - 1 - 2\sqrt{2}\cos(t)\sin(t)$
et
$D = y_1^2 = (1 + \sqrt{2}\cos(t)\sin(t))^2 =$
$D = 1 + 2\sqrt{2}\cos(t)\sin(t) + 2\cos(t)^2\sin(t)^2$ donc
$N/D = -1$
$AY^2 + BX^2 = 4 - 2 * 1 = 2$
ouf ! ! !

Ou plus simplement :
Soit le repère d'origine A et d'axe des x porté par $\overrightarrow{AB}$.
On a : $B = (\sqrt{2}, 0)$ et $D = (0 - 1)$
Soit (x_P, y_P) le coordonnées de P.
(x_P, y_P) vérifient léquation $x_P^2 + y_P^2 - \sqrt{2}x_P = 0$
On refait la figure on tape :

```
A:=point(0);B:=point(sqrt(2));
D:=point(-i);C:=point(sqrt(2)-i);
polygone(0,-i,-i+sqrt(2),sqrt(2));
supposons(xp=[1.0,0,1.4,0.1]);
P:=point(xp+i*sqrt(sqrt(2)*xp-xp^2));
c:=cercle(point(0),point(sqrt(2)),0,pi);
d1:=droite(D,P);
d2:=droite(C,P);
X:=inter_unique(d1,segment(A,B));
Y:=inter_unique(d2,segment(A,B));
```

On obtient :



L'équation de la droite PX est :

$$y = \frac{y_P + 1}{x_P}x - 1$$

donc l'abscisse de X est :

$$\frac{x_P}{y_P + 1}$$

$$BX = AB - AX = \sqrt{2} - \frac{x_P}{y_P + 1} = \frac{\sqrt{2} + \sqrt{2}y_P - x_P}{y_P + 1}$$

$$BX^2 = \frac{2 + 2y_p^2 + x_p^2 + 4y_P - 2\sqrt{2}x_P - 2\sqrt{2}y_P x_P}{(y_P + 1)^2}$$

L'équation de la droite PY est :

$$y = \frac{y_P + 1}{x_P - \sqrt{2}}(x - \sqrt{2}) - 1$$

donc l'abscisse de Y est :

$$\sqrt{2} + \frac{x_P - \sqrt{2}}{y_P + 1} = \frac{\sqrt{2}y_P + x_P}{y_P + 1}$$

$$AY^2 = \frac{2y_P^2 + x_P^2 + 2\sqrt{2}y_P x_P}{(y_P + 1)^2}$$

Donc :

$$AY^2 + BX^2 = \frac{4y_p^2 + 2x_p^2 - 2\sqrt{2}x_P + 4y_P + 2}{(y_P + 1)^2}$$

Puisque $x_P^2 + y_P^2 = \sqrt{2}x_P$ on a $4y_p^2 + 2x_p^2 = 2y_P^2 + 2\sqrt{2}x_P$ on a :

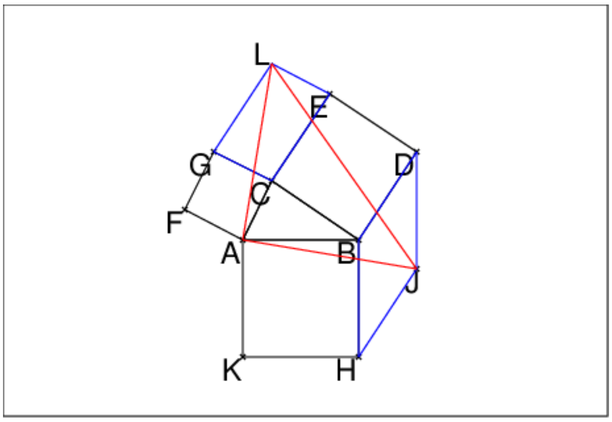
$$AY^2 + BX^2 = \frac{2y_p^2 + 4y_P + 2}{(y_P + 1)^2} = \frac{2(y_p^2 + 2y_P + 1)}{(y_P + 1)^2} = 2 = AB^2$$

Chapitre 3

Le théorème de 1968

3.1 Le théorème

Soit un triangle quelconque ABC. On construit sur les côtés du triangle ABC les carrés directs $CBDE$, $ACGF$ et $BAKH$, puis les parallélogrammes $DBHJ$ et $GCEL$.



Remarque : si ABC est direct les carrés sont à l'extérieur du triangle.
On a la propriété suivante : Le triangle AJL est isocèle rectangle direct.

3.2 La figure

Pour faire la figure, on tape les instructions suivantes qui se trouvent dans le fichier `th1968.xws` :

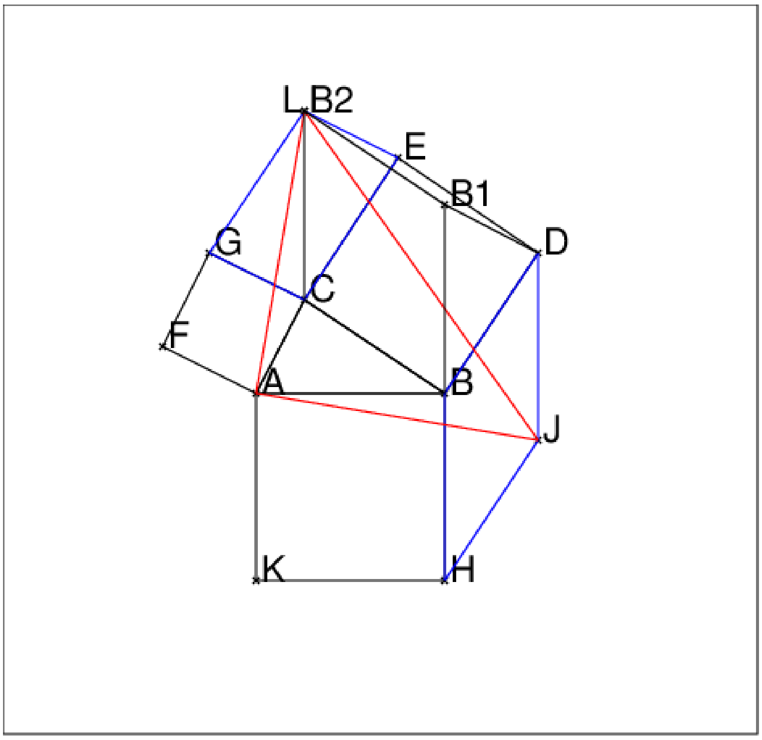
```
A:=point(-4,-1);
B:=point(-2,-1);
C:=point(-3.5,0);
triangle(A,B,C);
carre(B,A,K,H);
carre(C,B,D,E);
carre(A,C,G,F);
J:=H+D-B;
```

```
polygone(D,B,H,J,affichage=bleu);
L:=E+G-C:;legend(L,"L",quadrant2;
polygone(C,G,L,E,affichage=bleu);
triangle(A,J,L,affichage=rouge);
```

3.3 Une démonstration géométrique

On suppose que le triangle ABC est direct car la figure est plus lisible.
On fait des constructions supplémentaires et on tape :

```
B1:=translation(B-H,B);
B2:=translation(C-B,B1);
segment(B1,B);
segment(B1,L);
segment(B1,D);
segment(C,L);
```



On remarque que `B2` est confondu avec `L` puisque les triangles `B1BD` et `LCE` sont égaux au triangle `ABC` en effet :
- le triangle `B1BD` est le transformé du triangle `ABC` par rotation de centre `B` et d'angle `-pi/2`, et
- le triangle `LCE` est le transformé du triangle `ABC` par la composition de la rotation de centre `C` et d'angle `-pi/2` et de la translation de vecteur `E-C`.
La rotation de centre `A` et d'angle `pi/2` transforme `H` en `B1` et `HJ` en `B1B2`.
Donc la rotation de centre `A` et d'angle `pi/2` transforme `J` en `B2`.
Donc, puisque `B2` est confondu avec `L`, la rotation de centre `A` et d'angle `pi/2` transforme `J` en `L`
Donc le triangle `AJL` est isocèle rectangle direct.

3.4 Une démonstration avec les complexes

Soit `a=affixe(A), b=affixe(B), c=affixe(C)`
On a :
`j=affixe(J)=b+i*(a-b)-i*(c-b)=b+i(a-c)`
`l=affixe(L)=c+i*(b-c)-i*(a-c)=c+i(b-a)`
On a donc :

$$l-a = c-a+i*(b-a) = i^2*(a-c)+i*(b-a) = i*(b-a+i(a-c)) = i*(j-a)$$

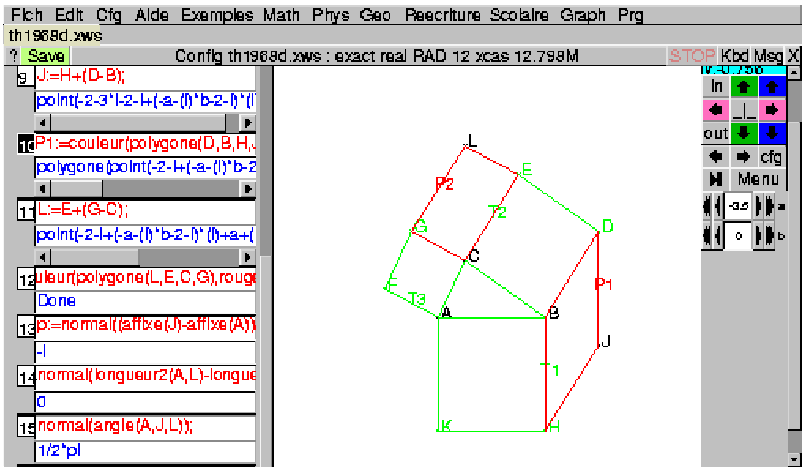
L'égalité `l-a=i*(j-a)` prouve que `L` se déduit de `J` par la rotation de centre `A` et d'angle `pi/2`.

3.5 La démonstration du théorème avec `Xcas`

On suppose que le point `A` est à l'origine du repère et que le point `B` est le point d'affixe 2. Le point `C` a comme affixe $a + ib$, avec a et b quelconques.
Pour faire la figure on suppose que $a = -1$ et que $b = -1$.
On tape les instructions suivantes qui se trouvent dans le fichier `th1968d.xws` :

```
assume(a=-3.5);
assume(b=0);
A:=point(-4,-1);
B:=point(-2,-1);
C:=point(a,b);
T1:=couleur(carre(B,A,K,H),vert);
T2:=couleur(carre(C,B,D,E),vert);
T3:=couleur(carre(A,C,G,F),vert);
J:=H+(D-B);
P1:=couleur(polygone(D,B,H,J),rouge);
L:=E+(G-C);
P2:=couleur(polygone(L,E,C,G),rouge);
p:=normal((affixe(J)-affixe(A))/(affixe(L)-affixe(A)));
normal(longueur2(A,L)-longueur2(A,J));
normal(angle(A,J,L));
```

On obtient `-i`, `0` et `pi/2` comme résultats des 3 dernières commandes :



Fich Edit Cfg Aide Exemples Math Phys Geo Reecriture Scolaire Graph Prg
th1968d.xws
? Save Config th1968d.xws : exact real RAD 12 xcas 12.798M STOP Kbd Msg X
J:=H+(D-B);
L:=E+(G-C);
Done
p:=normal((affixe(J)-affixe(A))
-i
0
normal(angle(A,J,L));
1/2*pi
L E D G C F A B J K H P1 P2 T1 T2 T3
in out cfg Menu

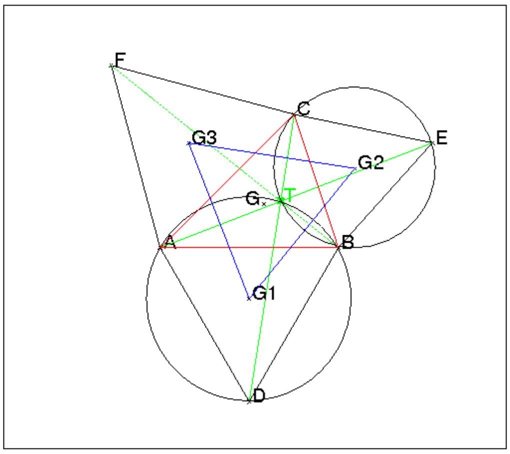
Chapitre 4

Le théorème de Napoléon

4.1 Le théorème

Soit un triangle quelconque ABC.
On construit à l'extérieur du triangle ABC les triangles équilatèraux BAD, CBE et ACF qui ont pour centre de gravité : $G1$, $G2$ et $G3$.
On a les propriétés suivantes :
Le triangle $G1G2G3$ est équilatéral et a même centre de gravité que le triangle ABC.
Les droites AE, DC, BF sont concourantes en un point T qui s'appelle le point de Torricelli.
Le point T est aussi le point de concours des cercles circonscrits aux triangles BAD, CBE et ACF.

4.2 La figure



Pour faire cette figure, on tape dans un éditeur de programme :

```
A:=point(-2.,-1.,'affichage'=0);
B:=point(2.,-1,'affichage'=0);
```

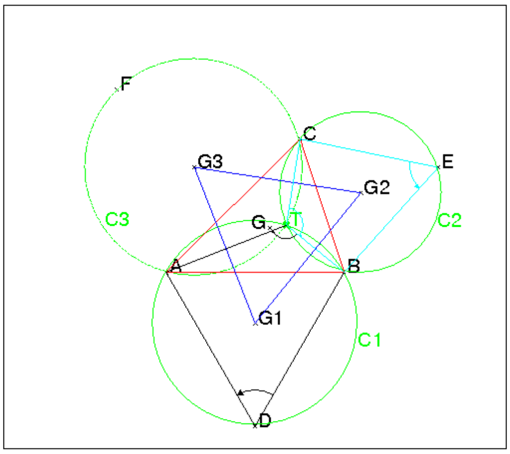
```
C:=point(1.,2,'affichage'=0);
triangle_equilateral(B,A,D);
triangle_equilateral(C,B,E);
triangle_equilateral(A,C,F);
segment(A,E,affichage=2);
segment(C,D,affichage=2);
segment(B,F,affichage=2+ligne_tiret);
T:=inter_droite(droite(A,E),droite(C,D),affichage=2+
                 epaisseur_point_2);
circonscrit(A,B,D);
circonscrit(C,B,E);
G1:=isobarycentre(A,B,D);
G2:=isobarycentre(C,B,E);
G3:=isobarycentre(A,C,F);
G:=isobarycentre(A,C,B,affichage=quadrant2);
triangle(A,B,C,affichage=1);
triangle(G1,G2,G3,affichage=4);
```

4.3 Les démonstrations géométriques

4.3.1 Avec les cercles circonscrits à BAD, CBE et ACF

On trace les cercles C_1, C_2 et C_3 circonscrits aux triangles ADB, BEC et CFA de centres respectifs G_1, G_2 et G_3.
Pour faire la figure, on tape :

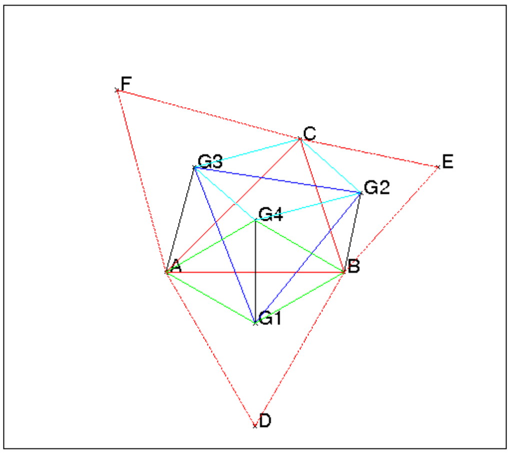
```
A:=point(-2.,-1.,'affichage'=0);
B:=point(2.,-1,'affichage'=0);
C:=point(1.,2,'affichage'=0);
D:=rotation(B,pi/3,A);
E:=rotation(C,pi/3,B);
F:=rotation(A,pi/3,C);
C1:=circonscrit(A,B,D,affichage=2);
C2:=circonscrit(C,B,E,affichage=2);
C3:=circonscrit(C,A,F,affichage=2+ligne_tiret);
G1:=isobarycentre(A,B,D);
G2:=isobarycentre(C,B,E);
G3:=isobarycentre(A,C,F);
G:=isobarycentre(A,C,B,affichage=quadrant2);
triangle(A,B,C,affichage=1);
triangle(G1,G2,G3,affichage=4);
T:=inter(C1,C2,affichage=2+epaisseur_point_2)[1];
angle(D,B,A,"");
angle(T,B,A,"");
angle(T,C,B,"");
angle(E,C,B,"");
```



Montrons que ces trois cercles sont concourants en un point T.
Soit T le point d'intersection de C_1 et C_2, on a :
$\widehat{BDA} = \frac{\pi}{3}$.
Comme l'angle $\widehat{BTA}$ intercepte le même arc BA que l'angle $\widehat{BDA}$ de C_1 on a :
$\widehat{BTA} = \frac{\pi}{3}$ ou $\widehat{BTA} = \frac{2\pi}{3}$ de même :
$\widehat{BTC} = \frac{\pi}{3}$ ou $\widehat{BTC} = \frac{2\pi}{3}$ donc :
soit $\widehat{CTA} = \widehat{BTC} + \widehat{BTA}$ si TB est entre TC et TA
soit $\widehat{CTA} = |\widehat{BTC} - \widehat{BTA}|$ sinon.
Donc $\widehat{CTA} = \frac{\pi}{3}$ ou $\widehat{CTA} = \frac{2\pi}{3}$ et donc $CAFT$ sont cocycliques.
On a donc montré que T se trouve sur $C3$.
Comme G_1G_2 (resp G_1G_3 ou G_2G_3) est perpendiculaire à BT (resp à AT ou à CT) et que $\widehat{G_2G_1G_3} + \widehat{G_1G_2G_3} + \widehat{G_1G_3G_2} = \pi$ on a :
$\widehat{G_2G_1G_3} = \widehat{G_1G_2G_3} = \widehat{G_1G_3G_2} = \frac{\pi}{3}$
ce qui prouve que le triangle $G_1G_2G_3$ est équilatèral.

4.3.2 Avec G_4 le symétrique de G_1 par rapport à AB

Soit G_4 le symétrique de G_1 par rapport à AB.



Pour faire cette figure, on tape dans un éditeur de programme :

```
A:=point(-2.,-1.,'affichage'=0);
B:=point(2.,-1,'affichage'=0);
C:=point(1.,2,'affichage'=0);
D:=rotation(B,pi/3.,A);
E:=rotation(C,pi/3.,B);
F:=rotation(A,pi/3.,C);
G1:=isobarycentre(A,B,D);
G2:=isobarycentre(C,B,E);
G3:=isobarycentre(A,C,F);
G4:=symetrie(droite(A,B),G1);
G:=isobarycentre(A,C,B,affichage=quadrant2);
triangle(A,B,C,affichage=1);
triangle(G1,G2,G3,affichage=4);
segment(A,G1);
segment(A,G3);
segment(G4,G3);
segment(G4,G2);
segment(C,G2);
segment(C,G3);
segment(B,G4);
segment(A,G4);
segment(B,G2);
segment(B,G1);
segment(G4,G1);
segment(A,F,affichage=1+ligne_tiret);
segment(C,F,affichage=1+ligne_tiret);
segment(C,E,affichage=1+ligne_tiret);
segment(E,B,affichage=1+ligne_tiret);
segment(B,D,affichage=1+ligne_tiret);
segment(A,D,affichage=1+ligne_tiret);
```

Les triangles AG_4G_3 et G_4BG_2 sont égaux et sont des triangles semblables à ABC en effet :
le quadrilatère AG_1BG_4 est un losange d'angle $A = \frac{\pi}{3}$ (4 cotés égaux à la diagonale G_1G_4), donc
l'angle $\widehat{G_4AG_3}$ est égale à l'angle $\widehat{BAC}$,
$AG_4 = AG_1 = BG_1 = BG_4 = \frac{AB}{\sqrt{3}}$,
$AG_3 = CG_3 = \frac{AC}{\sqrt{3}}$.
Le triangle AG_4G_3 est donc semblable au triangle ABC avec comme rapport de similitude $\frac{1}{\sqrt{(3)}}$.
De même l'angle $\widehat{G_4BG_2}$ est égale à l'angle $\widehat{ABC}$ et,
$BG_2 = CG_2 = \dfrac{BC}{\sqrt{3}}$
donc le triangle G_4BG_2 est semblable au triangle ABC avec comme rapport de similitude $\frac{1}{\sqrt{3}}$.
On en déduit que :
$BG_2 = G_4G_3$ et $AG_2 = G_4G_2$ et donc que le quadrilatère $G_2CG_3G_4$ est un parallélogramme.
Les triangles $G_1G_4G_3$ et G_1BG_2 sont donc égaux (l'angle $\widehat{G_3G_4G_1} = \widehat{CBA} + pi/3 = \widehat{G_2BG_1}$) et ces deux triangles se déduisent l'un de l'autre par une rotaion de centre G_1 et d'anble $\frac{\pi}{3}$ donc
$G_1G_3 = G_1G_2$ et l'angle $\widehat{G_2G_1G_3} = \frac{\pi}{3}$.
L'isobarycentre de A, B, C est aussi l'isobarycentre de G_1, G_4, C car le quadrilatère AG_1BG_4 est un losange.
L'isobarycentre de G_1, G_4, C est aussi l'isobarycentre de G_1, G_2, G_3 car le quadrilatère $CG_3G_4G_2$ est un parallélogramme.
Donc ABC et $G_1G_2G_3$ ont même centre de gravitè.

4.3.3 Avec les symétriques de G_1 (resp de G_2) par rapport à AB (resp à BE)

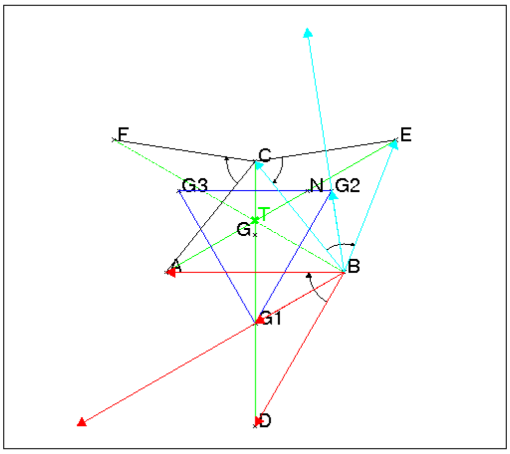
L'énoncé

Soit ABC un triangle quelconque. On construit à l'extérieur du triangle ABC les triangles équilatèraux : ABD, BCE et ACF.
1/ Montrer que $AE = BF = CD$.
2/ Montrer que AE, BF et CD sont concourantes.
3/ Théorème de Napoléon :
On note G_1, G_2 et G_3 les centres de gravité des triangles ABD, BCE et ACF.
Montrer que le triangle $G_1G_2G_3$ est équilatèral.

La solution

```
A:=point(-2.,-1.,'affichage'=0);
B:=point(2.,-1,'affichage'=0);
C:=point(0.,1.5,'affichage'=0);
D:=rotation(B,pi/3.,A);
```

```
E:=rotation(C,pi/3.,B);
F:=rotation(A,pi/3.,C);
segment(A,E,affichage=2);
segment(C,D,affichage=2);
segment(B,F,affichage=2+ligne_tiret);
T:=inter_droite(droite(A,E),droite(C,D),affichage=2+
                 epaisseur_point_2);
G1:=isobarycentre(A,B,D);
G2:=isobarycentre(C,B,E);
G3:=isobarycentre(A,C,F);
G:=isobarycentre(A,C,B,affichage=quadrant2);
triangle(A,B,C,affichage=1);
triangle(G1,G2,G3,affichage=4);
angle(B,D,A,"");
angle(B,C,E,"");
angle(C,E,B,"");
angle(C,A,F,"");
vecteur(B,G1,affichage=1)
vecteur(B,A,affichage=1);
vecteur(B,D,affichage=1);
vecteur(B,B+3*(G1-B),affichage=1);
vecteur(B,G2,affichage=6)
vecteur(B,E,affichage=6);
vecteur(B,C,affichage=6);
vecteur(B,B+3*(G2-B),affichage=6);
N:=rotation(C,pi/3.,T);
```



On suppose le triangle ABC direct.
1/ La rotation de centre B et d'angle $-\pi/3$ transforme :
D en A et C en E donc :
$DC = AE$ et $(\overrightarrow{DC}, \overrightarrow{AE}) = -\pi/3$.
De même la rotation de centre C et d'angle $-\pi/3$ transforme :

E en B et A en F donc :
$AE = FB$ et $(\overrightarrow{EA}, \overrightarrow{BF}) = -\pi/3$.
On montre, en utilisant les vecteurs, que $G_1G_2G_3$ est équilatèral, on a :
$\overrightarrow{BG_1} = \frac{1}{3}(\overrightarrow{BD} + \overrightarrow{BA})$ et $\overrightarrow{BG_2} = \frac{1}{3}(\overrightarrow{BE} + \overrightarrow{BC})$
$\overrightarrow{CG_2} = \frac{1}{3}(\overrightarrow{CE} + \overrightarrow{CB})$ et $\overrightarrow{CG_3} = \frac{1}{3}(\overrightarrow{CF} + \overrightarrow{CA})$
donc
$\overrightarrow{G_2G_1} = \frac{1}{3}(\overrightarrow{EA} + \overrightarrow{CD})$ et
$\overrightarrow{G_2G_3} = \frac{1}{3}(\overrightarrow{BF} + \overrightarrow{EA})$
$\overrightarrow{BF}$ (resp $\overrightarrow{EA}$) est le transformé de $\overrightarrow{EA}$ (resp $\overrightarrow{CD}$) par une rotation d'angle $-\pi/3$ donc $\overrightarrow{G_2G_3}$ est le transformé de $\overrightarrow{G_2G_1}$ par une rotation d'angle $-\pi/3$ donc le triangle $G_1G_2G_3$ est équilatèral.
2/ Soit T le point d'intersection de AE et de CD. D'après la première question :
$(\overrightarrow{TC}, \overrightarrow{TE}) = (\overrightarrow{DC}, \overrightarrow{AE}) = -\pi/3$.
On construit alors le point N sur AE pour que le triangle TCN soit équlatéral et donc $(\overrightarrow{CN}, \overrightarrow{CT}) = -\pi/3$. Ainsi, la rotation de centre C et d'angle $-\pi/3$ transforme AE en FB et le point N de AE en le point T de BF donc BF passe par T.
3/ Autre démonstration de $G_1G_2G_3$ est équilatèral. On construit :
G_4 le symétrique de G_1 par rapport à AB,
G_5 le symétrique de G_2 par rapport à BE,
donc les triangles AG_1G_4, BG_1G_4, BG_2G_5 sont équilatéraux.
La rotation de centre B et d'angle $-\pi/3$ transforme :
G_1 en G_4 et G_2 en G_5 donc $G_1G_2 = G_4G_5$ et $(\overrightarrow{G_2G_1}, \overrightarrow{G_5G_4}) = -\pi/3$.
On va montrer que le quadrilatère $G_3G_2G_5G_4$ est un parallélogramme et on aura ainsi montrer que le triangle $G_1G_2G_3$ est équilatèral puisque :
$G_3G_2 = G_4G_5 = G_1G_2$ et $(\overrightarrow{G_2G_3}, \overrightarrow{G_5G_4}) = 0$ donc $(\overrightarrow{G_2G_3}, \overrightarrow{G_2G_1}) = \pi/3$.
Les triangles AG_4G_3 et G_4BG_2 sont semblables au triangle ABC (même angle A (resp B) et deux cotés proportionnels) et comme $AG_4 = G_4B$, les triangles AG_4G_3 et G_4BG_2 sont égaux. Donc $G_3G_4 = BG_2 = BG_5 = G_2G_5$.
On a :
$(\overrightarrow{G_4B}, \overrightarrow{G_4G_3}) = (\overrightarrow{G_4A}, \overrightarrow{G_4G_3}) + 2\pi/3 = (\overrightarrow{BG_4}, \overrightarrow{BG_2}) + 2\pi/3 =$
$(\overrightarrow{BG_4}, \overrightarrow{BE}) + 2\pi/3 - \pi/6 = (\overrightarrow{BG_4}, \overrightarrow{BE}) + \pi/2$.
Donc G_3G_4 est paralléle à G_2G_5 puisque ces deux droites sont perpendiculaires à BE. On a ainsi montrer que le quadrilatère $G_3G_2G_5G_4$ est un parallélogramme, ce qui termine la démonstration.

4.3.4 Avec les nombres complexes

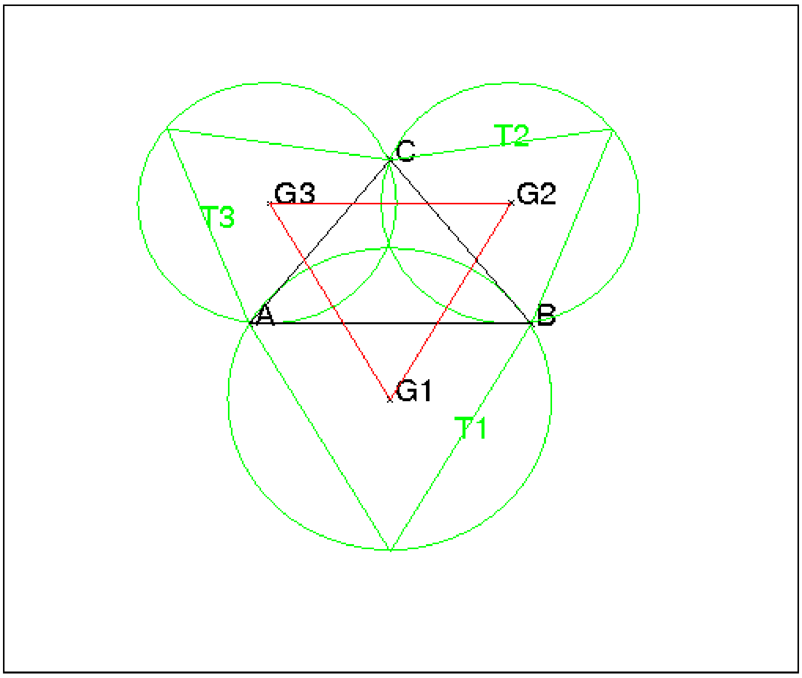
Soient a, b, c, g_1, g_2, g_2 les affixes de A, B, C, G_1, G_2, G_3 on a :
si le triangle ADB est de sens direct, A se déduit de B dans la rotation de centre g_1 et d'angle $2 * \pi/3$ donc puisque $j = e^{\frac{i2\pi}{3}}$:
$a - g_1 = j * (b - g_1)$
de même si les triangles BEC et CFA sont de sens direct on a :
$b - g_2 = j * (c - g_2)$
$c - g_3 = j * (a - g_3)$
On en déduit que :
$(1 - j)g_1 = a - jb$ et
$(1 - j)g_2 = b - jc$ et

$(1-j)g_3 = c - ja$ donc
$(1-j)(g_1-g_2) = a - b(1+j) + jc = a + jc + j^2b$ puisque $1+j+j^2 = 0$ et
$(1-j)(g_2-g_3) = b - c(1+j) + ja = j(a + jc + j^2b)$ puisque $j^3 = 1$ donc
$(1-j)(g_2-g_3) = j(1-j)(g_1-g_2)$ et après division par $j-1$ on a
$g_2 - g_3 = j(g_1 - g_2)$ et
cette égalité prouve que le triangle $G_1G_2G_3$ est équilatèral.
On a de plus :
$(1-j)g_1 + (1-j)g_2 + (1-j)g_3 = a - jb + b - jc + c - ja = (1-j)(a+b+c)$ donc
$(a+b+c)/3 = (g_1+g_2+g_3)/3$ ce qui veut dire que les triangles ABC et $G_1G_2G_3$ ont même centre de gravité.

4.4 La démonstration du théorème avec `Xcas`

On suppose que le point `A` est à l'origine du repère et que le point `B` est le point d'affixe 2. Le point `C` a comme affixe $a+ib$, avec a et b quelconques.
Pour faire la figure on suppose que $a = -1$ et que $b = -1$.
On tape les instructions suivantes qui se trouvent dans le fichier `napoleon` :

```
assume(a=0);
assume(b=1.5);
A:=point(-2,-1);
B:=point(2,-2);
C:=point(a,b);
T1:=couleur(triangle_equilateral(B,A),vert);
T2:=couleur(triangle_equilateral(C,B),vert);
T3:=couleur(triangle_equilateral(A,C),vert);
couleur(circonscrit(T1),vert);
couleur(circonscrit(T2),vert);
couleur(circonscrit(T3),vert);
AB:=segment(A,B);
AC:=segment(A,C);
CB:=segment(C,B);
G1:=normal(isobarycentre(T1));
G2:=normal(isobarycentre(T2));
G3:=normal(isobarycentre(T3));
G1G2:=couleur(segment(G1,G2),rouge);
G2G3:=couleur(segment(G2,G3),rouge);
G3G1:=couleur(segment(G3,G1),rouge);
normal(longueur2(G1,G2)-longueur2(G2,G3));
normal(longueur2(G1,G3)-longueur2(G3,G2));
```



On obtient la figure et comme réponses :
`0` à `normal(longueur2(G1,G2)-longueur2(G2,G3));`,
`0` à `normal(longueur2(G1,G2)-longueur2(G1,G3));`,
ce qui prouve que le triangle $G1G2G3$ est équilatèral.

Chapitre 5

Des exercices sur les transformations

5.1 Exercice 1 sur les rotations

5.1.1 L'énoncé et sa figure

Soit un triangle direct quelconque ABC. On construit à l'extérieur de ABC les carrés $ACDE$ et $AGHB$ et le parallélogramme $AEFG$.
Montrer que $BE = CG$ et que CG est perpendiculaire à EB
Montrer que $AF = BC$ et que AF est perpendiculaire à BC
Pour faire la figure, on définit le triangle direct ABC avec les 3 points A, B, C obtenus en cliquant en mode `point`, puis on tape :

```
carre(A,C,D,E);
carre(B,A,G,H);
F:=translation(G-A,E);
parallelogramme(A,E,F);
segment(B,E,affichage=1);
segment(C,G,affichage=1);
segment(B,C,affichage=2);
segment(A,F,affichage=2);
K:=projection(droite(B,C),A):;
segment(A,K,affichage=4+ligne_tiret_point);
```

On obtient :

5.1.2 La démonstration géométrique

La rotation de centre A et d'angle $\frac{\pi}{2}$ transforme C en E et G en B. Donc $CG = EB$ et CG est perpendiculaire à EB
Soit I le symétrique de G par rapport à A :

On a donc :
$\overrightarrow{IA} = \overrightarrow{AG} = \overrightarrow{EF}$ donc le quadrilatère $AIEF$ est un parallélogramme donc :

$\overrightarrow{AF} = \overrightarrow{IE}$ La rotation de centre A et d'angle $\frac{\pi}{2}$ transforme B en I donc $BC = IE$ et BC est perpendiculaire à IE et comme $\overrightarrow{AF} = \overrightarrow{IE}$, on en déduit que : $BC = AF$ et BC est perpendiculaire à AF.

5.1.3 La démonstration avec les nombres complexes

On choisit un repère d'origine A.
La rotation de centre A et d'angle $\frac{\pi}{2}$ transforme C en E et G en B. Donc si $zb, zc, ...$ désigne les affixes de $B, C...$ on a :
$ze = i * zc$ et $zb = i * zg$ donc $zg = -i * zb$
$\overrightarrow{AF} = \overrightarrow{AE} + \overrightarrow{AG}$ donc
$zf = ze + zg = i(zc - zb)$ donc $AF = BC$ et BC est perpendiculaire à AF.

5.1.4 La démonstration avec `Xcas`

On choisit un repère d'origine A et C sur l'axe des y.
On tape :

```
assume(a=[-2,-5,5,0.1]);
assume(b=[1,-5,5,0.1]);
assume(c=[-5/2,-5,5,0.1]);
A:=point([0,0,'affichage'=0]);
B:=point([a,b,'affichage'=0]);
C:=point([0,c,'affichage'=0]);
carre(A,C,D,E);
carre(B,A,G,H);
F:=translation(G-A,E);
parallelogramme(A,E,F);
```

Puis on demande à `Xcas` :
`longueur2(C,G)`, on obtient `(-b)^2+(c+a)^2`
`longueur2(B,E)`, on obtient `(a+c)^2+(b-c+c)^2`
`longueur2(A,F)`, on obtient `(c-b)^2+a^2`
`longueur2(B,C)`, on obtient `a^2+(b-c)^2`
`equation(droite(C,G))`, on obtient `y=((-a-c)*1/b*x+c)`
`equation(droite(B,E))`, on obtient `y=(b*1/(a+c)*x+(b*c)/(a+c))`
le produit des pentes vaut bien -1.
`equation(droite(A,F))`, on obtient `y=((-a)*1/(b-c)*x)`
`equation(droite(B,C))`, on obtient `y=((b-c)*1/a*x+c)`
le produit des pentes vaut bien -1.
On peut aussi utiliser directement :
`normal(pente(droite(C,G))*pente(droite(B,E)))` et
`normal(pente(droite(A,F))*pente(droite(B,C)))` et obtenir `-1`
ou
`est_perpendiculaire(droite(C,G),droite(B,E))` et
`est_perpendiculaire(droite(A,F),droite(B,C))` et obtenir `1`

5.2 Exercice 2 sur les similitudes

5.2.1 L'énoncé et sa figure

Soit un carré direct $ABCD$ de centre I et de côté a. Soit J le milieu de IB.
On construit le carré direct $IJKL$.
Trouver le centre, le rapport et l'angle de la similitude qui transforme :
A en I,
B en J
C en K et
D en L.
On tape :

```
A:=point(0);
assume(a=[5,0,10,0.1]);
B:=point(a);
c:=carre(A,B,C,D):;c
I:=milieu(A,C);
J:=milieu(B,I);
carre(I,J,K,L);
segment(A,C);
segment(B,D);
```

On obtient :

5.2.2 La démonstration géométrique

Si cette similitude existe on a :
le rapport de la similitude est $k = a*\sqrt{2}/4$.puisque $IJ = a*\sqrt{2}/4 = AB*\sqrt{2}/4$,
L'angle est de $-\frac{\pi}{4}$ puisque $(\overrightarrow{AB}, \overrightarrow{IJ}) = -\frac{\pi}{4}$

Cherchons son centre O.
On doit avoir $(\overrightarrow{OA}, \overrightarrow{OI}) = -\frac{\pi}{4}$ donc O se trouve sur l'arc ABI du cercle de diamètre AB (arc capable interceptant AI et d'angle $-\frac{\pi}{4}$ puisque l'angle au centre de ce cercle qui intercepte l'arc AI est droit).
On doit avoir $(\overrightarrow{OB}, \overrightarrow{OJ}) = -\frac{\pi}{4}$ donc O se trouve sur l'arc BKJ du cercle de diamètre BK (arc capable interceptant BJ et d'angle $-\frac{\pi}{4}$ puisque l'angle au centre de ce cercle qui intercepte l'arc BJ est droit).
Soit O l'intersection de ces 2 arcs : O est donc la projection de B sur AK (puisque l'angle $(\overrightarrow{OA}, \overrightarrow{OB}) = \frac{\pi}{2}$ et l'angle $(\overrightarrow{OB}, \overrightarrow{OK}) = \frac{\pi}{2}$, on a $(\overrightarrow{OA}, \overrightarrow{OK}) = \pi$).
La similitude de centre O, d'angle $-\frac{\pi}{4}$ et de rapport $k = a * \sqrt{2}/4$ transforme A en I et B en J. Comme une similitude transforme des longueurs égales en des longueurs égales et conserve aussi les angles, cette similitude transforme le carré direct $ABCD$ en le carré direct $IJKL$.
On peut donc en déduire que les 4 cercles :
`cercle(A,B),cercle(B,K),circonscrit(I,K,C),circonscrit(ADL)`
sont concourants en O.
En effet :
$(\overrightarrow{IC}, \overrightarrow{IK}) = -\frac{\pi}{4}$ donc I et O sont sur l'arc capable interceptant CK et d'angle $-\frac{\pi}{4}$. On en déduit que O est sur le cercle circonscrit à IKC.
De même $(\overrightarrow{AD}, \overrightarrow{AL}) = -\frac{\pi}{4}$ donc A et O sont sur l'arc capable interceptant CK et d'angle $-\frac{\pi}{4}$. On en déduit que O est sur le cercle circonscrit à JAD.

5.2.3 La démonstration avec les nombres complexes et `Xcas`

On choisit A à l'origine et B sur l'axe des x.
Si cette similitude existe on a :
le rapport de la similitude est $k = a*\sqrt{2}/4$.puisque $IJ = a*\sqrt{2}/4 = AB*\sqrt{2}/4$,
L'angle est de $-\frac{\pi}{4}$ puisque $(\overrightarrow{AB}, \overrightarrow{IJ}) = -\frac{\pi}{4}$
Cherchons O le centre de la similitude d'affixe $z0$, on a :
$-z0*\sqrt{2}/4*exp(-i*\pi/4) = a*(1+i)/2 - z0$ On tape :

```
normal(solve(-z0*sqrt( 2)/4*exp(-i*pi/4)=a*(1+i)/2-z0,z0))
```

On obtient :

```
[(4+2*i)/5*a]
```

donc $z0 = (4+2*i)/5*a$ On tape :

```
I:=similitude(point((4+2*i)/5*a),sqrt(2)/4,-pi/4,A,
affichage=1+epaisseur_point_2)
J:=similitude(point(4+2*i),sqrt( 2)/4,-pi/4,B,
affichage=1+epaisseur_point_2)
K:=similitude(point(4+2*i),sqrt( 2)/4,-pi/4,C,
affichage=1+epaisseur_point_2)
L:=similitude(point(4+2*i),sqrt( 2)/4,-pi/4,D,
affichage=1+epaisseur_point_2)
```

On obtient :

5.3 Exercice 3 : lieu et similitude

5.3.1 L'énoncé et sa figure

Soit un triangle direct OAB rectangle en O avec $OA = a$ et $OB = b$.
Soit $D = At$ une demi droite variable telle que :

$(\overrightarrow{OA}, \overrightarrow{At}) = c, 0 \leq c \leq \frac{\pi}{2}$.
Soient $A1$ et $B1$ les projections respectives de A et B sur D.
Quelle est la valeur de c pour laquelle $A1$ et $B1$ sont confondus en un point que l'on nommera P ? Trouver les lieux de $A1$ et de $B1$ quand c varie.
Montrer que le triangle $PA1B1$ reste semblable au triangle OAB quand c varie.
Trouver le lieu de M milieu de $A1B1$ quand c varie.
On tape (on choisit $a = 3$ et $b = 5$) : (les instructions `carre` sont la pour voir les droites perpendiculaires)

```
assume(a=3);
assume(b=5);
O:=point(0);
A:=point(a);
B:=point(b*i);
d:=segment(A,B);
supposons(c=[0.8,0,1.57,0.1]);
D:=droite(y=(tan(c)*x));
A1:=projection(D,A);
B1:=projection(D,B);
P:=projection(d,O);
carre(P,O+(P-0)*0.9);
segment(O,P,affichage=3+point_tiret);
carre(P,O+(P-0)*0.9);
carre(O+(P-0)*0.93,P);
segment(A,A1,affichage=3+point_tiret);
carre(A1,O+(A1-0)*0.9);
segment(B,B1,affichage=3+point_tiret);
carre(B1,O+(B1-0)*1.05);
triangle(P,A1,B1);
M:=milieu(A1,B1);
C1:=cercle(1.5,1.5,0,pi);
C2:=cercle(2.5*i,2.5,-pi/2,pi/2);
C3:=similitude(P,sqrt(a^2+b^2)/(2*b),atan(a/b),C2,affichage=5)
```

On obtient :

5.3.2 La démonstration géométrique

$A1$ et $B1$ sont confondus lorsque D est perpendiculaire à AB. P est donc la projection de O sur AB.
Quand c varie, l'angle $(\overrightarrow{A1O}, \overrightarrow{A1A}) = \frac{\pi}{2}$ donc le lieu de $A1$ est le demi-cercle $C1$ de diamètre OA contenant P.
Quand c varie l'angle $(\overrightarrow{B1B}, \overrightarrow{B1O}) = \frac{\pi}{2}$ donc le lieu de $B1$ est le demi-cercle $C2$ de diamètre OB contenant P.
Ces deux demi-cercles se coupent en P.
Les triangles POA et PBO sont semblables au triangle OBA.
Si $B1$ est sur l'arc OP de $C2$ alors A est sur l'arc AP de $C1$.
alors l'angle Si $B1$ est sur l'arc PB de $C2$ alors A est sur l'arc OP de $C1$.
Dans les 2 cas l'angle $B1$ de $PA1B1$ est égal à l'angle B de OAB et $A1$ de $PA1B1$ est égal à l'angle A de OAB.
Donc le triangle $PB1A1$ est direct et est semblable au triangle OBA.
On en déduit que $A1$ se déduit de $B1$ par une similitude de centre P, d'angle $\pi/2$ et de rapport $PA1/PB1 = OA/OB = a/b$. $C1$ est donc l'image de $C2$ par cette similitude.
Le milieu M de $A1B1$ se déduit de $B1$ par une similitude de centre P, d'angle :
$B1$ (égal à l'angle B de OAB valant $\text{atan}(a/b)$) et de rapport :
$PM/PB1 = AB/(2 * OB) = \sqrt{a^2 + b^2}/(2b)$ (car la médiane OK de OAB a pour longueur $AB/2 = \sqrt{a^2 + b^2}/2$).
Donc le lieu de M est le demi cercle $C3$ qui se déduit de $C2$ par cette similitude : c'est le demi-cercle diamètre le segment joingnant les milieux de OA et OB (car si $c = 0$ $A1$ est en A et $B1$ est en O et si $c = \pi/2$ $A1$ est en O et $B1$ est en B) qui passe par P (puisque quand $A1$ et $B1$ sont en P, Mmilieu de $A1B1$ est aussi en P).

5.3.3 La démonstration avec les nombres complexes et Xcas

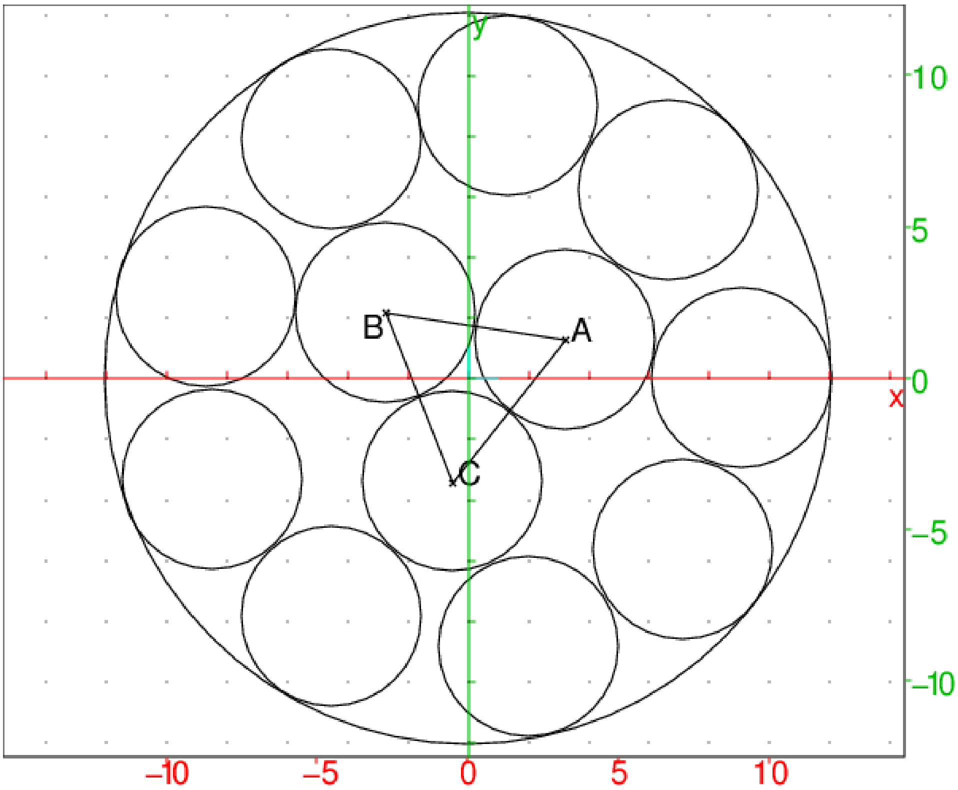
P est la projection de O sur AB.
On $PO * AB = a * b = PO * \sqrt{a^2 + b^2}$ donc $O - P = (B - A) * (iab/(a^2 + b^2))$
si p est l'affixe de P on a :
$p = (a - ib)(iab/(a^2 + b^2)) = (ab^2 + ia^2 b)/(a^2 + b^2)$
$A1$ a pour affixe $a1 = a\cos(c)\exp(ic)$
$B1$ a pour affixe $b1 = b\sin(c)\exp(ic)$
M a pour affixe $m = (a1 + b1)/2 = (a\cos(c) + b\sin(c))\exp(ic)/2$
On pose :
$x1 = re(m) = (a\cos(c) + b\sin(c))\cos(c)/2$ et
$y1 = im(m) = (a\cos(c) + b\sin(c))\sin(c)/2$
On a $x1 \geq 0$ et $y1 \geq 0$ car $c \in [0; \pi/2]$.
On tape :
```
x1,y1:=op(normal(coordonnees(M)))
factor(equation(plotparam(x1+i*y1,c)))
```
On obtient :
```
(b^2+a^2)*(2*x^2-x*a-b*y+2*y^2)
```
qui est l'équation du cercle de centre d'affixe $(a + ib)/2$ passant par O.
Comme $A1$ et $B1$ se trouve dans l'angle $(\overrightarrow{OA}, \overrightarrow{OB})$ c'est le demi-cercle se trouvant dans cet angle qui est le lieu de M.
ou bien, on calcule :
$x1^2 + y1^2 = (a\cos(c) + b\sin(c))^2/4 =$
$a/4 * \cos(c)(a\cos(c) + +b\sin(c)) + b/4\sin(c)(a\cos(c) + +b\sin(c)) =$
$a * x1/2 + b * y1/2$
Donc le lieu de M a pour équation pour $x \geq 0$ et $y \geq 0$:
$x^2 + y^2 - ax/2 - by/2 = (x - a/4)^2 + (y - b/4)^2 - (a^2 + b^2)/16$
qui est l'équation du demi-cercle $C3$.

5.4 Exercice : les verres et le plateau

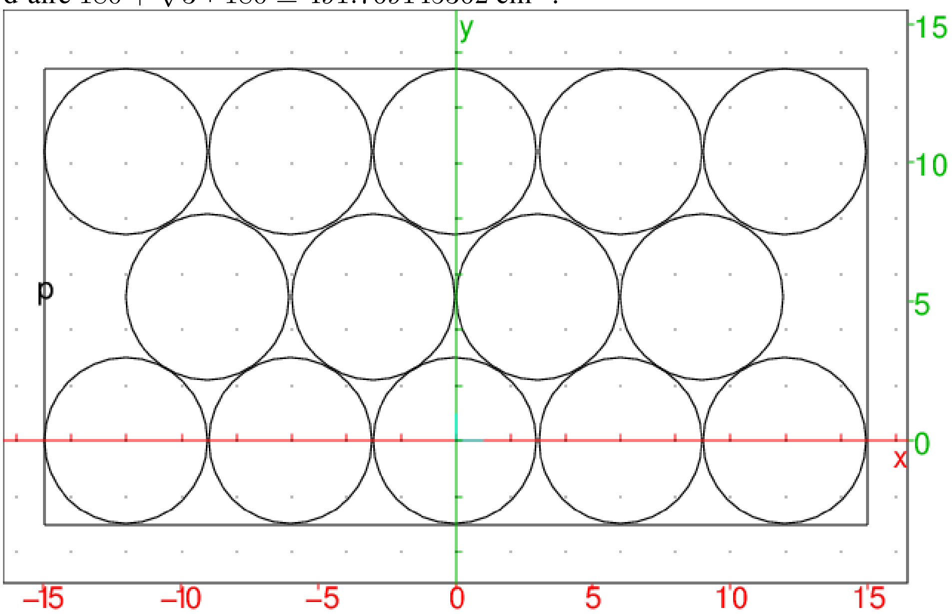
On dispose d'un plateau circulaire de 24.2 cm. Montrer à l'aide de `Xcas` que l'on peut poser sur ce plateau 12 verres cylindriques d'un diamètre de 6cm (sans les empiler !). **Solution** On tape :

```
supposons(d=[12.09,0,13,0.01]);
cercle(0,d);
cercle(d-3,3);
a:=2*asin(3/(d-3)); b:=2*pi/3-3*a;;
cercle((d-3)*exp(-i*a),3);
cercle((d-3)*exp(i*(a+b)),3);
cercle((d-3)*exp(i*(b+2*a)),3);
cercle((d-3)*exp(i*(b+3*a)),3);
cercle((d-3)*exp(i*(2*b+4*a)),3);
cercle((d-3)*exp(-2*i*a),3);
cercle((d-3)*exp(-3*i*a-i*b),3);
cercle((d-3)*exp(-4*i*a-i*b),3);
d1:=mediatrice((d-3),(d-3)*exp(i*(a+b))):;;
d2:=mediatrice((d-3)*exp(i*b+3*i*a),(d-3)*exp(2*i*b+4*i*a)):;;
d3:=mediatrice((d-3)*exp(-i*b-3*i*a),(d-3)*exp(-2*i*a)):;;
A:=inter_unique(d1,cercle((d-3),6),0);
cercle(A,3);
C:=inter_unique(d3,cercle((d-3)*exp(-2*i*a),6),0);
cercle(C,3);
triangle_equilateral(C,A,B);
cercle(B,3);
evalf(longueur(A,C))
```

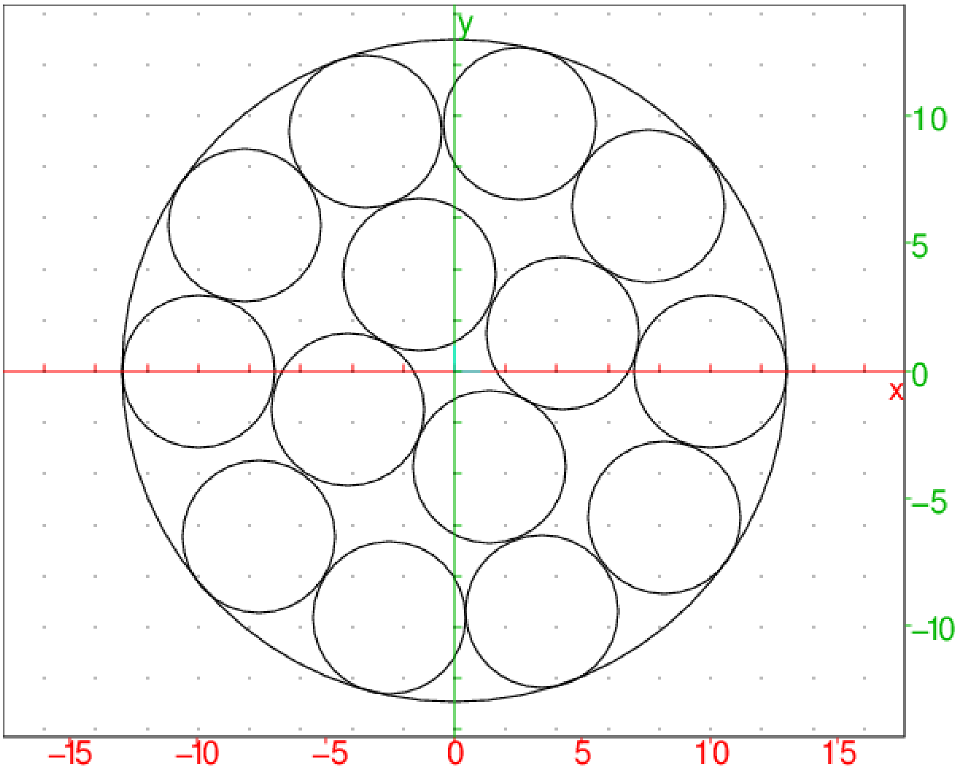
Pour `d=12.09`, on obtient comme longueur AC :
`6.00279577807`
et comme dessin :



Prolongement Quelles sont les dimensions minimum que doit avoir un plateau rectangulaire pour mettre 14 cylindrique verres de dimètre 6cm ? A surface égale faut-il choisir un plateau rectangulaire ou circulaire pour mettre 14 verres de diamètre 6 cm ? On obtient pour un plateau rectangulaire dimension 30 cm X $6(1+\sqrt{3})$ cm d'aire $180+\sqrt{3}*180 \simeq 491.769145362$ cm^2 :



On obtient pour un plateau circulaire de rayon 13 cm d'aire 529.93 cm^2 (le dessin a été fait en utilisant le mode `Pointeur` pour placer 3 cercles situés au centre) :


y
10
5
0
x
–5
–10
–15
–10
–5
0
5
10
15

Chapitre 6

Le théorème de Morley

6.1 Le théorème

Soit un triangle ABC et ses trissectrices intérieures.
Soient P, Q, R les points de ces trissectrices tels que le triangle ABP (respectivement BCQ, CAR) ait comme angles $\frac{\widehat{A}}{3}$ et $\frac{\widehat{B}}{3}$ (respectivement $\frac{\widehat{B}}{3}$ et $\frac{\widehat{C}}{3}$, $\frac{\widehat{C}}{3}$ et $\frac{\widehat{A}}{3}$), alors le triangle PQR est équilatéral.
De plus si U, V, W sont les points de ces trissectrices tels que le triangle ABU (respectivement BCV, CAW) ait comme angles $\frac{\widehat{2A}}{3}$ et $\frac{\widehat{2B}}{3}$ (respectivement $\frac{\widehat{2B}}{3}$ et $\frac{\widehat{2C}}{3}$, $\frac{\widehat{2C}}{3}$ et $\frac{\widehat{2A}}{3}$), alors les triangles URQ, VRP, WPQ sont isocèles.
On remarquera que le triangle isocèle URQ a comme angles :
$\widehat{U} = \pi - \frac{\widehat{2A}}{3} - \frac{\widehat{2B}}{3}$
$\widehat{R} = \widehat{Q} = \frac{\widehat{A}}{3} + \frac{\widehat{B}}{3}$.

6.2 La figure

On prend comme paramètres `a1` et `b1` qui représentent le tiers des angles `A` et `B` du triangle `ABC`.
Quitte à faire une similitude, on peut choisir `A` à l'origine du repère et `B` au point d'affixe 1, le point `C` a alors comme affixe :

$$\frac{\tan(3*b1)}{\tan(3*a1)+\tan(3*b1)*(1+i*\tan(3*a1))}$$

C est donc sur la droite passant par A de pente $\tan(3*a1)$ et sur la droite passant par B et de pente $-tan(3*b1)$.
On tape :

```
A:= point(-4.95-1.777*i);
B:= point(3.786-2.876*i);
C:= point(0.722+1.9*i);
a1:=(angle(A,B,C))/3;
b1:=(angle(B,C,A))/3;
c1:=eval(pi/3-a1-b1);
P:=inter(rotation(A,a1,droite(A,B)),rotation(B,-b1,
                                    droite(B,A)))[0];
```

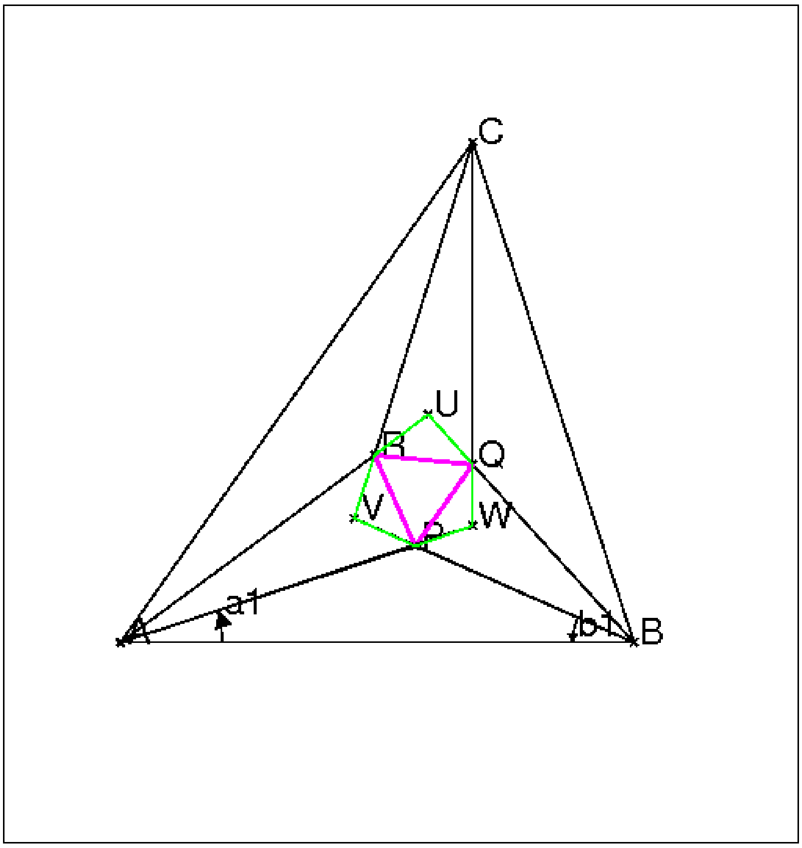
```
Q:=inter(rotation(C,-c1,droite(C,B)),rotation(B,b1,
                                        droite(B,C)))[0];
R:=inter(rotation(A,-a1,droite(A,C)),rotation(C,c1,
                                        droite(C,A)))[0];
U:=inter(rotation(A,2*a1,droite(A,B)),rotation(B,-2*b1,
                                        droite(B,A)))[0];
V:=inter(rotation(C,-2*c1,droite(C,B)),rotation(B,2*b1,
                                        droite(B,C)))[0];
W:=inter(rotation(A,-2*a1,droite(A,C)),rotation(C,2*c1,
                                        droite(C,A)))[0];
triangle(A,R,C);
triangle(B,Q,C);
triangle(A,P,B);
triangle(P,Q,R);
triangle(U,Q,R);
triangle(V,P,R);
triangle(W,Q,P);
```

Remarque On peut aussi utiliser la définition d'une droite par un point et sa pente pour définir les points `C` et `P` ce qui nous dispense faire les calculls des coordonnées de `C` et `P`.
On peut aussi taper :

```
A:=point(0);
B:=point(1);
a1:=0.32;
b1:=0.42;
TA1:=droite(A,pente=tan(a1));
TA2:=droite(A,pente=tan(2a1));
TA3:=droite(A,pente=tan(3a1)):;
TB1:=droite(B,pente=-tan(2b1)):;
TB2:=droite(B,pente=-tan(b1)):;
TB3:=droite(B,pente=-tan(3b1)):;
C:=inter_unique(TA3,TB3);
TC1:=droite(C,pente=tan(2a1-b1+pi/3)):;
TC2:=droite(C,pente=tan(a1-2b1+2pi/3)):;
TC3:=droite(C,pente=tan(3a1+pi/3)):;
P:=inter_unique(TA1,TB2);
Q:=inter_unique(TB1,TC2);
R:=inter_unique(TC1,TA2);
triangle(A,R,C);
triangle(B,Q,C);
triangle(A,P,B);
triangle(P,Q,R,couleur=magenta+line_width_2);
pq2:=longueur2(P,Q);
pr2:=longueur2(P,R);
qr2:=longueur2(Q,R);
U:=inter_unique(TB1,TA2);
V:=inter_unique(TB2,TC1);
```

```
W:=inter_unique(TC2,TA1)
normal(longueur2(W,Q)-longueur2(W,P));
normal(longueur2(U,Q)-longueur2(U,R));
normal(longueur2(V,P)-longueur2(V,R));
angle(A,B,W,"a1");
angle(B,V,A,"b1");
polygone(W,Q,U,R,V,P,affichage=vert);
```

On obtient la figure du triangle de Morley :



et 0.0379352631434 comme valeur des carrés des 3 longueurs et des valeurs de l'ordre de 10^{-14} pour les différence de longueurs.
Mais cela n'est pas une démonstration....

6.3 La démonstration du théorème avec Xcas

On place A à l'origine, B en (1,0) et on utilise 2 paramètres a1 et b1, tels que les angles en A et B soient 3*a1, 3*b1 C est donc sur la droite passant par A de pente tan(3*a1) et sur la droite passant par B et de pente -tan(3*b1), etc.

Pour faire une démonstration avec `Xcas`, on va remplacer : `a1:=0.32;` par `assume(a1=0.32);` et
`b1:=0.42;` par `assume(b1=0.42);`
La figure reste la même mais les calculs sont faits avec les paramètres formels `a1` et `b1` :

```
assume(a1=0.32);
assume(b1=0.42);
A:=point(0);
B:=point(1);
TA1:=droite(A,pente=tan(a1));
TB1:=droite(A,pente=texpand(tan(2a1)));
TA3:=droite(A,pente=texpand(tan(3a1))):;
TB1:=droite(B,pente=-texpand(tan(2b1))):;
TB2:=droite(B,pente=-tan(b1)):;
TB3:=droite(B,pente=-texpand(tan(3b1))):;
C:=inter_unique(TA3,TB3);
TC1:=droite(C,pente=texpand(tan(2a1-b1+pi/3))):;
TC2:=droite(C,pente=texpand(tan(a1-2b1+2pi/3))):;
TC3:=droite(C,pente=texpand(tan(3a1+pi/3))):;
P:=inter_unique(TA1,TB2);
Q:=inter_unique(TB1,TC2);
R:=inter_unique(TC1,TA2);
triangle(A,R,C);
triangle(B,Q,C);
triangle(A,P,B);
triangle(P,Q,R,couleur=magenta+line_width_2);
pq2:=longueur2(P,Q);
pr2:=longueur2(P,R);
qr2:=longueur2(Q,R);
U:=inter_unique(TB1,TA2);
V:=inter_unique(TB2,TC1);
W:=inter_unique(TC2,TA1)
angle(A,B,W,"a1");
angle(B,V,A,"b1");
polygone(W,Q,U,R,V,P,affichage=vert);
normal(pq2-pr2),normal(pq2-qr2);
normal(longueur2(W,Q)-longueur2(W,P));
normal(longueur2(U,Q)-longueur2(U,R));
normal(longueur2(V,P)-longueur2(V,R));
```

Ou on tape dans un éditeur de programme

```
assume(a1=0.32);
assume(b1=0.42);
A:=point(0);
B:=point(1);
C:=point(texpand(tan(b1*3)/(tan(a1*3)+
                 tan(b1*3))*(1+i*tan(a1*3))));
```

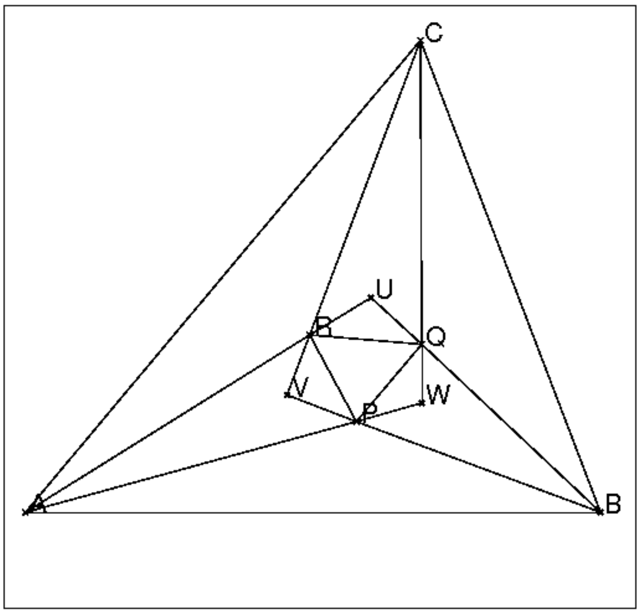
```
P:=point(texpand(tan(b1)/(tan(a1)+
                 tan(b1))*(1+i*tan(a1))));
R:=(inter(droite(0,1+i*texpand(tan(2*a1))),
      droite(C,C+1+i*texpand(tan(pi/3+2*a1-b1)))))[0];
Q:=(inter(droite(1,i*texpand(tan(2*b1))),
      droite(C,C+1+i*texpand(tan(2*pi/3+a1-2*b1)))))[0];
U:=point(texpand(tan(2*b1)/(tan(2*a1)+
                 tan(2*b1))*(1+i*tan(2*a1))));
W:=(inter(droite(0,1+i*texpand(tan(a1))),
      droite(C,C+1+i*texpand(tan(2*pi/3+a1-2*b1)))))[0];
V:=(inter(droite(1,i*texpand(tan(b1))),
       droite(C,C+1+i*texpand(tan(pi/3+2*a1-b1)))))[0];
triangle(A,R,C);
triangle(B,Q,C);
triangle(A,P,B);
triangle(P,Q,R);
triangle(P,Q,W);
triangle(P,V,R);
triangle(U,Q,R);
pq2:=longueur2(P,Q);
pr2:=longueur2(P,R);
qr2:=longueur2(Q,R);
ur2:=longueur2(U,R);
uq2:=longueur2(U,Q);
vr2:=longueur2(V,R);
vp2:=longueur2(V,P);
pw2:=longueur2(P,W);
qw2:=longueur2(Q,W);
normal(pq2-pr2),normal(pq2-qr2);
normal(uq2-ur2);
normal(vp2-vr2);
normal(pw2-qw2);
```

On valide par `OK`

On obtient alors :
`0,0,0,0,0`
ce qui prouve que le triangle `PQR` est équilatéral et que les triangles `UQR`, `VPR` et `WPQ` sont isocèles.

6.3.1 Une démonstration géométrique

`La figure`



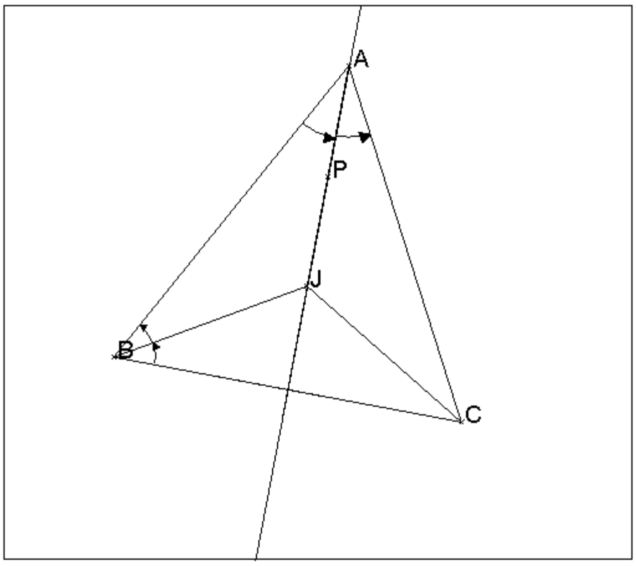
La démonstration

On suppose que les angles du triangle ABC valent $3a, 3b, 3c$ ($a + b + c = \pi/3$).
On trace les trissectrices des angles B et C.
Soit D l'intersection des deux trissectrices tels que le triangle BCD aient comme angles b et c et $\pi - b - c$.
Soient F et H tels que le triangle HFD soit isocéle d'angle à la base $a+c$ et H sur CD et F sur la 2-ième trissectrice de B : on construit F sur la 2-ième trissectrice de B tel que l'angle $\widehat{FDB} = \pi/3 + c = a + c + b + c$,
Soient E et K tels que le triangle KED soit isocéle d'angle à la base $a+b$ et K sur BD et E sur la 2-ième trissectrice de C : on construit F sur la 2-ième trissectrice de C tel que l'angle $\widehat{CDK} = \pi/3 + b = a + b + b + c$)
On a :
$\widehat{HDB} = b + c = \widehat{KDC}$
$\widehat{FDE} + (a + c) + (b + c) + (\pi - b - c) + (a + b) + (b + c) = 2\pi$ donc,
$\widehat{FDE} = \pi - 2a - 2b - 2c = \pi/3$ et
$\widehat{BFD} = \pi - b - (a + c) - (b + c) = a + \pi/3$
$\widehat{CED} = \pi - c - (a + b) - (b + c) = a + \pi/3$
D est sur les bissectrices des angles $\widehat{CBF}$ et $\widehat{BCE}$, donc D est équidistant de BF et CE et puisque $\widehat{BFD} = \widehat{CED}$ on a $DE = DF$
donc le triangle DEF est équilatèral ($DE = DF$ et $\widehat{FDE} = \pi/3$).
Il reste donc à montrer que F et E sont sur les trissectrices de A : on va montrer que FH et EK sont les trissectrices de A.
FK est la bissectrice de $\widehat{DKE}$
FB est la bissectrice de $\widehat{DBA}$ donc
si on suppose que KE coupe AB an L, LF est la bissectrice de $\widehat{L}$ et $\widehat{BLF} = \widehat{FLK} = a$ ($\widehat{BKL} = \pi - 2(a + b)$ $\widehat{KBL} = 2b$ donc $\widehat{BLK} = 2a$) et comme

$\widehat{BFH} = a + b$, L, F, H sont alignés.
De même si on suppose que HF coupe AC en J, JE est la bissectrice de $\widehat{J}$ et $\widehat{CJE} = \widehat{EJH} = a$ et J, E, K sont alignés.
On a K, E, L sont alignés et J, E, K sont alignés.
On a L, F, H sont alignés et F, H, J sont alignés
donc J et L sont à l'intersection de EK et FH. Mais J est sur AB et L sur AC, donc J et L sont confondus en A et on en déduit que FH et EK sont les trissectices de l'angle A.

6.3.2 La démonstration d'un Lemme

Les instructions pour faire cette figure se trouve dans le fichier `morley1.xws`.

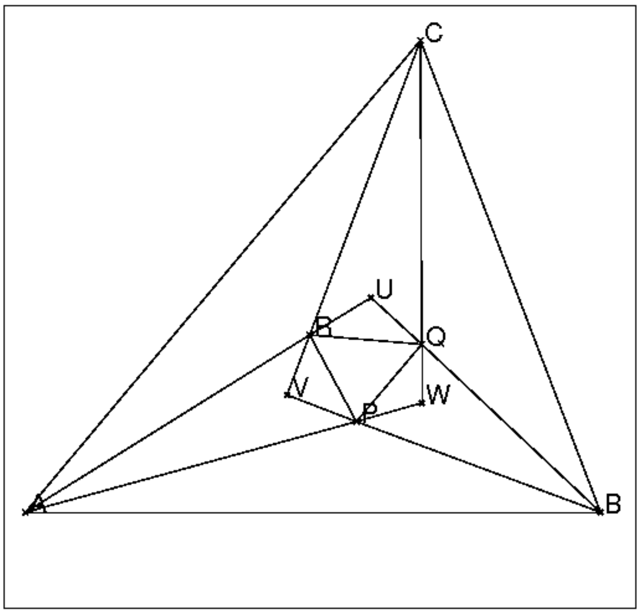


Lemme : Soient un triangle ABC, J le point de concours de ses bissectrices intérieures et P un point de la bissectrice intérieure de l'angle A se trouvant à l'intérieur du triangle ABC alors :
J et P sont confondus si et seulement si $\widehat{BPC} = \dfrac{(\pi + \widehat{A})}{2}$.

En effet :
- si J et P sont confondus on a bien : $\widehat{BJC} = \pi - \dfrac{(\widehat{B} + \widehat{C})}{2} = \dfrac{(\pi + \widehat{A})}{2}$.
- soit P est un point de la bissectrice intérieure de l'angle A se trouvant à l'intérieur du triangle ABC et tel que $\widehat{BPC} = \dfrac{(\pi + \widehat{A})}{2}$.
On remarque que $\widehat{BPC} = \pi - (\widehat{PBC} + \widehat{PCB})$ croît lorsque P s'éloigne de A en restant sur la bissectrice intérieure de l'angle A, donc si $P \neq J$ on a soit $\widehat{BPC} > \widehat{BJC} = \dfrac{(\pi + \widehat{A})}{2}$ soit $\widehat{BPC} < \widehat{BJC} = \dfrac{(\pi + \widehat{A})}{2}$.
On en déduit donc que P et J sont confondus.

6.3.3 Une démonstration géométrique pas courante

On va démontrer le théorème pour un triangle ABC d'angles de mesure $3*a$, $3*b$ et $3*c$ ($a+b+c=\frac{\pi}{3}$).
La forme de la preuve n'est pas courante : on part d'un triangle PQR équilatéral et on construit à l'extérieur de ce triangle les triangles isocèles UQR, VPR et WPQ d'angles :
$\widehat{QRU}=\widehat{UQR}=a+b=\frac{\pi}{3}-c$,
$\widehat{VPR}=\widehat{VRP}=b+c=\frac{\pi}{3}-a$ et
$\widehat{WPQ}=\widehat{WQP}=c+a=\frac{\pi}{3}-b$.
En effet on sait d'après les calculs faits par `Xcas` que le triangle PQR est équilatéral et que les triangles UQR, VPR et WPQ sont isocèles.
Puis on construit le triangle ABC :
A est l'intersection de UR et de PW,
B est l'intersection de UQ et de PV et
C est l'intersection de VR et de QW.

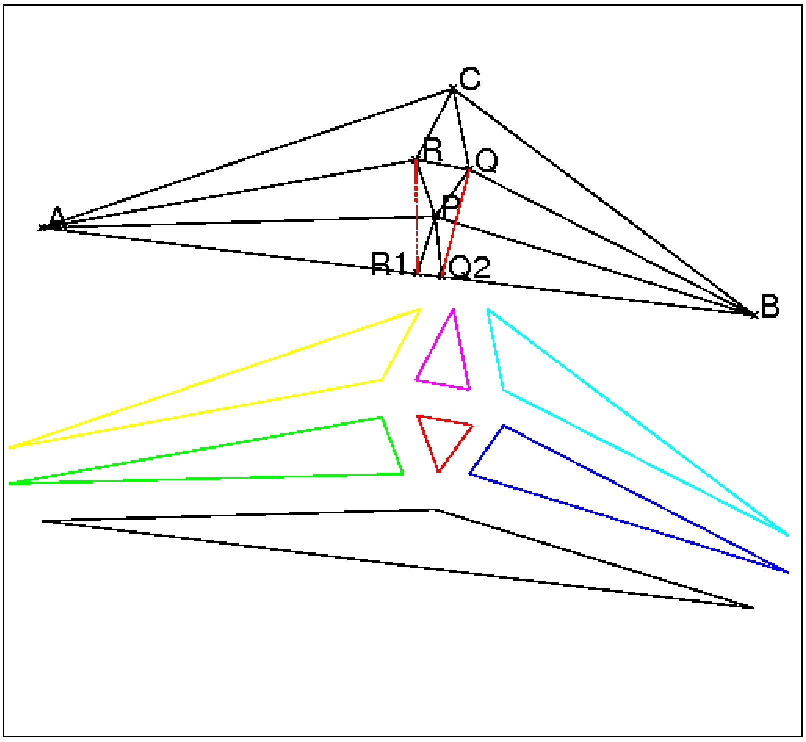


On obtient la figure ci-dessus :
On peut montrer par des considération d'angles que V se trouve à l'intérieur du triangle ARP ($\widehat{PRU}+\widehat{RPW}>\pi$ et $\widehat{VPR}<\pi-\widehat{RPW}$) etc...
On va montrer qu'alors P, Q, R sont les points de concours des trissectrices intérieures de Morley.
Calculons quelques angles :
$\widehat{APV}=\pi-(\dfrac{\pi}{3}-b)-\dfrac{\pi}{3}-(\dfrac{\pi}{3}-a)=a+b=\dfrac{\pi}{3}-c$ donc
$\widehat{APR}=\dfrac{\pi}{3}-c+\dfrac{\pi}{3}-a=\dfrac{\pi}{3}+b$
de même

$\widehat{ARV} = \dfrac{\pi}{3} - b$ et
$\widehat{ARP} = \dfrac{\pi}{3} - b + \dfrac{\pi}{3} - a = \dfrac{\pi}{3} + c$
donc $\widehat{PAR} = a$
de même $\widehat{PBQ} = b$ et $\widehat{QCR} = c$
$\widehat{AUB} = \widehat{RUQ} = \pi - (\dfrac{2\pi}{3} - c) = \dfrac{\pi}{3} + 2 * c$
$\widehat{APB} = \pi - \widehat{APV} = \dfrac{2\pi}{3} + c = \dfrac{\pi}{2} + \dfrac{\pi}{6} - c = \dfrac{\pi}{2} + \dfrac{\widehat{AUB}}{2}$
PU est la médiatrice de RQ puisque $PQ = PR$ et $UQ = UR$.
PU est donc aussi la bissectrice intérieure de l'angle U et $\widehat{APB} = \frac{(\pi+\widehat{AUB})}{2}$ donc d'après le lemme :
P est le point de concours des bissectrices intérieures du triangle UAB de même,
Q est le point de concours des bissectrices intérieures du triangle VBC et
R est le point de concours des bissectrices intérieures du triangle WAC.
L'angle A vaut donc $3a$, l''angle B vaut donc $3b$ et l''angle C vaut donc $3c$ ce qui prouve que P, Q, R sont les points de concours des trissectrices intérieures de Morley.

6.3.4 Une démonstration selon Conway

On suppose tout d'abord le problème résolu c'est à dire que le triangle PQR est équilatéral.
Soient :
a, b, c le tiers des mesures des angles du triangle ABC on a donc $a + b + c = \frac{\pi}{3}$,
R_1 le symétrique de R par rapport à PA : R_1 se trouve sur AB puisque $\widehat{PAR} = \widehat{PAB} = a$,
et de même, Q_2 le symétrique de Q par rapport à PB se trouve sur AB.
Le triangle PQ_2R_1 est donc isocèle ($PR_1 = PR = PQ = PQ_2$),
on en déduit que $\widehat{Q_2R_1P} = \widehat{PQ_2R_1}$, et par symétrie que $\widehat{ARP} = \widehat{PQB} = \gamma$.
On a : $\widehat{APB} + \widehat{BPQ} + \widehat{QPR} + \widehat{RPA} = 2\pi$ c'est à dire
$(\pi - a - b) + (\pi - b - \gamma) + \dfrac{\pi}{3} + (\pi - a - \gamma) = 2\pi$ soit
$\gamma = 2 * \dfrac{\pi}{3} - a - b = c + \dfrac{\pi}{3}$.
Donc le triangle APR a comme angles : $a,\ c + \pi/3,\ b + \pi/3$,
le triangle BQP a comme angles : $b,\ c + \frac{\pi}{3},\ a + \frac{\pi}{3}$,
de même le triangle CRQ a comme angles : $c,\ b + \frac{\pi}{3},\ a + \frac{\pi}{3}$.
L'écriture en Latex de cette figure se trouve dans le fichier `morleypuzzel.tex`.



L'idée de Conway est de faire un puzzle avec 7 pièces (cf figure) :
avec ces 7 pièces on va pouvoir reconstituer un triangle ABC d'angle $3a$, $3b$ et $3c$ dans lequel les trissectrices se coupent selon le triangle équilatéral de Morley.
Les pièces du puzzles sont :

- 1 triangle équilatéral de cotés de longueur 1 (en rouge),
- 3 triangles
  - le premier PAR (en vert) est défini par PR de longueur 1, l'angle $\widehat{P}$ de mesure $\frac{\pi}{3}+b$ et l'angle $\widehat{R}$ de mesure $\frac{\pi}{3}+c$ et APR est de sens direct.
    On a donc :
    $PA = \sin(c+\pi/3)/\sin(a)$, $RA = \sin(b+\pi/3)/\sin(a)$ et l'angle $\widehat{A}$ est de mesure a,
  - le second PQB (en bleu) est défini par PQ de longueur 1, l'angle $\widehat{P}$ de mesure $\frac{\pi}{3}+a$ et l'angle $\widehat{Q}$ de mesure $\frac{\pi}{3}+c$ et BQP est de sens direct.
    On a donc :
    $PB = \sin(c+\frac{\pi}{3})/\sin(b)$, $QB = \sin(a+\frac{\pi}{3})/\sin(b)$ et l'angle $\widehat{B}$ est de mesure b,
  - le troisième QRC (en magenta) est défini par RQ de longueur 1, l'angle $\widehat{R}$ de mesure $\frac{\pi}{3}+a$ et l'angle $\widehat{Q}$ de mesure $\frac{\pi}{3}+b$ et CRQ est de sens direct.
    On a donc :
    $QC = \sin(a+\frac{\pi}{3})/\sin(c)$, $RC = \sin(b+\frac{\pi}{3})/\sin(c)$ et l'angle $\widehat{C}$ est de mesure c.

— 3 triangles
  — le premier $PAB1$ (en noir) est défini par un coté de longueur PA trouvée ci-dessus ($PA = \sin(c + \frac{\pi}{3})/\sin(a)$) l'angle $\widehat{A}$ de mesure a et l'angle $\widehat{P}$ de mesure $c+2*\frac{\pi}{3}$. Dans ce triangle l'angle $\widehat{B1}$ a comme mesure b et on a : $PB1 = PA*\sin(a)/\sin(b) = \sin(c+\frac{\pi}{3})/\sin(b) = PB$,
  — le second $RCA1$ (en jaune) est défini par un coté de longueur RC trouvée ci-dessus ($RC = \sin(b + \frac{\pi}{3})/\sin(c)$) l'angle $\widehat{C}$ de mesure c et l'angle $\widehat{R}$ de mesure $b + \frac{2\pi}{3}$. Dans ce triangle l'angle $\widehat{A1}$ a comme mesure a et on a : $RA1 = RC * \sin(c)/\sin(a) = \sin(b + \frac{\pi}{3})/\sin(a) = RA$,
  — le troisième $QBC1$ (en cyan) est défini par un coté de longueur QB trouvée ci-dessus ($QB = \sin(a + \frac{\pi}{3})/\sin(b)$) l'angle $\widehat{B}$ de mesure b et l'angle $\widehat{Q}$ de mesure $a + \frac{2\pi}{3}$. Dans ce triangle l'angle $\widehat{C1}$ a comme mesure c et on a : $QC1 = QB * \sin(b)/\sin(c) = \sin(a+\frac{\pi}{3})/\sin(c) = QC$.

Il est alors facile de montrer, par des considérations de mesure d'angles et de longueurs, que les triangles s'emboitent bien.

6.3.5 Une autre démonstration

On pose :
$\widehat{ARP} = \gamma$ et
$\widehat{APR} = \beta$.
Si R est le rayon du cercle circonscrit du triangle ABC on a la relation :

$$\frac{AB}{\sin(3c)} = \frac{AC}{\sin(3b)} = \frac{BC}{\sin(3a)} = 2R$$

En considérant les triangles APR, APB, ARC, ABC on peut montrer en utilisant les relations ci-dessus que :

$$PR = \frac{2R\sin(a)\sin(b)\sin(3c)}{\sin(\gamma)\sin(c+2\pi/3)} = \frac{2R\sin(a)\sin(3b)\sin(c)}{\sin(\beta)\sin(b+2\pi/3)}$$

On en déduit après différents calculs laissés au lecteur (voir les indications ci-dessous) que :
$\gamma = c + \frac{\pi}{3}$, $\beta = b + \frac{\pi}{3}$ et
$PR = 8R\sin(a)\sin(b)\sin(c)$
Indications :
1/ Pour tout x on a :
$2\sin(x + \frac{\pi}{3})\sin(x + \frac{2\pi}{3}) = \frac{1}{2} - \cos(2x + \pi) = \frac{1}{2} + \cos(2x)$ et
$2\sin(x)(\frac{1}{2} - \cos(2x)) = 3\sin(x) - 4\sin(x)^3 = \sin(3x)$ donc
$4\sin(x)\sin(x + \frac{\pi}{3})\sin(x + \frac{2\pi}{3}) = \sin(3x)$.
On obtient donc en remplacant $\sin(3c)$ et $\sin(3b)$ dans :

$$\frac{2R\sin(a)\sin(b)\sin(3c)}{\sin(\gamma)\sin(c+\frac{2\pi}{3})} = \frac{2R\sin(a)\sin(3b)\sin(c)}{\sin(\beta)\sin(b+\frac{2\pi}{3})}$$

l'égalité :
$\sin(\beta)\sin(c + \frac{\pi}{3}) = \sin(\gamma)\sin(b + \frac{\pi}{3})$

et donc
$PR = \frac{8R\sin(a)\sin(b)\sin(c)\sin(c+\frac{\pi}{3})}{\sin(\gamma)}$ 2/ si $\sin(a) \neq 0,\ \sin(b) \neq 0,\ \sin(a+b) \neq 0$, le système :

$$\begin{cases} x+y=a+b \\ \sin(x)/\sin(a)=\sin(y)/\sin(b) \end{cases}$$

a comme solution $x = a + k\pi,\ y = b - k\pi$ où $k \in Z.$
En effet on a $y = a + b - x$
donc $\sin(x)\sin(b) = \sin(a)(\sin(a+b)\cos(x) - \cos(a+b)\sin(x))$
$\sin(x)(\sin(b) + \sin(a)\cos(a+b)) = \cos(x)\sin(a)\sin(a+b)$
on développe $\cos(a+b)$ et on obtient :
$\sin(x)\cos(a)\sin(a+b) = \cos(x)\sin(a)\sin(a+b)$
et donc puisque $\sin(a+b) \neq 0$
$\sin(x-a) = 0$ d'ou le résultat.
Donc le système :

$$\begin{cases} \sin(\beta)/\sin(b+\frac{\pi}{3}) = \sin(\gamma)/\sin(c+\frac{\pi}{3}) \\ \beta+\gamma = b+\frac{\pi}{3}+c+\frac{\pi}{3} = \pi - a \end{cases}$$

avec $\sin(b+\frac{\pi}{3}) \neq 0,\ \sin(c+\frac{\pi}{3}) \neq 0,\ \sin(\pi-a) \neq 0$
a pour solutions :
$\beta = b + \frac{\pi}{3} + k\pi$ et $\gamma = (c+\frac{\pi}{3}) - k\pi$ où $k \in Z.$
Donc $\sin(\gamma) = \sin(c+\frac{\pi}{3})$ et donc
$PR = 8 * R * \sin(a) * \sin(b) * \sin(c)$ La formule etant symétrique par rapport à $a,\ b,\ c$ on a :
$PR = 8 * R * \sin(a) * \sin(b) * \sin(c) = RQ = PQ$
Le triangle PQR est donc équilatèral.

6.4 Le théorème de Morley généralisé

Soit un triangle ABC et ses trissectrices.
On considère toutes les intersections de ces trissectrices.
Combien trouve-t-on de triangles équilatéraux ?

6.4.1 Les trissectrices d'un angle de droites

Soit un angle de droites (AB, AC) de mesure $3t + k\pi$ (k entier) : il y a 6 trissectrices ($D1$, $D2$, $D3$, $D4$, $D5$, $D6$) qui forment avec AB des angles de mesure :

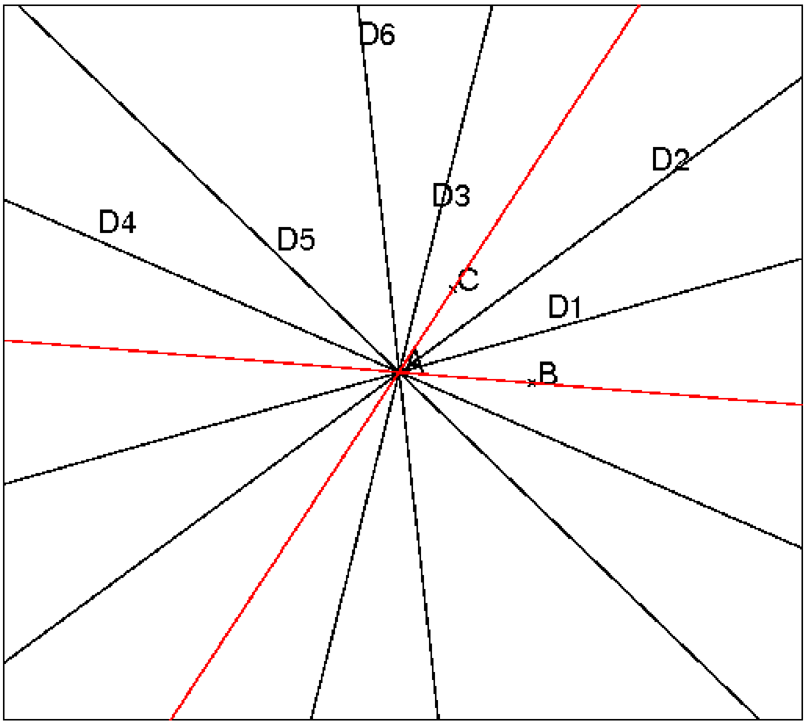
$$t,\ 2t,\ t+\frac{\pi}{3},\ 2t+\frac{2\pi}{3},\ t+\frac{2\pi}{3},\ 2t+\frac{4\pi}{3} = 2t+\frac{\pi}{3}+\pi$$

On considère deux groupes de trois trissectrices ($D1$, $D3$, $D5$) et ($D2$, $D4$, $D6$) : dans chaque groupe les trissectrices se déduisent l'une de l'autre par des rotations d'angle $\frac{\pi}{3}$.
Voici les instructions contenu dans le fichier `morleytri6` qui permet de tracer les 6 trissectrices de l'angle de droite (AB,AC) :

```
A:= point(-1.2*i);
B:= point(3.7-1.5*i);
```

```
C:= point(1.5+1.1*i);
t:=angle(A,B,C)/3;
D1:=droite(A,A+(B-A)*exp(i*t));
D2:=droite(A,A+2*(B-A)*exp(i*t*2));
D3:=droite(A,A+(B-A)*exp(i*(t+pi/3)));
D4:=droite(A,A+2*(B-A)*exp(i*2*(t+pi/3)));
D5:=droite(A,A+(B-A)*exp(i*(t+2*pi/3)));
D6:=droite(A,A+2*(B-A)*exp(i*(2*t+pi/3)));
couleur(droite(A,B),1);
couleur(droite(A,C),1);
```

On obtient la figure :



6.4.2 Calcul numérique des différentes longueurs

On fait tout d'abord avec `Xcas` un calcul numérique pour avoir une idée du résultat. Quitte à faire une similitude on peut choisir `A` à l'origine et `B` au point d'affixe 1.
La site d'instructions ci-dessous calcule les coordonnées des 108 points d'intersections des trissectrices entre elles : il y a $3 \times 6 = 18$ trissectrices, sur chaque trissectrices il y a 6+6=12 points d'intersections donc en tout $18 \times 12/2 = 108$ points d'intersections que l'on met dans la liste `P`.
Puis on met dans une matrice symétrique `LO` de dimension 108×108 le carré des distances entre ces 108 points.
On met ensuite dans la variable `trequi` la liste des triplets formant un triangle

équilatéral, selon le calcul numérique.
`A` est une liste de 7 éléments qui sont :
`A[0]=0`, c'est l'affixe du sommet A du triangle,
`A[1]` est l'affixe d'un point situé sur la trissectrice `AD1`,
`A[2]` est l'affixe d'un point situé sur la trissectrice `AD2`, etc...
de même pour `B` et `C` (`B[0]=1`...).
Voici les instructions qui calcule les coordonnées des 108 points d'intersections des trissectrices entre elles :

```
a1:=0.2;
b1:=0.4;
A:=[0,1+i*texpand(tan(a1)),1+i*texpand(tan(2*a1)),
   1+i*texpand(tan(pi/3+a1)),
   1+i*texpand(tan(2*a1+2*pi/3)),1+i*texpand(tan(a1+2*pi/3)),
   1+i*texpand(tan(pi/3+2*a1))];
B:=[1,i*texpand(tan(2*b1)),i*texpand(tan(b1)),
    i*texpand(tan(2*b1+2*pi/3)),i*texpand(tan(b1+pi/3)),
    i*texpand(tan(pi/3+2*b1)),i*texpand(tan(2*pi/3+b1))];
C0:=texpand(tan(b1*3)/(tan(a1*3)+tan(b1*3))*(1+i*tan(a1*3)));
C:=[C0,C0+1+i*texpand(tan(pi/3+2*a1-b1)),
    C0+1+i*texpand(tan(2*pi/3+a1-2*b1)),
    C0+1+i*texpand(tan(2*pi/3+2*a1-b1)),
    C0+1+i*texpand(tan(pi/3+a1-2*b1)),
    C0+1+i*texpand(tan(2*a1-b1)),C0+1+i*texpand(tan(a1-2*b1))];
P:=[];
for (k:=1;k<=6;k++) {
 for (j:=1;j<=6;j++){
   P:=concat(P,affixe((inter(droite(A[0],A[k]),
      droite(B[0],B[j])))[0]));
   P:=concat(P,affixe((inter(droite(B[0],B[k]),
      droite(C[0],C[j])))[0]));
   P:=concat(P,affixe((inter(droite(A[0],A[k]),
      droite(C[0],C[j])))[0]));
 }
};
LO:=[];
for (k:=0;k<108;k++) {
  LOL:=[];
  for (j:=0;j<108;j++){
    LOL:=concat(LOL,longueur2(P[k],P[j]));
  }
LO:=append(LO,LOL);
};
trequi:=[];
for (k:=0;k<106;k++) {
  for (j:=k+1;j<107;j++){
    l:=LO[k,j];
    for (s:=j+1;s<108;s++){
      if ((abs(normal(l-LO[j,s]))<0.0000001) and
```

```
          (abs(normal(1-LO[k,s]))<0.0000001)){
        trequi:=append(trequi,[k,j,s]);
      }
    }
  }
};
trequi;
```

On tape dans `Xcas` : `size(trequi)` : on obtient `54`.
Cette liste contient donc 54 triplets ce qui veut dire qu'il y a numériquement 54 triangles équilatéraux. Parmi ces triangles il y en a 18 seulement qui mettent en jeu les trissectrices des 3 angles du triangle.
Les autres triangles ont des sommets qui sont des intersections de trissectrices de 2 angles du triangle : ceci était à prévoir et se démontre facilement.
En effet si on considère deux angles de sommets A et B et leurs trissectrices : les 3 trissectrices $AD1$, $AD3$, $AD5$ de l'angle A se déduisent l'une de l'autre par des rotations de centre A et d'angle $\frac{\pi}{3}$ et les intersections des 3 trissectrices ($AD1$, $AD3$, $AD5$) de l'angle A avec les 3 trissectrices $BD1$, $BD3$, $BD5$ de l'angle B déterminent 3 triangles équilatéraux.

Plus précisément soient M le point d'intersection de $AD1$ avec $BD1$, N le point d'intersection de $AD3$ avec $BD3$, O le point d'intersection de $AD5$ avec $BD5$. Alors le triangle MNO est équilatèral.
On tape :

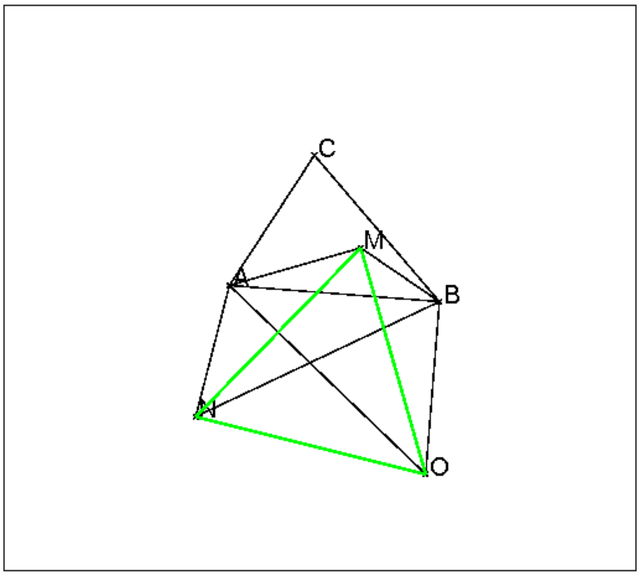
```
trissect(A,B,C):={
local t;
t:=angle(A,B,C)/3;
D1:=droite(A,A+(B-A)*exp(i*t));
D2:=droite(A,A+(B-A)*exp(i*t*2));
D3:=droite(A,A+(B-A)*exp(i*(t+pi/3)));
D4:=droite(A,A+(B-A)*exp(i*2*(t+pi/3)));
D5:=droite(A,A+(B-A)*exp(i*(t+2*pi/3)));
D6:=droite(A,A+(B-A)*exp(i*2*(t+2*pi/3)));
return([D1,D2,D3,D4,D5,D6,
  couleur(droite(A,B),1),couleur(droite(A,C),1)]);
}:;
```

Puis, on tape :

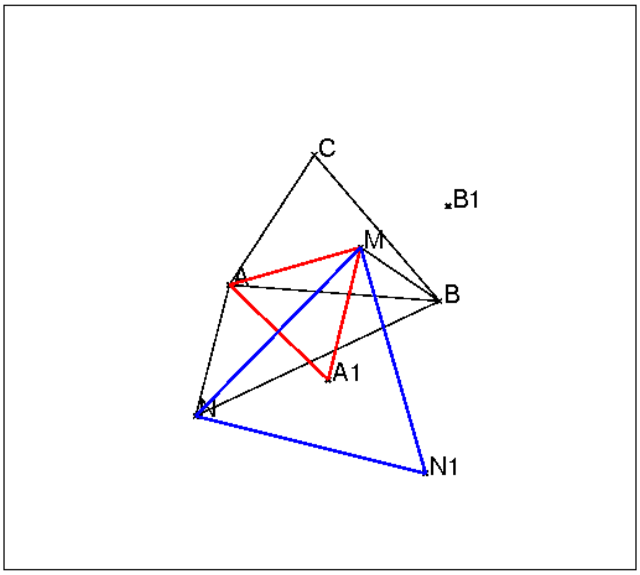
```
A:= point(-1.2*i);
B:= point(3.7-1.5*i);
C:= point(1.5+1.1*i);
TA:=trissect(A,B,C):;
TB:=trissect(B,C,A):;
M:=inter_droite(TA[0],TB[0]);
N:=inter_droite(TA[2],TB[2]);
O:=inter_droite(TA[4],TB[4]);
triangle(A,B,C);
segment(A,M),segment(B,M);
segment(A,N),segment(B,N);
```

```
segment(A,O),segment(B,O);
triangle(M,N,O,affichage=2+epaisseur_ligne_2)
```

On obtient :



Montrons par exemple que MNO est un triangle équilatéral : la rotation de centre M et d'angle $\frac{\pi}{3}$ transforme A en $A1$, B en $B1$ et N en $N1$.



les triangles $MAA1$ et $MNN1$ sont des triangles équilatéraux directs (triangle isocèle ayant un angle de $\pi/3$). M étant sur $AD1$, $A1$ se trouve donc sur $AD5$. $A1N1$ fait un angle de $\pi/3$ avec AN. N étant sur $AD3$, $A1N1$ est donc parallèle à $AD5$. On a ainsi montré que $N1$ se trouve sur $AD5$ c'est à dire que $A,\ A1,\ N1$ sont alignés.

On montre de même que $B,\ B1,\ N1$ sont alignés et donc $N1$ se trouve sur $BD5$ d'où $N1$ et O sont confondus. Ce qui prouve que MNO est un triangle équilatéral. Donc les 3 trissectrices $AD1,\ AD3,\ AD5$ de l'angle A forment avec les 6 trissectrices de l'angle B, 6 triangles équilatéraux. Donc les 6 trissectrices de l'angle A forment avec les 6 trissectrices de l'angle B, 12 triangles équilatéraux. Donc il existe $3 \times 12 = 36$ triangles équilatéraux de ce type.

6.4.3 Les 18 triangles équilatéraux

Qui sont-ils ?

On modifie le programme `morley108` pour ne tester que des triangles obtenus par les intersections des trissectrices des 3 angles.
Un triangle est désigné soit :
- par un triplet qui sont ses sommets par exemple : `(A1B2, B1C2, C1A2)` ce qui veut dire que le premier sommet est l'intersection de la trissectrice 1 issue de `A` avec la trissectrice 2 issue de `B` etc... soit
- par un triplet de couples par exemple : `[[1,2],[1,4],[3,2]]` désigne le même triangle que `(A1B2, B1C4, C3A2)`.
On écrit le fichier `morley18` qui met dans la liste `trequi` les triangles qui sont numériquement équilatéraux :

```
a1:=0.2;
b1:=0.4;
A:=[0,1+i*texpand(tan(a1)),1+i*texpand(tan(2*a1)),
1+i*texpand(tan(pi/3+a1)),1+i*texpand(tan(2*a1+2*pi/3)),
1+i*texpand(tan(a1+2*pi/3)),1+i*texpand(tan(pi/3+2*a1))];
B:=[1,i*texpand(tan(2*b1)),i*texpand(tan(b1)),
i*texpand(tan(2*b1+2*pi/3)),i*texpand(tan(b1+pi/3)),
i*texpand(tan(pi/3+2*b1)),i*texpand(tan(2*pi/3+b1))];
C0:=texpand(tan(b1*3)/(tan(a1*3)+tan(b1*3))*(1+i*tan(a1*3)));
C:=[C0,C0+1+i*texpand(tan(pi/3+2*a1-b1)),
    C0+1+i*texpand(tan(2*pi/3+a1-2*b1)),
    C0+1+i*texpand(tan(2*pi/3+2*a1-b1)),
    C0+1+i*texpand(tan(pi/3+a1-2*b1)),
    C0+1+i*texpand(tan(2*a1-b1)),C0+1+i*texpand(tan(a1-2*b1))];
P1:=[];
P2:=[];
P3:=[];
for (k:=1;k<=6;k++) {
  for (j:=1;j<=6;j++){
    P1:=concat(P1,affixe((inter(droite(A[0],A[k]),
        droite(B[0],B[j])))[0]));
    P3:=concat(P3,affixe((inter(droite(A[0],A[k]),
        droite(C[0],C[j])))[0]));
    P2:=concat(P2,affixe((inter(droite(B[0],B[k]),
        droite(C[0],C[j])))[0]));
  }
};
```

```
LO12:=[];
for (k:=0;k<36;k++) {
  LOL12:=[];
  for (j:=0;j<36;j++){
    LOL12:=concat(LOL12,longueur2(P1[k],P2[j]));
  }
  LO12:=append(LO12,LOL12);
};
LO23:=[];
for (k:=0;k<36;k++) {
  LOL23:=[];
  for (j:=0;j<36;j++){
    LOL23:=concat(LOL23,longueur2(P2[k],P3[j]));
  }
  LO23:=append(LO23,LOL23);
};
LO13:=[];
for (k:=0;k<36;k++) {
  LOL13:=[];
  for (j:=0;j<36;j++){
    LOL13:=concat(LOL13,longueur2(P1[k],P3[j]));
  }
  LO13:=append(LO13,LOL13);
};
trequi:=[];
for (k:=0;k<36;k++) {
  for (j:=0;j<36;j++){
    l:=LO12[k,j];
    for (s:=0;s<36;s++){
      if ((abs(normal(l-LO23[j,s]))<0.0000001) and
          (abs(normal(l-LO13[k,s]))<0.0000001)){
        trequi:=append(trequi,[[iquo(k,6)+1,irem(k,6)+1],
                               [iquo(j,6)+1,irem(j,6)+1],
                               [irem(s,6)+1,iquo(s,6)+1]]);
      }
    }
  }
};
trequi;
```

On fait `Charger session` du menu `Fich` de `Xcas` et on sélectionne `morley18` du répértoire `examples/geo` pour exécuter ce fichier.
La liste `trequi` contient cette fois 18 triplets, il y donc numèriquement 18 triangles équilatéraux dont les sommets sont définis comme intersection des trissectrices des 3 angles du triangle.
On trouve comme contenu de `trequi` :

```
[[[1,2],[1,2],[1,2]],
[[1,2],[1,4],[3,2]],
```

```
[[1,4],[3,2],[1,2]],
[[1,4],[3,6],[5,2]],
[[1,6],[5,4],[3,2]],
[[1,6],[5,6],[5,2]],
[[3,2],[1,2],[1,4]],
[[3,2],[1,6],[5,4]],
[[3,4],[3,4],[3,4]],
[[3,4],[3,6],[5,4]],
[[3,6],[5,2],[1,4]],
[[3,6],[5,4],[3,4]],
[[5,2],[1,4],[3,6]],
[[5,2],[1,6],[5,6]],
[[5,4],[3,2],[1,6]],
[[5,4],[3,4],[3,6]],
[[5,6],[5,2],[1,6]],
[[5,6],[5,6],[5,6]]]
```

Chaque triplet désigne les sommets d'un triangle. Avec la notation `A1B2` qui désigne l'intersection de la trissectrice 1 issue de `A` avec la trissectrice 2 issue de `B` on a :

```
(A1B2, B1C2, C1A2),
(A1B2, B1C4, C3A2),
(A1B4, B3C2, C1A2),
(A1B4, B3C6, C5A2),
(A1B6, B5C4, C3A2),
(A1B6, B5C6, C5A2),
(A3B2, B1C2, C1A4),
(A3B2, B1C6, C5A4),
(A3B4, B3C4, C3A4),
(A3B4, B3C6, C5A4),
(A3B6, B5C2, C1A4),
(A3B6, B5C4, C3A4),
(A5B2, B1C4, C3A6),
(A5B2, B1C6, C5A6),
(A5B4, B3C2, C1A6),
(A5B4, B3C4, C3A6),
(A5B6, B5C2, C1A6),
(A5B6, B5C6, C5A6)
```

6.4.4 La figure

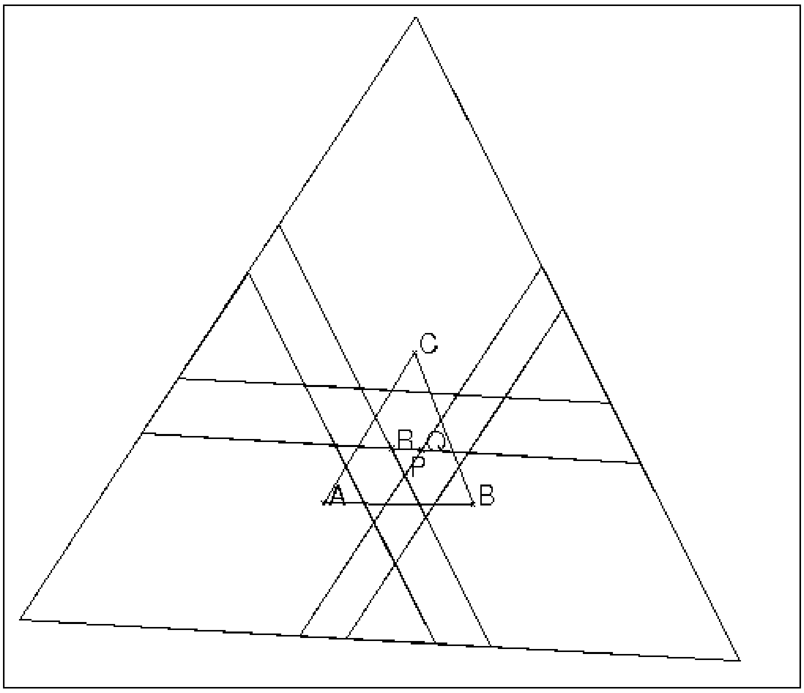
On dessine les 18 triangles.
On écrit un fichier `morleydess18` qui va calculer les coordonnées des 27 points sommets de ces 18 triangles et dessiner ces 18 triangles.
On pourra avec un tel dessin faire bouger `A` ou `B` ou `C`.
On note `DA` (resp `DB`, `DC`) la liste des 6 trissectrices de l'angle `A` (resp `B`, `C`).
On remarquera dans la définition de `DA, DB, DC` :
- un zéro au début de la liste pour que la première trissectrice soit d'indice 1 etc...
- un zéro en fin de liste pour que les objets graphiques situés dans la liste ne soient

pas dessinés (pour ne pas surcharger la figure) (on aurait aussi pu employer la fonction `nodisp`).

```
A:=point(-2.35-i*2.28);
B:=point(-0.684-i*2.3);
C:=point(-1.35-i*0.61);
a1:=angle(A,B,C)/3;
a2:=angle(B,C,A)/3;
a3:=pi/3-a1-a2;
DA:=0,droite(A,A+(B-A)*exp(i*a1)):;
DA:=DA,droite(A,A+(B-A)*exp(i*a1*2)):;
DA:=DA,droite(A,A+(B-A)*exp(i*(a1+pi/3))):;
DA:=DA,droite(A,A+(B-A)*exp(i*2*(a1+pi/3))):;
DA:=DA,droite(A,A+(B-A)*exp(i*(a1+2*pi/3))):;
DA:=DA,droite(A,A+(B-A)*exp(i*2*(a1+2*pi/3))),0:;
DB:=0,droite(B,B+(C-B)*exp(i*a2)):;
DB:=DB,droite(B,B+(C-B)*exp(i*a2*2)):;
DB:=DB,droite(B,B+(C-B)*exp(i*(a2+pi/3))):;
DB:=DB,droite(B,B+(C-B)*exp(i*2*(a2+pi/3))):;
DB:=DB,droite(B,B+(C-B)*exp(i*(a2+2*pi/3))):;
DB:=DB,droite(B,B+(C-B)*exp(i*2*(a2+2*pi/3))),0:;
DC:=0,droite(C,C+(A-C)*exp(i*a3)):;
DC:=DC,droite(C,C+(A-C)*exp(i*a3*2)):;
DC:=DC,droite(C,C+(A-C)*exp(i*(a3+pi/3))):;
DC:=DC,droite(C,C+(A-C)*exp(i*2*(a3+pi/3))):;
DC:=DC,droite(C,C+(A-C)*exp(i*(a3+2*pi/3))):;
DC:=DC,droite(C,C+(A-C)*exp(i*2*(a3+2*pi/3))),0:;
P1:=[];
P1:=concat(P1,affixe((inter(DA[1],DB[2]))[0]));
P1:=concat(P1,affixe((inter(DA[1],DB[4]))[0]));
P1:=concat(P1,affixe((inter(DA[1],DB[6]))[0]));
P1:=concat(P1,affixe((inter(DA[3],DB[2]))[0]));
P1:=concat(P1,affixe((inter(DA[3],DB[4]))[0]));
P1:=concat(P1,affixe((inter(DA[3],DB[6]))[0]));
P1:=concat(P1,affixe((inter(DA[5],DB[2]))[0]));
P1:=concat(P1,affixe((inter(DA[5],DB[4]))[0]));
P1:=concat(P1,affixe((inter(DA[5],DB[6]))[0]));
P2:=[];
P2:=concat(P2,affixe((inter(DB[1],DC[2]))[0]));
P2:=concat(P2,affixe((inter(DB[1],DC[4]))[0]));
P2:=concat(P2,affixe((inter(DB[1],DC[6]))[0]));
P2:=concat(P2,affixe((inter(DB[3],DC[2]))[0]));
P2:=concat(P2,affixe((inter(DB[3],DC[4]))[0]));
P2:=concat(P2,affixe((inter(DB[3],DC[6]))[0]));
P2:=concat(P2,affixe((inter(DB[5],DC[2]))[0]));
P2:=concat(P2,affixe((inter(DB[5],DC[4]))[0]));
P2:=concat(P2,affixe((inter(DB[5],DC[6]))[0]));
P3:=[];
P3:=concat(P3,affixe((inter(DC[1],DA[2]))[0]));
```

```
P3:=concat(P3,affixe((inter(DC[1],DA[4]))[0]));
P3:=concat(P3,affixe((inter(DC[1],DA[6]))[0]));
P3:=concat(P3,affixe((inter(DC[3],DA[2]))[0]));
P3:=concat(P3,affixe((inter(DC[3],DA[4]))[0]));
P3:=concat(P3,affixe((inter(DC[3],DA[6]))[0]));
P3:=concat(P3,affixe((inter(DC[5],DA[2]))[0]));
P3:=concat(P3,affixe((inter(DC[5],DA[4]))[0]));
P3:=concat(P3,affixe((inter(DC[5],DA[6]))[0]));
triangle(A,B,C);
triangle(P1[0],P2[0],P3[0]);
P:=point(P1[0]);
Q:=point(P2[0]);
R:=point(P3[0]);
triangle(P1[0],P2[1],P3[3]);
triangle(P1[1],P2[3],P3[0]);
triangle(P1[1],P2[5],P3[6]);
triangle(P1[2],P2[7],P3[3]);
triangle(P1[2],P2[8],P3[6]);
triangle(P1[3],P2[0],P3[1]);
triangle(P1[3],P2[2],P3[7]);
triangle(P1[4],P2[4],P3[4]);
triangle(P1[4],P2[5],P3[7]);
triangle(P1[5],P2[6],P3[1]);
triangle(P1[5],P2[7],P3[4]);
triangle(P1[6],P2[1],P3[5]);
triangle(P1[6],P2[2],P3[8]);
triangle(P1[7],P2[3],P3[2]);
triangle(P1[7],P2[4],P3[5]);
triangle(P1[8],P2[6],P3[2]);
triangle(P1[8],P2[8],P3[8]);
```

On obtient :



On voit que ces triangles ont leur cotés parallèles.
On peut le faire montrer par `Xcas` avec le programme suivant se trouvant dans le fichier `morleypara` :

```
est_parallele(droite(P1[0],P2[0]),droite(P1[8],P2[8]));
est_parallele(droite(P1[0],P2[0]),droite(P1[0],P3[6]));
est_parallele(droite(P1[0],P2[0]),droite(P1[4],P2[4]));
est_parallele(droite(P1[4],P2[4]),droite(P1[4],P3[1]));
est_parallele(droite(P1[0],P2[0]),droite(P1[8],P2[8]));
est_parallele(droite(P1[8],P2[8]),droite(P1[8],P3[5]));
est_parallele(droite(P1[0],P2[0]),droite(P3[6],P2[5]));
est_parallele(droite(P1[4],P2[4]),droite(P1[2],P3[3]));
```

6.4.5 La démonstration avec `Xcas`

Pour montrer plus facilement que les 18 triangles trouvés sont équilatéraux on va écrire dans le fichier `equimorley` une fonction ayant `m,n,p,q,r,s` comme paramètres représentant le triangle `AmBn,BpCq,CrAs` et qui teste si ce triangle est équilatéral :

```
equimore(m,n,p,q,r,s):={
assume(a1=0.3);
assume(a2=0.4);
A:=[0,1+i*texpand(tan(a1)),1+i*texpand(tan(2*a1)),
```

```
1+i*texpand(tan(pi/3+a1)),1+i*texpand(tan(2*a1+2*pi/3)),
1+i*texpand(tan(a1+2*pi/3)),1+i*texpand(tan(pi/3+2*a1))];
B:=[1,i*texpand(tan(2*a2)),i*texpand(tan(a2)),
i*texpand(tan(2*a2+2*pi/3)),i*texpand(tan(a2+pi/3)),
i*texpand(tan(pi/3+2*a2)),i*texpand(tan(2*pi/3+a2))];
C0:=texpand(tan(a2*3)/(tan(a1*3)+tan(a2*3))*(1+i*tan(a1*3)));
C:=[C0,C0+1+i*texpand(tan(pi/3+2*a1-a2)),
C0+1+i*texpand(tan(2*pi/3+a1-2*a2)),
C0+1+i*texpand(tan(2*pi/3+2*a1-a2)),
C0+1+i*texpand(tan(pi/3+a1-2*a2)),
C0+1+i*texpand(tan(2*a1-a2)),C0+1+i*texpand(tan(a1-2*a2))];
P:=affixe((inter(droite(A[0],A[m]),droite(B[0],B[n])))[0]);
Q:=affixe((inter(droite(B[0],B[p]),droite(C[0],C[q])))[0]);
R:=affixe((inter(droite(C[0],C[r]),droite(A[0],A[s])))[0]);
lpq:=longueur2(P,Q);
lpr:=longueur2(P,R);
lqr:=longueur2(Q,R);
return([normal(lpq-lpr),normal(lpq-lqr)]);
};
```

On fait `Charger session` du menu `Fich` de `Xcas` et on selectionne `equimorley` du répértoire `examples/geo` pour exécuter ce fichier, puis :
`equimorley(5,6,5,6,5,6)`
On trouve :
`[0,0]`
et cela prouve que le triangle `(A5B6, B5C6, C5A6)` est équilatéral.
puis par exemple :
`equimorley(5,4,3,2,1,6)`
On trouve :
`[0,0]`
et cela prouve que le triangle `(A5B4, B3C2, C1A6)` est équilatéral.
Avec la fonction `equimore`, on peut donc montrer que les 18 triangles trouvés précédemment sont bien tous équilatéraux.

Chapitre 7

Quelques exemples de géométrie dynamique

7.1 Exemples de problème de maxima-minima

7.1.1 Variation d'une longueur

Soient deux points A et B. Un point M se déplace sur le cercle C de centre B et de rayon 2 : $\overrightarrow{BM} = 2 * \exp(it)$.
On considère la fonction :
$L(A, B, t)$= longueur(AM)
Avec `Xcas` on peut avoir sur le même dessin la construction géomètrique et le graphe de la fonction.
On clique avec la souris pour avoir les points `A` et `B` puis, on exécute la liste des instructions qui se trouve dans `geo12` (faire `Charger session` du menu `Fich` de `Xcas` et selectionner `geo12` du répértoire `examples/geo` pour exécuter ce fichier).
Voici le détail de `geo12` :
`//2pts A et B, calcul de la longueur de AE qd E=B+2*exp(`$i*$
t`)`
`//A:=point(-2);`
`//B:=point(i);`
`C:=cercle(B,2);`
définit le cercle C,
`L(A,B,t):=evalf(longueur(A,B+2*exp(i*t)));`
définit la fonction L,
`D:=plotfunc(L(A,B,x),x);`
dessine le graphe de la longueur(AM) en fonction de x ($M = B + 2 * exp(i * x)$),
`t:=element(0..pi);`
t est un élément que l'on pourra faire varier entre 0 et π
`E:=element(cercle(B,2),t);`
E est un point du cercle qui varie quand t varie,
`F:=element(D,t);`
F est un point du graphe qui varie quand t varie, F a donc comme ordonné la longueur de AE
Lorsqu'on fait bouger le curseur correspondant à t (situé dans la plage grise en haut

et à droite de l'écran géométrique), on fait varier E et F simultanément.

7.1.2 Dimensions du rectangle de périmetre $2p$ et de surface maximum

Une famille de rectangles a pour périmètre $2p$. Trouver les dimensions du (ou des) rectangle(s) d'aire maximum.
On suppose pour faire le dessin avec `Xcas` que $p = 3$.
Les cotés d'un rectangle de la famille sont donc x et $3 - x$ ou encore $3t$ et $3(1 - t)$
L'aire d'un tel rectangle est donc égale à $f(t) = 9t(1 - t)$.
On dessine les différents rectangles et le graphe de la fonction f sur un même graphique pour pouvoir observer la variation des formes des rectangles et la variation de leurs aires.
On exécute la liste des instructions : `A:=point(-3);`
`p:=3;`
p est le demi-périmètre des rectangles,
`K:= point(A+p);`
$\overline{AK} = p = 3$
`t:=element(0..1); B:=A+t*(K-A);`
B sommet du rectangle $\overrightarrow{AB} = t * \overrightarrow{AK}$ donc $\overline{AB} = 3 * t$
`C:=rotation(B,pi/2,K);`
C est un sommet du rectangle
`D:= C+A-B;`
D est l'autre sommet
`segment(A,B);`
`segment(C,B);`
`segment(C,D);`
`segment(A,D);`
ces 4 segments dessinent le rectangle `ABCD`.
`f(x):=9*x*(1-x);`
définit de la fonction f égale à l'aire de `ABCD`.
`G:=plotfunc(f(x),x);`
dessine le graphe de f.
`M:=element(G,t);`
définit M un point du graphe, qui a comme ordonné l'aire du rectangle `ABCD`.
En faisant varier t, le rectangle change de forme et le point M se déplace sur le graphe en ayant pour ordonné l'aire du rectangle dessiné. **Remarque** Il est facile de montrer algébriquement que l'aire est maximum quand le rectangle est un carré, c'est à dire que l'aire maximum vaut $(p/2)^2 = p^2/4$. En effet on a :
l'aire d'un rectangle de côtés a et $p - a$ vaut $(p - a) * a = a * p - a^2$, et on a $p^2/4 \geq a * p - a^2 = (p/2 - a)^2$ car $p^2/4 - a * p + a^2 = (p/2 - a)^2 \geq 0$

7.1.3 Variante du problème précédent : minimiser une surface

Soient un rectangle $ABCD$ de côtés a et b et $c \leq \texttt{min}(a, b)$. Soient $A1$ sur le côté AB tel que $AA1 = c$, $B1$ sur le côté BC tel que $BB1 = c$, $C1$ sur le côté CD tel que $CC1 = c$ et $D1$ sur le côté DA tel que $DD1 = c$. Comment choisir c pour que l'aire du parallélogramme $A1B1C1D1$ soit minimum.

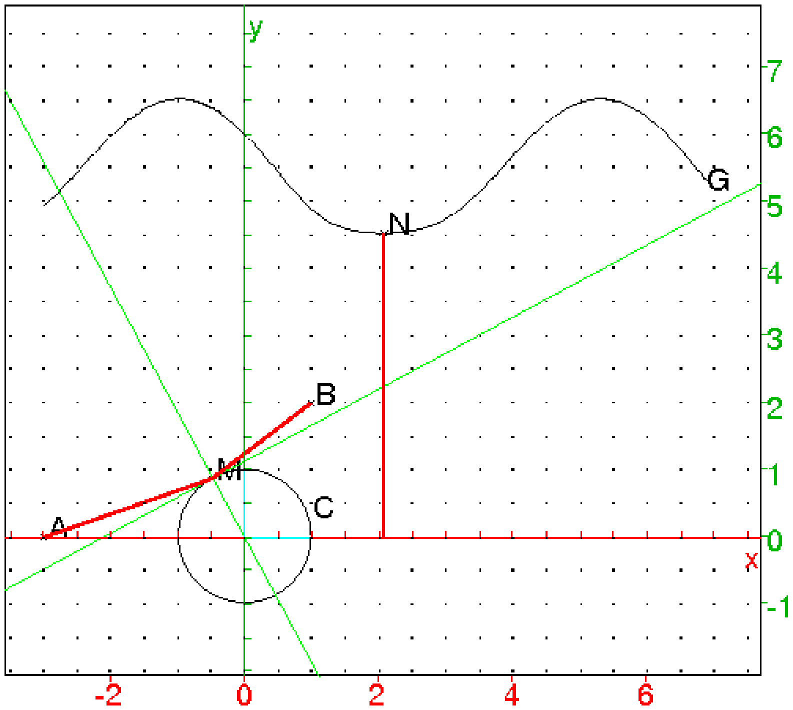
Ce problème se raméne au précédent en effet : la différence entre les aires de $ABCD$ et de $A1B1C1D1$ est l'aire de 4 triangles rectangles égaux deux à deux, c'est donc aussi l'aire de 2 rectangles de côtés c et $a-c$ pour l'un et c et $b-c$ pour l'autre ou encore l'aire d'un rectangle de côtés c et $a+b-2*c$. L'aire de ces 4 triangles rectangles vaut donc $(a+b-2*c)*c$.
On cherche comment choisir c pour que l'aire du rectangle de côtés c et $a+b-2*c$ soit maximum ou ce qui revient au même pour que l'aire du "rectangle double" (de côtés $2*c$ et $a+b-2*c$) soit maximum. Ce rectangle double a pour périmètre $2*p = 2*(a+b)$, donc d'après ce qui précède il faut choisir $2*c = p/2 = (a+b)/2$ c'est à dire $c = (a+b)/4$.

7.1.4 Un trajet difficile : minimiser AMB avec M sur un cercle

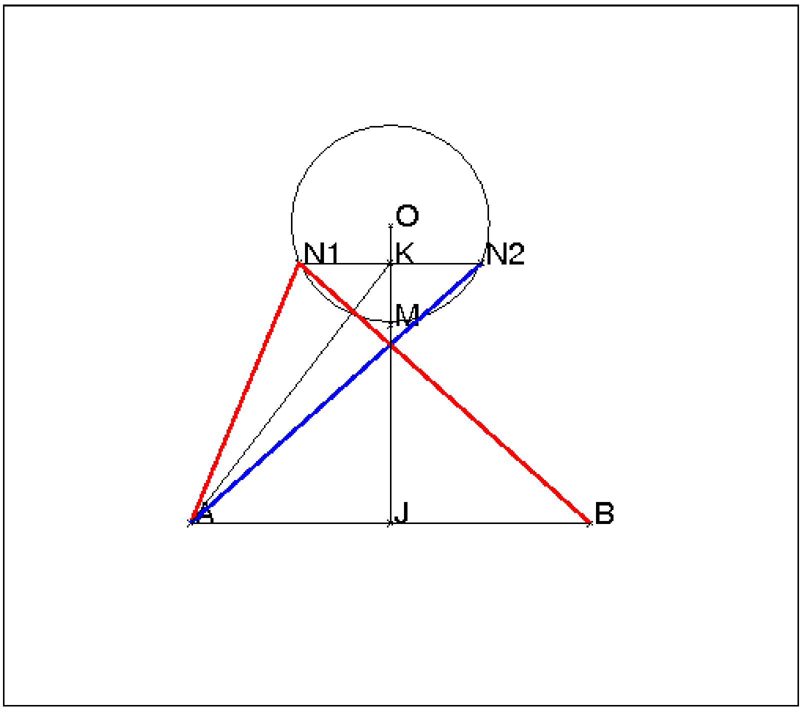
Un point `M` se déplace sur le cercle `C` de centre `O` et de rayon `1`. On choisit deux points `A` et `B` pour que la droite `AB` ne coupe pas le cercle `C`.
On cherche dans ce cas, à minimiser le trajet `AM+MB`.
Avec `Xcas` on va faire apparaître sur le même écran, le dessin géométrique et le graphe `G` de `L:=longueur(AM)+longueur(MB)` : lorsque `M:=exp(i*t)` se déplace sur le cercle `C`, le point `N` de coordonnèes `(t,L)` se déplace sur le graphe `G`.
On tape :

```
A:=point(-3);
B:=point(1+2*i);
C:=cercle(0,1);
t:=element(-3..7);
M:=point(exp(i*t));
L(A,B,t):=evalf(longueur(A,exp(i*t))+longueur(B,exp(i*t)));
G:=plotfunc(L(A,B,t),t=-3..7);
//N:=element(G,t);
N:=point(t,L(A,B,t));
segment(A,M,affichage=1+epaisseur_ligne_2);
segment(B,M,affichage=1+epaisseur_ligne_2);
segment(N,t,affichage=1+epaisseur_ligne_2);
bissectrice(M,A,B);
exbissectrice(M,A,B);
```

Ensuite lorsque l'on fait bouger `t` les points `M` et `N` bougent, l'un sur le cercle `C`, l'autre sur le graphe `G` et l'on peut voir que le minimum est atteint quand une bissectrice intérieure de l'angle `M` passe par `O`.
On peut aussi faire varier `B` pour voir ce qu'il se passe quand la droite `AB` coupe `C` c'est à dire quand la solution est evidente...



Solution dans un cas particulier



On peut démontrer que lorsque le triangle OAB est isocéle de sommet O le point M du cercle C de centre O qui rend le trajet $AM + MB$ minimum se trouve sur la bissectrice intérieure de l'angle $\widehat{AOB}$. En effet soient deux points $N1$ et $N2$ du cercle C symétriques par rapport à cette bissectrice (qui est aussi la médiatrice de AB). On a donc $AN1 = BN2$ et $AN2 = BN1$ et donc

$$AN1 + N1B = AN1 + AN2.$$

Soient K le milieu de $N1N2$ et J le milieu de AB. Les points $0,\ K,\ M,\ J$ sont tous sur la médiatrice de AB et puisque $JK > JM$ (K milieu de la corde $N1N2$ et M milieu de l'arc $N1N2$), on en déduit que :

$$AK > AM$$
$$\overrightarrow{AN1} + \overrightarrow{AN2} = 2\overrightarrow{AK}$$

d'aprés l'inégalité triangulaire on a

$$2AK < AN1 + AN2$$

donc

$$AM + MB = 2AM < 2AK < AN1 + AN2$$

ce qui prouve que $AM + MB$ est minimum.
La figure avec `Xcas` :

```
A:=point(-2-3*i);
B:=point(2-3*i);
```

```
O:=point(0);cercle(0,1);
M:=point(-i);
N1:=point(exp(-7*pi*i/8));
N2:=point(exp(-pi*i/8));
K:=milieu(N1,N2);J:=milieu(A,B);
segment(A,N1,affichage=1+epaisseur_ligne_2);
segment(A,N2,affichage=4+epaisseur_ligne_2);
segment(A,M);
segment(A,K);
segment(N2,N1);
segment(B,N1,affichage=1+epaisseur_ligne_2);
segment(O,J);
segment(A,B);
```

7.1.5 Maximiser une surface

Le problème est le suivant :
Étant donné un rectangle ou un parallélogramme ou un trapèze $OABC$ avec $OA//BC$, on cherche la position de $A1$ sur le segment OA et de $A2$ sur le segment BC pour que l'aire de l'intersection des triangles $BCA1$ et $OAA2$ soit maximum.

— Avec un rectangle,

  — La conjecture
  On appelle $a1$ et $a2$ les abscisses respectives de $A1$ et de $A2$. On conjecture que la réponse est $a1 = a2$.
  On voit cela en exécutant le fichier ci-dessous.
  Pour cela :
  — ouvrir un écran de géométrie (en tapant `Alt+g`)
  — utiliser le menu `Session`▶`Montrer`▶`Montrer la fenetre de script`.

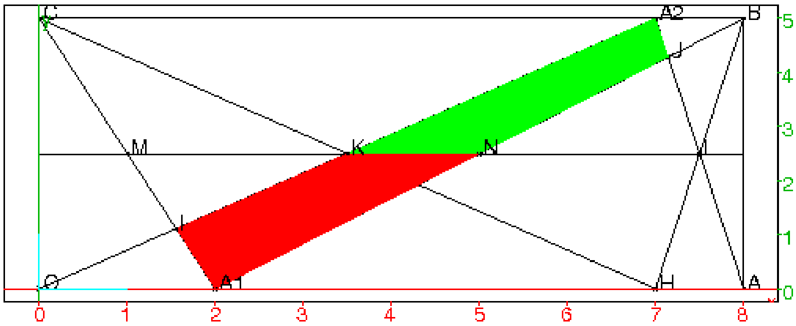
  — recopier le script dans cette fenêtre, avec la souris, depuis un éditeur de programmes ou bien charger le fichier contenant le script avec le menu `Fich`▶`Charger` de cette fenêtre.
  — exécuter le script ligne par ligne, on clique sur la ligne à exécuter, puis, sur la ligne de commandes où on veut que cela s'exécute, puis, on clique sur le bouton `exec` : le script s'exécute alors ligne par ligne. **Attention**
  Il ne faut rien faire entre, cliquer sur la ligne à exécuter et choisir la ligne de commandes. De plus, cette ligne de commandes doit se trouver à la fin de votre session, ou à la fin des lignes de commandes d'un écran graphique ou tortue, pour que de nouvelles lignes soient créées automatiquement au fur et à mesure de l'exécution du script.

```
erase;
xyztrange(-0.5,8.5,-0.5,10.5,-10,10,-1,6,-0.5,8.5,-0.5,10.5,1,
rectangle(0,8,5/8);
assume(a1=1);
```

```
A1:=point(a1,0);
assume(a2=6);
A2:=point(a2,5);
O:=point(0,0);
A:=point(8,0);
B:=point(8,5);
C:=point(0,5);
droite(O,A2);
droite(C,A1);
I:=(inter(droite(O,A2),droite(C,A1)))[0];
droite(A,A2);
droite(B,A1);
J:=(inter(droite(A,A2),droite(B,A1)))[0];
couleur(polygone(A1,J,A2,I),rempli+rouge);
f(a1):=normal(aire(A1,J,A2,I));
plotfunc(f(x),x=0..8);
S:=point(a1,f(a1));
couleur(droite(A1,S),vert);
H:=point(a2,0);
K:=(inter(droite(O,A2),droite(C,H)))[0];
L:=(inter(droite(A,A2),droite(B,H)))[0];
droite(H,C);
droite(H,B);
M:=(inter(droite(C,A1),droite(K,L)))[0];
N:=(inter(droite(B,A1),droite(K,L)))[0];
couleur(polygone(L,H,K),rempli+bleu);
couleur(polygone(A1,M,N),rempli+bleu);
normal(aire(H,K,L)-aire(A1,M,N));
```

On considere au debut $A2$ fixe et on peut faire bouger $A1$ en cliquant sur le petit trait rouge $a1$ situé en haut et à droite de l'écran.
La courbe que l'on voit, est celle de l'aire $A1, I, A2, J$ en fonction de $a1$ (aire = ordonnee de S).
Puis on peut faire bouger $A2$: on a une nouvelle courbe qui est toujours celle de l'aire $A1, I, A2, J$ en fonction de $a1$. On peut taper dans une ligne de commande $f(x)$ pour avoir la fonction aire de parametre $a2$.
On trace ensuite les differents graphes en faisant varier $a2$ de 0 a 8, on les voit ds l'ecran `DispG` en tapant `DispG` et en exécutant : `erase`;
$\texttt{xyztrange}(-0.5, 8.5, -0.5, 10.5, -10, 10, -1, 6, -0.5, 8.5, -0.5, 10.5, 1, 0, 1, 1)$;
$\texttt{f(x)} := \dfrac{20 * x^2 - 160 * x + 20 * a2^2 - 160 * a2}{x^2 + 2 * x * a2 - 16 * x + a2^2 - 16 * a2}$;
`pour a2 de 0 jusque 8 faire nodisp(plotfunc(f(x), x = 0..8)) fpour`;

— Une démonstration
  On conjecture que le maximum est quand $a1 = a2$. On trace alors les 2 surfaces que l'on veut comparer : $A1JA2I$ et $HLA2K$.



On remarque que les triangles $A1MN$, HKL et $A2KL$ ont la même aire. Cette aire est d'ailleurs égale au huitième de l'aire du rectangle puisque ces triangles ont comme base la moitié de la longueur du rectangle et comme hauteur correspondant la moitié de la largeur du rectangle. On trace en rouge le morceau $A1NKO$ et en vert le morceau $A2KNJ$. On suppose que $A1$ se trouve entre O et H (on se ramèneà ce cas en faisant une symétrie par rapport à la droite ML médiane du rectangle). Le point I se trouve alors en dessous de la droite ML et J se trouve alors en dessus de la droite ML : en effet, I est entre O et K donc en dessous de la droite KL et J entre L et $A2$ donc en dessus de la droite KL.
Le morceau rouge $A1NKO$ a donc une aire plus petite que l'aire de $A1MN$ et le morceau vert $A2KNJ$ a donc une aire plus petite que l'aire de $A2KL$.
On voit cela en exécutant le fichier ci-dessous.
Pour cela :

— ouvrir un écran de géométrie (en tapant `Alt+g`)
— utiliser le menu `Session`▶`Montrer`▶`Montrer la fenetre de script`.

— recopier le script dans cette fenêtre, avec la souris, depuis un éditeur de programmes ou bien charger le fichier contenant le script avec le menu `Fich`▶`Charger` de cette fenêtre.
— exécuter le script ligne par ligne, on clique sur la ligne à exécuter, puis, sur la ligne de commandes où on veut que cela s'exécute, puis, on clique sur le bouton `exec` : le script s'exécute alors ligne par ligne. **Attention**
Il ne faut rien faire entre, cliquer sur la ligne à exécuter et choisir la ligne de commandes. De plus, cette ligne de commandes doit se trouver à la fin de votre session, ou à la fin des lignes de commandes d'un écran graphique ou tortue, pour que de nouvelles lignes soient créées automatiquement au fur et à mesure de l'exécution du script.

```
erase;
xyztrange(-0.5,8.5,-0.5,10.5,-10,10,-1,6,-0.5,8.5,-0.5,10.5,1,
rectangle(0,8,5/8);
a1:=element(0..8,2);
```

```
A1:=point(a1,0);
a2:=element(0..8,7);
A2:=point(a2,5);
O:=point(0,0);
A:=point(8,0);
B:=point(8,5);
C:=point(0,5);
segment(O,A2);
segment(C,A1);
I:=(inter(segment(O,A2),segment(C,A1)))[0];
segment(A,A2);
segment(B,A1);
J:=(inter(segment(A,A2),segment(B,A1)))[0];
H:=point(a2,0);
segment(C,H);
segment(B,H);
K:=(inter(segment(O,A2),segment(C,H)))[0];
L:=(inter(segment(A,A2),segment(B,H)))[0];
segment(2.5*i,8+2.5*i);
M:=(inter(segment(C,A1),segment(K,L)))[0];
N:=(inter(segment(B,A1),segment(K,L)))[0];
si (a1<a2) alors [couleur(polygone(A1,N,K,I),rempli+rouge),
                  couleur(polygone(A2,K,N,J),rempli+vert)];
    sinon [couleur(polygone(A2,I,M,L),rempli+vert),
           couleur(polygone(A1,J,L,M),rempli+rouge)];
fsi;
```

— Une autre façon de rédiger la démonstration
  On remarque que :
  — si $a1 < a2$, I est au dessous de la médiane MN et J se trouve au dessus : le triangle $A1MN$ a donc une aire plus grande que l'aire de $A1IKN$ et le triangle $A2LK$ a donc une aire plus grande que l'aire de $A2JNK$. Or quelquesoit la position de $A1$ (resp de $A2$), l'aire du triangle $A1MN$ vaut le quart de l'aire du triangle $A1CB$ ou encore le huitième de l'aire du rectangle (resp le triangle $A2LK$ vaut le huitième de l'aire du rectangle) et donc l'aire de $A1IA2J$ est inférieure à un quart de l'aire du rectangle.
  — si $a1 = a2$,I et J sont sur la médiane et donc l'aire de $A1IA2J$ est égale à un quart de l'aire du rectangle.
  — si $a1 > a2$, on se ramène au cas $a1 < a2$ en faisant une symétrie par rapport à la médiane.

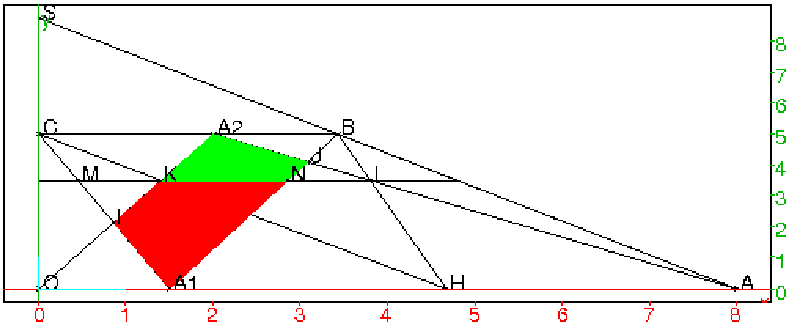
— Avec $OABC$, un parallélogramme,
On obtient le même résultat qu'avec le rectangle.

— Avec $OABC$, un trapèze rectangle,
Soit S l'intersection de OC et de AB.
$SA2$ coupe OA en H.
La paralléle à OA passant par le point de concours des diagonales OB et CA va jouer le même rôle que la médiane du rectangle.
En effet, si $A1$ se trouve en H, les points K intersection de CH et de $A2O$

et L intersection de BH et de $A2A$ se trouve sur cette paralléle qui est la polaire de S par rapport à OA et BC (cf). Si I est l'intersection de $OA2$ avec $CA1$, et si J est l'intersection de $AA2$ avec $BA1$, I et J sont de part et d'autre de cette paralléle et donc l'aire de $A1IA2J$ est inférieure à l'aire de $HKA2L$. On fait la figure en exécutant le script suivant :

```
erase;
xyztrange(-0.5,8.5,-0.5,9,-10,10,-1,6,-0.5,8.5,-0.5,9,1,0,1,1);
polygone(0,8,5*i+24/7,5*i);
a1:=element(0..8,1.5);
A1:=point(a1,0);
a2:=element(0..24/7,2);
A2:=point(a2,5);
O:=point(0,0);
A:=point(8,0);
B:=point(24/7,5);
C:=point(0,5);
segment(O,A2);
segment(C,A1);
I:=(inter(segment(O,A2),segment(C,A1)))[0];
segment(A,A2);
segment(B,A1);
J:=(inter(segment(A,A2),segment(B,A1)))[0];
h:=7/3*a2;
H:=point(h,0);
segment(C,H);
segment(B,H);
S:=(inter(demi_droite(O,C),demi_droite(A,B)))[0];
segment(A,S);
segment(O,S);
K:=(inter(segment(O,A2),segment(C,H)))[0];
L:=(inter(segment(A,A2),segment(B,H)))[0];
segment(3.5*i,4.8+3.5*i);
M:=(inter(segment(C,A1),droite(K,L)))[0];
N:=(inter(segment(B,A1),segment(K,L)))[0];
si (a1<h) alors [couleur(polygone(A1,N,K,I),rempli+rouge),
                 couleur(polygone(A2,K,N,J),rempli+vert)];
   sinon [couleur(polygone(A2,I,M,L),rempli+vert),
          couleur(polygone(A1,J,L,M),rempli+rouge)];
fsi;
```

On obtient :



La surface rouge est inférieure à l'aire de $A1NM$ qui est égale à l'aire de HKL (triangle ayant des bases égales et des hauteurs égales) et, la surface verte est inférieure à l'aire de $A2KL$. Donc, l'aire de $A1IA2J$ est inférieure à l'aire de $A2KHL$.

— Avec $OABC$, un trapèze quelconque,
  On obtient le même résultat qu'avec le trapèze rectangle.
— Avec $OABC$, un quadrilatère quelconque,
  On suppose, quitte à faire des symétries que le quadrilatère $OABC$ est direct et est tel que si S est l'intersection de OA et de BC, S, O, A se trouve dans cet ordre sur OA et si P est l'intersection de OC et de AB, P, B, A se trouve dans cet ordre sur BA.
  Comme précédemment, on choisit $A1$ sur le segment OA et $A2$ sur le segment BC. On appelle H l'intersection de $PA2$ et du segment OA, I l'intersection de $CA1$ et de $OA2$, J l'intersection de $BA1$ et de $AA2$, Soit la figure :

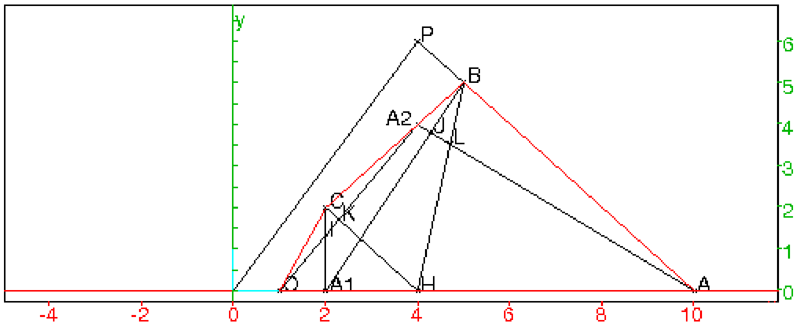
```
O:=point(1,0);
A:=point(10,0);
B:=point(5,5);
C:=point(2,2);
a1:=element(1..10,2);
a2:=element(2..5,4);
A1:=point(a1);
A2:=point(a2*(1+i));
P:=inter_droite(droite(A,B),droite(O,C));
H:=inter_droite(droite(P,A2),droite(O,A));
segment(A,A2);
segment(O,A2);
segment(B,A1);
segment(C,A1);
I:=inter_droite(segment(O,A2),segment(C,A1));
J:=inter_droite(segment(A,A2),segment(B,A1));
segment(C,H);
segment(B,H);
K:=inter_droite(segment(O,A2),segment(C,H));
L:=inter_droite(segment(A,A2),segment(B,H));
s1:=normal(aire(polygone(A1,J,A2,I)));
```

```
s2:=normal(aire(polygone(H,L,A2,K)));
segment(O,P,ligne_tiret);
segment(B,P,ligne_tiret);
quadrilatere(O,A,B,C,affichage=rouge)
```

On trouve :

`s1=42/21` et `s2=576/119`

On obtient la figure :



On peut trouver la valeur de l'aire du polygone $A1JA2I$ lorsque $A2$ et $A1$ varient ($A2$ entre 0 et A et $A1$ entre B et C) . Pour cela on tape :

```
O:=point(1,0);
A:=point(10,0);
B:=point(5,5);
C:=point(2,2);
polygone(O,A,B,C);
assume(a1=[2,1,10]);
assume(a2=[4,1,5]);
A1:=point(a1);
A2:=point(a2*(1+i));
P:=inter_droite(droite(A,B),droite(O,C));
H:=inter_droite(droite(P,A2),droite(O,A));
segment(A,A2);
segment(O,A2);
segment(B,A1);
segment(C,A1);
I:=inter_droite(segment(O,A2),segment(C,A1));
J:=inter_droite(segment(A,A2),segment(B,A1));
f(a1):=normal(aire(A1,J,A2,I));
```

On obtient :

`H=point((-(2*a2))/(a2-6),0)` on a bien si `a2=2` alors H est en O et si `a2=5` alors H est en A.

$$\mathtt{f}(\mathtt{x}) = \frac{9*\mathsf{a2}^3*\mathsf{x}^2+3*\mathsf{a2}^2*\mathsf{x}^3+(-(96*\mathsf{a2}^2))*\mathsf{x}^2+30*\mathsf{a2}^2*\mathsf{x}+90*\mathsf{a2}*\mathsf{x}^2}{2*\mathsf{a2}^2*\mathsf{x}^2+(-(104*\mathsf{a2}))*\mathsf{x}+200}$$

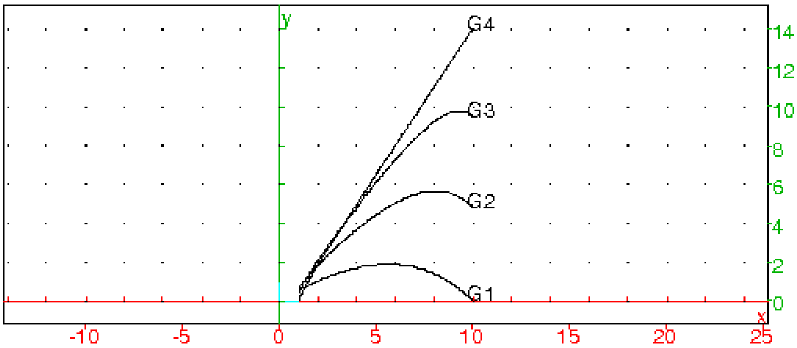
On peut faire les courbes de l'aire du polygone $A1JA2I$ lorsque $A2$ est fixe et lorsque $A1$ varie entre O et A. Pour cela on tape : Pour avoir les graphes de f selon le paramètre $a2$, on tape successivement :

```
a2:=2;G1:=plotfunc(f(x),x=1..10);
....
a2:=5;G4:=plotfunc(f(x),x=1..10);
```

ou bien, on tape :

```
f(x):=(9*a2^3*x^2+3*a2^2*x^3+(-(96*a2^2))*x^2+30*a2^2*x+90*a2*x^2)/(2*a2
L:=[];
for (a2:=2;a2<6;a2++) {L:=append(L,plotfunc(f(x),x=1..10));}
L;
```

On obtient :

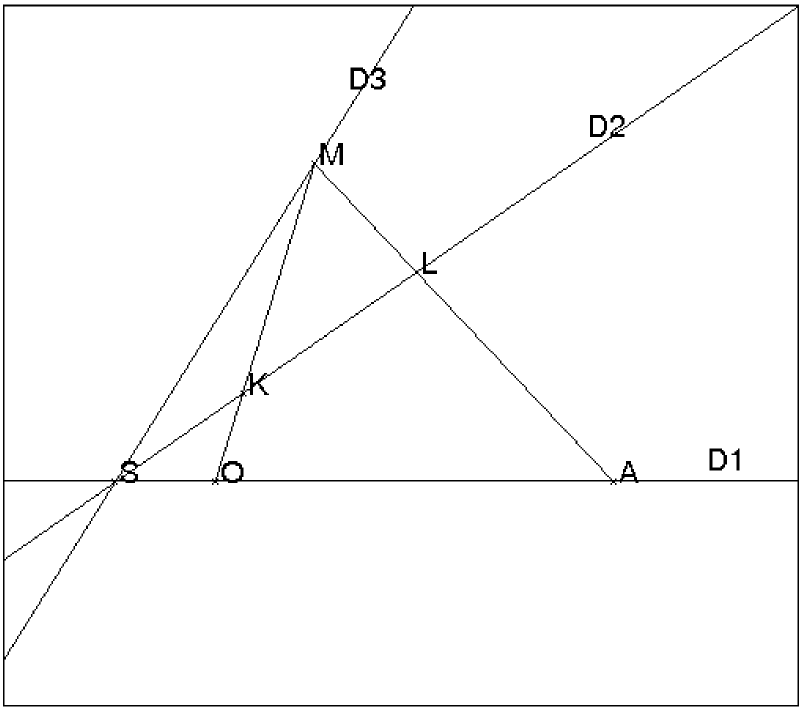


Pour montrer que l'aire de $A1JA2I$ est maximum lorsque $A1$ est en A et $A2$ en B, on va montrer que cette aire croit lorsque $A2$ est fixe et que $A1$ se déplace sur le segment OH de O à H. Puis on va montrer que l'aire de $HLA2K$ croit, lorsque $A2$ se déplace sur le segment CB de C à B. Pour cela, il suffit de demontrer le lemme suivant :

Lemme Soient trois demi-droites $D1, D2, D3$ de même origine S et tels que, $0 < (D1, D2) < (D1, D3) < \pi/2$. Soient deux points fixes O et A sur $D1$ tels que $SO < SA$ et un point variable M sur $D3$. Le segment MO coupe $D2$ en K et le segment MA coupe $D2$ en L. Alors le segment KL augmente lorsque le segment SM augmente.

Donc l'aire du triangle $AMKL$ augmente avec SM puisque KL et la hauteur relative à KL augmentent avec SM.

Voici la figure :



La démonstration du lemme peut se faire avec `Xcas` de façon analytique ou de façon purement géométrique.

— avec `Xcas`

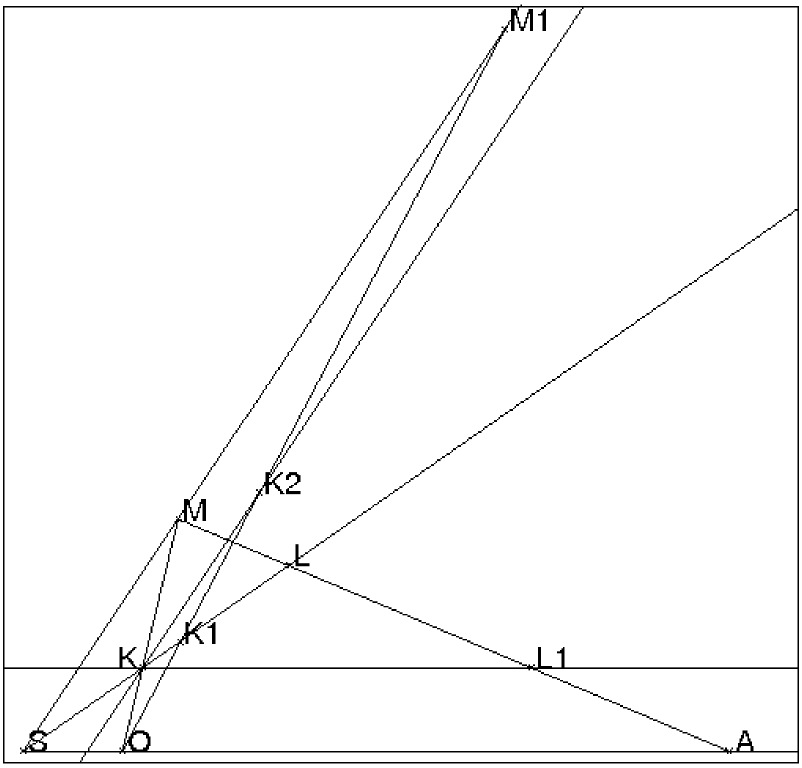
  On se donne 3 points S,O et A alignés et deux droites, puis on fait la construction et on calcule la longueur KL et on montre que cette longueur croit lorsque SM augmente.

```
S:=point(0);
O:=point(1);
assume(a=5);
assume(k=2);
assume(m=3);
assume(t=2);
A:=point(a);
M:=point(t,m*t);
N:=point(2*t,2*k*t):;
D1:=demi_droite(S,A);
D2:=demi_droite(S,N);
D3:=demi_droite(S,M);
K:=inter_droite(droite(M,O),D2);
L:=inter_droite(droite(M,A),D2);
segment(M,O);
segment(M,A);
l(t):=longueur2(K,L);
d(t):=diff(l(t),t);
```

```
factor(numer(d(t)));
factor(denom(d(t)));
```

On trouve comme numérateur :
$(-(2*(k^2+1)*k))*(-m+k)^2*(-m)^2*t^3*$
$(-a*t*k+a*t*m--2*a*k-t*k+t*m)*(a-1)^2$
On trouve comme dénominateur :
$(-((-t*k+t*m+k)^3))*(a*k-t*k+t*m)^3$
Le numérateur et le dénominateur sont tous les deux négatifs puisque l'on suppose $m > k > 0$, $t > 0$ et $a > 0$ donc $d(x)$ est positif. On a donc montré que $l(t)$ croit avec t.

— de façon purement géométrique,
La parallèle à OA passant par K coupe MA en $L1$. On a donc :
$KL1 = OA * MK/MO$ et
en considérant le triangle $LKL1$,on a :
$KL/\sin(\widehat{L}1) = KL/\sin(\widehat{A}) = KL1/\sin(\widehat{L})$.
Le rapport MK/MO augmente avec OM en effet soit une autre position $M1$ de M ($SM < SM1$) et $K1$ l'intersection de $M1O$ avec $D2$.



La parallèle à $D3$ passant par K coupe $M1O$ en $K2$ on a :
$MK/MO = M1K2/M1O < M1K1/M1O$ puisque $M1K2 < M1K1$.
Donc $KL1$ augmente lorsque SM augmente.
L'angle A augmente quand SM augmente et donc l'angle L dimi-

nue puisque l'angle $(D1, D2)$ est fixe. Donc puisque $KL = KL1 * sin(\widehat{A})/sin(\widehat{L})$, on en déduit que KL augmente lorsque SM augmente.

Remarque

On a le même résultat si M se trouve sur $D1$ et si K et L sont les intersections des segments qui joignent M à deux points fixes C et B de $D3$.

On va utiliser ce lemme en prenant pour M soit le point $A2$, soit le point $A1$, en effet :

- si S est l'intersection des droites OA et BC, les points S, K, L sont alignés sur la polaire de P intersection des droites OC et AB. De plus si $A1$ se trouve entre O et H, le point I est en dessous de cette droite et J se trouve au dessus de cette droite. Cela prouve que si $A1$ se trouve entre O et H, l'aire de $A1JA2I$ est inférieure à l'aire de $HLA2K$.
- lorsque $A2$ va de C à B, H va de O à A et l'aire de $A2KLH$ augmente.

7.1.6 Maximiser et minimiser une somme de longueurs

Énoncé Soient un triangle ABC et un point M situé à l'intérieur de ABC. Soient P, Q, R les projections de M sur AB, AC, BC.

Où faut-il placer le point M pour que $MP + MQ + MR$ soit maximum ?

Où faut-il placer le point M pour que $MP + MQ + MR$ soit minimum ?

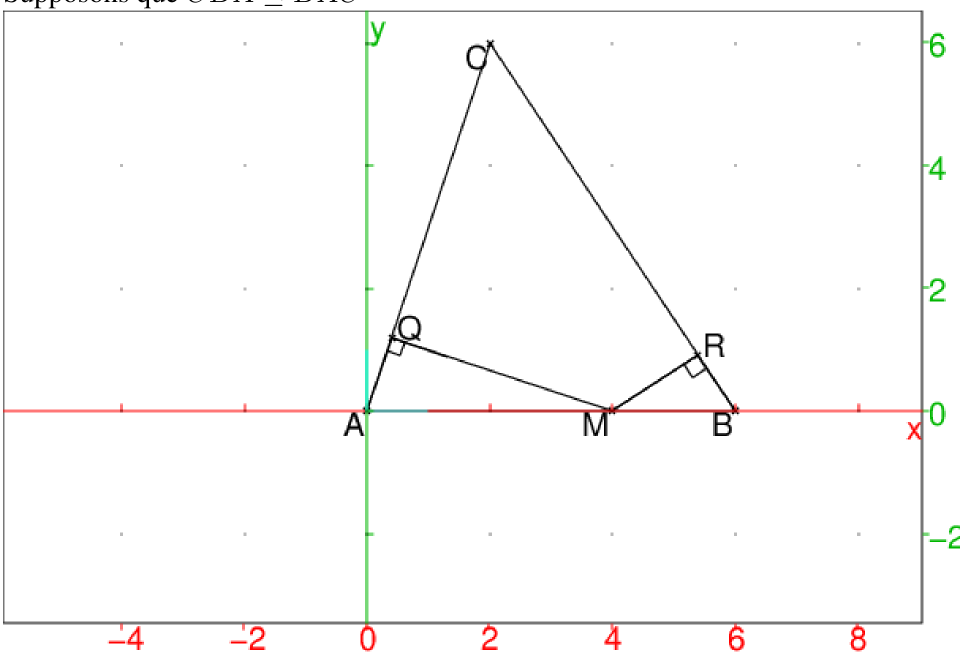
Lemme Soit un triangle ABC et un point M situé sur AB. Soient Q, R les projections de M sur AC et BC.

Où faut-il placer le point M pour que $MQ + MR$ soit maximum ?

Où faut-il placer le point M pour que $MQ + MR$ soit minimum ?

Solution du lemme

— Supposons que $\widehat{CBA} \leq \widehat{BAC}$



On a :

$MQ = AM\sin(\widehat{BAC})$ et $MR = MB\sin(\widehat{CBA})$ et $MB = AB - AM$
donc
$MQ + MR = AM\sin(\widehat{BAC}) + MB\sin(\widehat{CBA}) = AM(\sin(\widehat{BAC}) - \sin(\widehat{CBA})) + AB\sin(\widehat{CBA})$
$AB\sin(\widehat{CBA})$ est une constante qui ne dépend que du triangle ABC.
Il faut donc maximiser et minimiser :
$L = AM(\sin(\widehat{BAC}) - \sin(\widehat{CBA}))$.
Puisque $0 < \widehat{CBA} < \widehat{BAC} < \pi - \widehat{CBA}$, on a $\sin(\widehat{CBA})) < \sin(\widehat{BAC})$
On a donc $L = AM * k$ avec $0 < k = \sin(\widehat{BAC}) - \sin(\widehat{CBA})$ Le maximum de L est atteint lorsque M est en B et
le minimum de L est atteint lorsque M est en A. Donc si l'angle A est plus grand que l'angle B alors le minimum de $MQ+MR$ (égal à $AB\sin(\widehat{CBA})$) est atteint lorsque M est en A et le maximum de $MQ + MR$ (égal à $AB\sin(\widehat{BAC})$) est atteint lorsque M est en B.

— Supposons que $\widehat{CBA} = \widehat{BAC}$
Dans ce cas $MQ+MR = AM\sin(\widehat{BAC})+MB\sin(\widehat{CBA}) = AM(\sin(\widehat{BAC})-\sin(\widehat{CBA})) + AB\sin(\widehat{CBA}) = AB(\sin(\widehat{CBA}))$
Donc si l'angle A est égale à l'angle B alors $MQ + MR$ est constant.

Solution de l'énoncé Supposons que :
$\widehat{ACB} \le \widehat{CBA} \le \widehat{BAC}$
D'après le lemme, lorsque M décrit un segment A_1B_1 parallèle à AB on a MP=cste et
puisque $\widehat{CBA} = \widehat{CB_1A} \le \widehat{BA_1C} = \widehat{BAC}$, $MQ+MR$ est maximum lorsque M est en B_1
$MQ + MR$ est minimum lorsque M est en A_1
Puisque $\widehat{ACB} \le \widehat{CBA}$ d'après le lemme, lorsque M décrit le segment BC, $MP + MQ + MR$ est maximum lorsque M est en C.
Puisque $\widehat{ACB} \ge \widehat{ACB}$ d'après le lemme, lorsque M décrit le segment AC, $MP + MQ + MR$ est minimum lorsque M est en A.

Remarques

1. Si ABC est un triangle équilatéral, on a :
   $\widehat{BAC} = \widehat{CBA} = \widehat{ACB}$
   $(MP+MQ+MR)*AB$ =2*aire(PAB)+2*aire(PAC)+2*aire(PBC)=2*aire(ABC)=$AB^2\frac{\sqrt{3}}{2}$
   donc $MP + MQ + MR = AB\frac{\sqrt{3}}{2}$=cste
2. Si ABC est un triangle isocèle de sommet A et que l'on a
   $\widehat{ACB} = \widehat{CBA} \le \widehat{BAC}$ alors
   $MP + MQ + MR$ est minimum lorsque M est en A et d'après le lemme
   $MP + MQ + MR$ est maximum lorsque M est sur le segment BC.
3. Si ABC est un triangle isocèle de sommet C et que l'on a :
   $\widehat{ACB} \le \widehat{CBA} = \widehat{BAC}$ alors
   d'après le lemme $MP+MQ+MR$ est minimum lorsque M est le segment AB et
   $MP + MQ + MR$ est maximum lorsque M est en C.

Autre méthode On note $a = BC$, $b = AC$, $c = AB$ et S=2*aire(ABC).
On remarque que :

cMP+bMQ+aMR =2*aire(PAB)+2*aire(PAC)+2*aire(PBC)=2*aire(ABC) $=$ S Donc on peut exprimer MR en fonction de MP et de MQ ou exprimer MP en fonction de MR et de MQ :
$MR = \frac{1}{a}S - (\frac{c}{a}MP + \frac{b}{a}MQ)$
$MP = \frac{1}{c}S - (\frac{a}{c}MR + \frac{b}{c}MQ)$
Supposons que : $\widehat{ACB} \leq \widehat{CBA} \leq \widehat{BAC}$ i.e.
$c \leq b \leq a$ car $b\sin(\widehat{BAC}) = a\sin(\widehat{CBA})$ et $c\sin(\widehat{BAC}) = a\sin(\widehat{ACB})$ et donc :
$\sin(\widehat{BAC})/a = \sin(\widehat{CBA})/b = \sin(\widehat{ACB})/c$ De plus si $0 < \widehat{CBA} \leq \widehat{BAC} < \pi - \widehat{CBA}$, on a $\sin(\widehat{CBA})) \leq \sin(\widehat{BAC})$
Donc :
$MP+MQ+MR = \frac{1}{a}S+(1-\frac{c}{a})MP+(1-\frac{b}{a})MQ$ avec $1-\frac{c}{a} > 0$ et $1-\frac{b}{a} > 0$.
$\frac{1}{a}S$ est une constante qui ne dépend que du triangle ABC.
Donc $MP + MQ + MR$ sera minimum lorsque $L = (1 - \frac{c}{a})MP + (1 - \frac{b}{a})MQ$ sera minimum.
$L = (1 - \frac{c}{a})MP + (1 - \frac{b}{a})MQ$ est minimum lorsque $MP = 0$ et $MQ = 0$ c'est à dire lorsque M se trouve sur AB et sur AC donc lorsque M est en A.
et on a aussi :
$MP+MQ+MR = \frac{1}{c}S+(1-\frac{a}{c})MR+(1-\frac{b}{c})MQ$ avec $1-\frac{a}{c} < 0$ et $1-\frac{b}{c} < 0$ et $\frac{1}{a}S$ est une constante qui ne dépend que du triangle ABC.
Donc MP+MQ+MR sera maximum lorsque $L = (1-\frac{a}{c})MR+(1-\frac{b}{c})MQ \geq 0$ sera nul.
$L = (1 - \frac{a}{c})MR + (1 - \frac{b}{c})MQ$ est nul lorsque $MR = 0$ et $MQ = 0$ c'est à dire lorsque M se trouve sur BC et sur AC donc lorsque M est en C.
On retrouve aussi le même résultat pour les cas particuliers : ABC équilatéral ou isocèle.

7.2 Les courbes de Bézier et le barycentre

7.2.1 Courbe de Bézier définie par 3 points

Etant donné 3 points $A,\ B,\ C$ la courbe de Bézier qui passe par A et C en étant tangente à AB et à BC a pour équation paramétrique :
$A(1-x)^2 + 2Bx(1-x) + Cx^2$ pour $x \in [0; 1]$.
On peut donc définir la fonction :
`bezier3(A,B,C,x) := {evalf(A*(1-x)^2+2*B*x*(1-x)+C*x^2);};`
La représentation de cette équation paramétrique se fait en utilisant la fonction `plotparam` qui permet de représenter des courbes en paramétrique.
On écrit par exemple :
```
courb(A,B,C):={plotparam(affixe(bezier3(A,B,C,x)),x,0,1);};
```

7.2.2 Courbe de Bézier pour une liste de points

On suppose que l'on met dans une liste une suite de points par exemple :
`L:=[A,B,C,D,E,F,G]` et on veut tracer les courbes de Bézier définies par `A, B, C` puis, par `C, D, E` puis, par `E, F, G`.
On peut donc définir la fonction :

```
bezierl(L,x):={
  local LS,A,B,C;
  LS:=[];
  for(j:=0;j<size(L)-2;j:=j+2){
    A:=L[j];B:=L[j+1];C:=L[j+2];
    LS:=append(LS,affixe(evalf(A*(1-x)^2+2*B*x*(1-x)+C*x^2)));
  };
  eval(LS);
};
```

Pour représenter cette équation paramétrique on écrit :

```
courbl(L):={
  local LB,LS;
  LS:=[];
  LB:=bezierl(L,x);
  for (j:=0;j<size(LB);j:=j+1) {
  LS:=concat(LS,plotparam(LB[j],x,0,1));
  };
  return(feuille(LS));
};
```

Puis, on clique dans l'écran graphique pour obtenir par exemple les points `A, B, C, D, E, F, G` et on tape :
`courbl([A,B,C,D,E,F,G])`, on obtient trois courbes de Bézier successives (passant par `A,C,E,G`) et que l'on peut déformer en déplacant l'un des points `A, B, C, D, E, F, G`.
Attention la liste `L` doit avoir un nombre impair de points car sinon le dernier point n'est pas pris en compte...
Si on veut obtenir une courbe fermée il faut terminer la liste par le premier élément de la liste.

7.2.3 La commande `bezier`

On peut utiliser la commande `bezier` de `Xcas` qui existe maintenant.
Soient $n+1$ points P_j de contrôle ($j=0..n$) et L la séquence de ces points.
La courbe de Bézier ayant les points de la séquence `L` comme points de contrôle, a comme équation paramétrique :
$\sum_{j=0}^{n} comb(n,j)t^j(1-t)^{n-j} * L[j]$.
`bezier(L,plot)` renvoie le tracé de la courbe d'équation paramétrique : $\sum_{j=0}^{n} comb(n,j)t^j(1-t)^{n-j} * L[j]$.
`parameq(bezier(L))` renvoie l'équation paramétrique de la courbe de Bézier ayant comme points de contrôle les points de la séquence `L`. On tape :

```
bezier(1,1+i,2+i,3-i,plot)
```

On obtient :

```
Le tracé de la courbe de Bézier ayant comme points de
      contrôle les points d'affixe 1,1+i,2+i,3-i
```

On tape :

```
parameq(bezier(1,1+i,2+i,3-i))
```

On obtient :

```
L'équation paramétrique de la courbe précédente
```

On tape :

```
bezier(point([0,0,0]),point([1,1,0]),point([0,1,1]),plot)
```

On obtient :

```
Le tracé de la courbe de Bézier ayant comme points de
contrôle les points
point([0,0,0]),point([1,1,0]),point([0,1,1])
```

On tape :

```
parameq(bezier(point([0,0,0]),point([1,1,0]),point([0,1,1])))
```

On obtient :

```
L'équation paramétrique de la courbe précédente
```

7.2.4 Morphing

Une application amusante du barycentre est le morphing.
Prenons tout d'abord un exemple simple :
On considère six points $A,\ B,\ C,\ D,\ E,\ F$, la courbe de Bézier $C1$ définie par $A,\ B,\ C$ et la courbe de Bézier $C2$ définie par $D,\ E,\ F$.
Soient $t \in [0;1]$ et M (resp $N,\ P$) le barycentre de $(A, 1-t)$ et de (D, t) (resp de $(B, 1-t)$ et de (E, t), de $(C, 1-t)$ et de (F, t)).
Lorsque t varie de 0 à 1 la courbe de Bézier définie par $M,\ N,\ P$ se déforme en passant de $C1$ à $C2$.
On écrit la fonction `baryc` égale à la courbe intermédiaire de paramètre t :

```
baryc(A1,B1,C1,A2,B2,C2,t):={
  local M1,M2,M3;
  M1:=bary(A1,A2,t);
  M2:=bary(B1,B2,t);
  M3:=bary(C1,C2,t);
  return(courb(M1,M2,M3));
};
```

puis, les instructions permettant de faire bouger la courbe intermédiaire :

```
courb(A,B,C);
courb(D,E,F);
t:=element(0..1);
baryc(A,B,C,D,E,F,t);
```

Prenons maintenant l'exemple où deux courbes de Bézier sont définies par des listes de points `L1` et `L2` de même longueur. On définit la liste `L3` obtenue en formant le barycentre des points de `L1` avec ceux de `L2` affectés des coefficients `1-t` et `t` ($0 \leq t \leq 1$).
On représente la courbe de Bézier définie par la liste `L3` en faisant varier `t` cette courbe se déforme et passe de la courbe définie par `L1` à celle définie par `L2` lorsque `t` varie de `0` à `1`.
Voici par exemple le programme :

```
baryl(L1,L2,t):={
  local L3,s1,s2;
  s1:=size(L1);
  s2:=size(L2);
  if (s1 !=s2) {s1:=min(s1,s2)};
  L3:=[];
  for (k:=0;k<s1;k++) {
  L3:=append(L3,evalf(L1[k]*(1-t)+L2[k]*t));
  }
  return(eval(L3));
};
```

ou plus simplement
`barycl(L1,L2,t):=evalf(t*L2+(1-t)*L1)`
Par exemple, on clique avec la souris les points :
`A,B,C,D,E,F,G,H,J,K,L,M,N,O,P`
Par exemple vous pouvez utiliser les points qui se trouvent dans le fichier `pointmorph` puis on exécute un fichier contenant les commandes (c'est le fichier `morphing`) :

```
t:=element(0..1);
courbl([A,B,C,D,E,F,G,H,A]);
courbl([J,K,L,M,N,O,L,P,J]);
courbl(baryl([A,B,C,D,E,F,G,H,A],[J,K,L,M,N,O,L,P,J],t));
```

ou encore

```
courbl([A,B,C,D,E,F,G,H,A]);
courbl([J,K,L,M,N,O,L,P,J]);
t:=element(0..1);
baryc(A,B,C,J,K,L,t);
baryc(C,D,E,L,M,N,t);
baryc(E,F,G,N,O,L,t);
baryc(G,H,A,L,P,J,t);
```

On peut ensuite s'amuser à changer la valeur de `t` et aussi déplacer les différents points. On pourra se référer aux fichiers `bezier` qui contient les fonctions qui suivent :

```
// -*- mode:C++ -*-
//fonction f(0)=A f(1)=C f'(0)=AB f'(1)=BC qui renvoie 1 pt
bezier3(A,B,C,x):={
  evalf(A*(1-x)^2+2*B*x*(1-x)+C*x^2);
};

//dessin de la courbe en parametrique passant par A et C
//et tgte a AB et a BC
courb(A,B,C):={plotparam(affixe(bezier3(A,B,C,x)),x,0,1);};

//fonction donnant le barycentre de (A1,1-t) et de (A2, t)
bary(A1,A2,t):={evalf(t*A2+(1-t)*A1);};

//dessin de la courbe barycentre de (courb(A1,B1,C1),1-t)
//et de (courb(A2,B2,C2),t)
//on place les 6 pts puis on definit t:=element(0..1) on peut voir
//la deformation de la courbe qd on fait varier t.
baryc(A1,B1,C1,A2,B2,C2,t):={
  local M1,M2,M3;
  M1:=bary(A1,A2,t);
  M2:=bary(B1,B2,t);
  M3:=bary(C1,C2,t);
  courb(M1,M2,M3);
};

baryl(L1,L2,t):={
  local L3,s1,s2;
  s1:=size(L1);
  s2:=size(L2);
  if (s1 !=s2) return("erreur");
  L3:=[];
  for (k:=0;k<s1;k++) {
  L3:=append(L3,bary(L1[k],L2[k],t));
  }
  return(eval(L3));
};
barycl(L1,L2,t):=evalf(t*L2+(1-t)*L1);
bezierl(L,x):={
  local LS,A,B,C;
  LS:=[];
  for(j:=0;j<size(L)-2;j:=j+2){
  A:=L[j];B:=L[j+1];C:=L[j+2];
  LS:=append(LS,affixe(evalf(A*(1-x)^2+2*B*x*(1-x)+C*x^2)));
  };
  eval(LS);
};

courbl(L):={
```

```
  local LB,LS;
  LS:=[];
  LB:=bezierl(L,x);
  for (j:=0;j<size(LB);j:=j+1) {
  LS:=append(LS,plotparam(LB[j],x,0,1));
  };
  return(feuille(LS));
};
```

7.3 Enveloppe de droites

Commençons par un exemple :
On cherche l'enveloppe des droites définies par $y - 2tx - t^2$ lorsque t varie.
On va tracer le lieu des points d'intersection `M` des droites `ed` d'équation $y - 2tx - t^2 = 0$ et des droites `ed1` d'équation $-2x - 2t = 0$ obtenue en dérivant $y - 2tx - t^2 = 0$ par rapport à t. Les instructions suivantes tracent les droites `ed` et `ed1` de paramètre t et le lieu de `M`.
Ces instructions se trouvent dans le fichier `envelopp`.

```
ed:=y-2*t*x-t^2;
ed1:=derive(ed,t);
M:=solve([ed,ed1],[x,y])[0];
plotparam(M[0]+i*M[1],t);
t:=element(-3..3);
d:=plotfunc(2*t*x-t^2,x);
```

On peut aussi définir la droite d'équation $ay + bx + c = 0$ par la liste $[a, b, c]$.
On traite alors l'exemple avec les instructions ci-dessous (elles se trouvent dans le fichier `envelopl`).

```
//ld=[a,b,c] si ay+bx+c=0
xyztrange(-6,6,-7,4,-10,10,-3,3,-6,6,-5,1,1);
purge(t);
ld:=[1,-2*t,-t^2];
a:=ld[0]:
b:=ld[1];
c:=ld[2];
ld1:=derive(ld,t);
dpd1:=a*ld1[1]-b*ld1[0];
//dpd1=-2 <>0 donc ici ld et ld1 ne sont pas paralleles
M:=(i*(-c*ld1[1]+b*ld1[2])+(c*ld1[0]-a*ld1[2]))/dpd1;
plotparam(M,t);
t:=element(-3..3);
d:=plotfunc(2*t*x+t^2,x);
```

On écrit maintenant deux fonctions (`enveloppe3` et `enveloppe`) qui tracent l'enveloppe d'une famille de droites.
La fonction `enveloppe3` a trois paramètres `a,b,c` qui sont des fonctions de la variable t et qui représente les droites d'équation `ay+bx+c=0`.

```
//enveloppe d'une droite def par a(t),b(t),c(t) (ay+bx+c=0)
enveloppe3(a,b,c):={
  local ld,ld1,dd1,M;
  ld:=[a,b,c];
  ld1:=derive(ld,t);
  dd1:=ld[0]*ld1[1]-ld[1]*ld1[0];
  if (dd1!=0) {
     M:=(i*(-ld[2]*ld1[1]+ld[1]*ld1[2])+
          (ld[2]*ld1[0]-ld[0]*ld1[2]))/dd1;
     return(plotparam(M,t));
  } else {
    return("droites paralleles");
  }
}
```

La fonction `enveloppe` a un paramètre `d` qui est :
`d` est l'expression `a(t)y+b(t)x+c(t)` (on sous-endend $= 0$ et les variables doivent être `x,y,t`).

```
enveloppe(d):={
  local zM,a,b,c,a1,b1,c1,dpd1;
  a:=derive(d,y);
  b:=derive(d,x);
  c:=subst(subst(d,x=0),y=0);
  a1:=derive(a,t);b1:=derive(b,t);c1:=derive(c,t);
  dpd1:=a*b1-b*a1;
  if (dpd1!=0) {
     zM:=(i*(-c*b1+b*c1)+(c*a1-a*c1))/dpd1;
     return(plotparam(zM,t));
  }
  else
    return("Droites paralleles");
};
```

On peut alors écrire les fichiers `envelopt` et `envelop3t` pour avoir une figure animée : l'enveloppe `E` et les différentes droites qui bougent selon les valeurs de `t` en restant tangentes à `E`.
Voici le fichier `envelop3t` qui trace l'enveloppe des droites :
$\cos(t) * (1 - \cos(2t))y + \sin(t) * \cos(2t)x = \sin(t)\cos(t)$:

```
purge(t);
purge(tt);
purge(x);
purge(y);
xyztrange(-6,6,-7,4,-10,10,-3,3,-6,6,-2,4,1);
a:=cos(t)*(1-cos(2*t));
b:=sin(t)*cos(2*t);
c:=-sin(t)*cos(t);
enveloppe3(a,b,c);
tt:=element(-3..4);
```

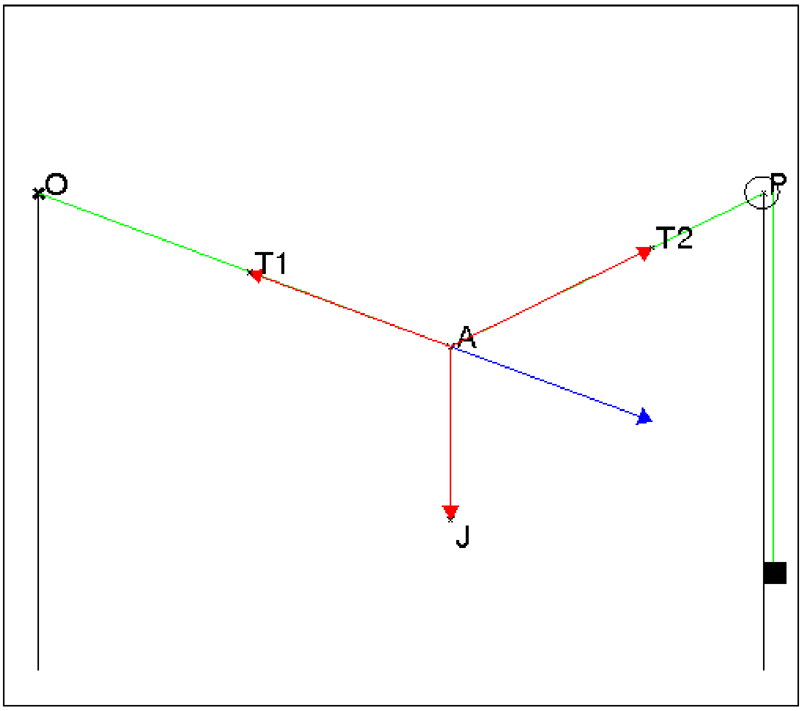
```
aa:=subst(a,t,tt);
bb:=subst(b,t,tt);
cc:=subst(c,t,tt);
plotfunc((-bb*x-cc)/aa,x);
```

Voici le fichier `envelopt` qui trace l'enveloppe des droites $x*(cos(2*t)-cos(t))+y*(sin(2*t)-sin(t))-sin(2*t)=0$:

```
xyztrange(-6,6,-7,4,-10,10,-3,3,-6,6,-2,4,1);
purge(x);
purge(y);
purge(t);
purge(tt);
d:=x*(cos(2*t)-cos(t))+y*(sin(2*t)-sin(t))-sin(2*t);
enveloppe(d);
tt:=element(-3..4);
dd:=subst(d,t,tt);
aa:=derive(dd,y);
bb:=derive(dd,x);
cc:=subst(subst(dd,x=0),y=0);
plotfunc((-bb*x-cc)/aa,x);
```

7.4 Le pantalon

Un "jean" de poids j est accroché en A à un étendage spécial :
l'un des poteaux de l'étendage possède une poulie P !
Le fil de l'étendage passe sur la poulie et est accroché à l'autre poteau en O. On suppose que la masse du fil est négligeable.
Quel poids faut-il mettre au bout du fil pour avoir un équilibre ?



Dans ce qui suit on suppose que :
- le "jean" pèse j unités et on note $\overrightarrow{J} = (0, -j)$,
- les deux poteaux sont distants de l,
- on choisit le repère Oxy pour que la poulie P ait comme coordonnées $(0, l)$,
- le "jean" est fixé au point A de coordonnées (a, b) dans le repère Oxy,
- le poids est de p unités.

Quelles sont alors les positions d'équilibre du "jean" ?
Si $\overrightarrow{T1} = (t11, t12)$ est la tension du fil AP et
si $\overrightarrow{T2} = (t21, t22)$ est la tension du fil AO, on a :
$\overrightarrow{T1} + \overrightarrow{T2} + \overrightarrow{J} = 0$ et $p^2 = ||T1||^2$ donc :
$t11 + t21 = 0$, $t12 + t22 - j = 0$ et $p^2 = t11^2 + t12^2$.
$\overrightarrow{T1}$ est dirigé selon AP donc $-b/(l-a) = t12/t11$,
$\overrightarrow{T2}$ est dirigé selon AO donc $-b/(-a) = t22/t21$.
On remarquera que le problème n'est pas le même selon que le pantalon coulisse sur le fil ou qu'il est fixé sur le fil par une pince à linge :

- si le pantalon coulisse sur le fil les deux tensions $\overrightarrow{T1}$ et $\overrightarrow{T2}$ ont même module.

On tape :
$\texttt{solve}([\texttt{t11} + \texttt{t21} = 0, \texttt{t12} + \texttt{t22} - \texttt{j} = 0, \texttt{t11}^2 + \texttt{t12}^2 = \texttt{t21}^2 + \texttt{t22}^2],$
$[\texttt{t11}, \texttt{t12}, \texttt{t21}, \texttt{t22}])$
On obtient :
$[[-\texttt{t21}, \texttt{j}/2, \texttt{t21}, \texttt{j}/2]]$
ce qui signifie que $\overrightarrow{T1}$ et $\overrightarrow{T2}$ sont symétriques par rapport à la verticale et donc que $OA = AP$.

On tape pour déterminer le poids p en fonction de j, l, a, b :
$\mathtt{solve([t11+t21=0, t12+t22-j=0, t11^2+t12^2=t21^2+t22^2,}$
$\mathtt{b/a=t22/t21, p^2=t11^2+t12^2], [p, t11, t12, t21, t22])}$
On obtient :
$\mathtt{[[sqrt(4*b^4*j^2+4*b^2*j^2*a^2)/(4*b^2), (-(j*a))/(2*b), j/2, j*a/(2*b), j/2],}$
$\mathtt{[(-(sqrt(4*b^4*j^2+4*b^2*j^2*a^2)))/(4*b^2), (-(j*a))/(2*b), j/2, j*a/(2*b), j/2]]}$

- si le pantalon est fixé en A (par exemple au moyen d'une pince à linge), il faut supprimer l'équation $\mathtt{t11^2+t12^2=t21^2+t22^2}$ et rajouter : $\mathtt{-b/(l-a)=t12/t11}$.

On tape :
$\mathtt{normal(solve([t11+t21=0, t12+t22-j=0, b/a=t22/t21,}$
$\mathtt{-b/(l-a)=t12/t11, t11^2+t12^2=p^2], [p, t11, t12, t21, t22])[1][0])}$
On obtient :
$\mathtt{(sqrt(l^2+-2*l*a+a^2+b^2)*abs(l)*abs(b)*abs(j)*abs(a))/(l^2*b^2)}$
On a donc : $p^2=j^2*a^2*((l-a)^2+b^2)/(l^2*b^2)$,
d'où $b^2=j^2*a^2*(-l+a)^2/(p^2*l^2-j^2*a^2)$.
L'ordonnée de A est négative, l'équation de la courbe d'équilibre est :
$\mathtt{b := -sqrt((a\hat{}2*(-l+a)\hat{}2*j\hat{}2)/(p\hat{}2*l\hat{}2-j\hat{}2*a\hat{}2))};.$
Pour simuler la situation on écrit les instructions suivantes dans le fichier `pantalon`.
Ces instructions permettent de faire varier les paramètres j, p et a.
Voici le fichier `pantalon`.

```
switch_axes(0);
xyztrange(-1,10,-7,1,-10,10,-1,6,-1,10,-7,1,0);
p:=element(0..5);
h:=2;//hauteur des poteaux
l:=8;//distance entre les poteaux
f:=l+5;//longueur du fil
j:=element(1..3);//poids du jean
segment(l-h*i,l);P;=point(l,0);//dessin d'un poteau
segment(-h*i,0);O:=point(0,0);//dessin de l'autre poteau
y:=-sqrt((x^2*(-l+x)^2*j^2)/(p^2*l^2-j^2*x^2));
//m:=min(l,p*l/j);
//plotfunc(y,x,0,m);
a:=element(0..l);
b:=-sqrt((a^2*(-l+a)^2*j^2)/(p^2*l^2-j^2*a^2));
A:=point(a,b);
couleur(segment(A,0),2);
couleur(segment(A,l),2);
ap:=sqrt((l-a)^2+b^2);
ao:=sqrt(a^2+b^2);
c:=ao+ap;
//dessin des forces
couleur(vecteur(A,a+(b-j)*i),1);
couleur(vecteur(A,A+(l-a-b*i)*p/sqrt((-l+a)^2+b^2)),1);
couleur(vecteur(A,A+(-l+a+b*i)*p/sqrt((-l+a)^2+b^2)+j*i),1);
couleur(vecteur(A,A+(l-a-b*i)*p/sqrt((-l+a)^2+b^2)-j*i),4);
T1:=A+(l-a-b*i)*p/sqrt((-l+a)^2+b^2);
T2:=A+(-l+a+b*i)*p/sqrt((-l+a)^2+b^2)+j*i;
J:=A+(-j)*i;
```

```
//dessin de la poulie
cercle(l,0.2);
//dessin du poids p
if (c<f){
[couleur(segment(l+0.1,l+0.1+i*(c-f)),2),
couleur( segment(l+i*(c-f),l+i*(c-f-p/10)),1),
couleur( segment(l+0.2+i*(c-f),l+0.2+i*(c-f-p/10)),1),
couleur( segment(l+0.2+i*(c-f),l+i*(c-f)),1),
couleur( segment(l+0.2+i*(c-f-p/10),l+i*(c-f-p/10)),1)];
};
```

On peut aussi utiliser le tableur pour avoir des valeurs numériques.
On a $p^2 = j^2 * x^2 * ((-l+x)^2 + y^2)/(l^2 * y^2)$.
On définit la fonction g (égale à p par :
$\mathtt{g}(\mathtt{j},\mathtt{l},\mathtt{x},\mathtt{y}) = \mathtt{j} * \mathtt{x} * \mathtt{sqrt}((-\mathtt{l}+\mathtt{x})^2 + \mathtt{y}^2)/(\mathtt{l} * \mathtt{y})$
Ne pas oublier auparavant de purger les variables j, l, x, y !
Puis on ouvre le tableur (on tape `Alt+t` pour ouvrir le tableur).
On tape par exemple :
`table g(3.0,8,4,y),y)` et on complète la table....
On peut recopier la deuxième colonne : on sélectionne la deuxième colonne puis on se place en début de colonne et on clique sur `coller`.
Si on veut voir l'influence de `l` on met comme formule dans la troisième case :
`g(3.0,4,2,y),y)` et on change la formule (=`eval(subst..)` comme ci-dessous...
On obtient :

	A	B	C
0	y	g(3,8,4,y)	g(3,4,2,y)
1	0.1	"Table"	0
2	0	=eval(subst(B$0,$A$0,$A2))	=eval(subst(C$0,$A$0,$A2))
3	=A2+A$1	=eval(subst(B$0,$A$0,$A3))	=eval(subst(C$0,$A$0,$A3))
4	=A3+A$1	=eval(subst(B$0,$A$0,$A4))	=eval(subst(C$0,$A$0,$A4))
5	=A4+A$1	=eval(subst(B$0,$A$0,$A5))	=eval(subst(C$0,$A$0,$A5))
6	=A5+A$1	=eval(subst(B$0,$A$0,$A6))	=eval(subst(C$0,$A$0,$A6))
7	=A6+A$1	=eval(subst(B$0,$A$0,$A7))	=eval(subst(C$0,$A$0,$A7))
8	=A7+A$1	=eval(subst(B$0,$A$0,$A8))	=eval(subst(C$0,$A$0,$A8))
9	=A8+A$1	=eval(subst(B$0,$A$0,$A9))	=eval(subst(C$0,$A$0,$A9))

=

	A	B	C
0	y	$\frac{12\sqrt{16+y^2}}{8y}$	$\frac{6\sqrt{4+y^2}}{4y}$
1	0.1	Table	0
2	0	∞	∞
3	0.1	60.0187470712	30.0374765918
4	0.2	30.0374765918	15.0748134317
5	0.3	20.0561711201	10.1118742081
6	0.4	15.0748134317	7.64852927039
7	0.5	12.0933866224	6.18465843843
8	0.6	10.1118742081	5.22015325446
9	0.7	8.70168878753	4.54063287866

On voit l'influence de la longueur du fil : si $j = 3$, pour avoir A au point de coordonées $(l/2, -0.1))$, il faut avoir $p = 60$ si $l = 8$ et $p = 30$ si $l = 4$.

On définit la fonction `equi` qui est l'ordonnée de A par :
$\mathtt{equi}(\mathtt{j},\mathtt{l},\mathtt{p},\mathtt{x}) := -\mathtt{sqrt}(\mathtt{j}^2 * \mathtt{x}^2 * (\mathtt{x}-\mathtt{l})^2/(\mathtt{p}^2 * \mathtt{l}^2 - \mathtt{j}^2 * \mathtt{x}^2))$
On tape le fichier suivant :

```
xyztrange(0,8,-7,1,-10,10,-1,6,0.,8,-7,1,1);
L:=plotfunc(equi(3,8,1,x),x,0,8/3);
for (p:=2;p<10;p:=p+1) {
m:=min(8,8*p/3);
L:=L,plotfunc(equi(3,8,p,x),x,0,m);
};
L;
```

On obtient les diffèrentes courbes d'équilibre lorsque le poids p varie (on peut vérifier que `limit(equi(3,8,3,x),x,8)=0`)

Chapitre 8

Un exemple de géométrie dans l'espace

8.1 L'énoncé

Soient $OABC$ un tétraèdre tel que :
— OAB, OAC, OBC sont des triangles rectangles
— $OA = OB = OC = 1$
Soient I le pied de la hauteur issue de C du triangle ABC
H le pied de la hauteur issue de O du triangle OIC et
D le point défini par $\overrightarrow{OD} = \overrightarrow{HO}$ Montrer :

1. Les droites OH et AB sont orthogonales
2. H est l'orthocentre du triangle ABC
3. Calculer OH
4. Le tétraèdre $ABCD$ est régulier
5. Calculer les coordonnées du centre s de la sphère circonscrite à $ABCD$.

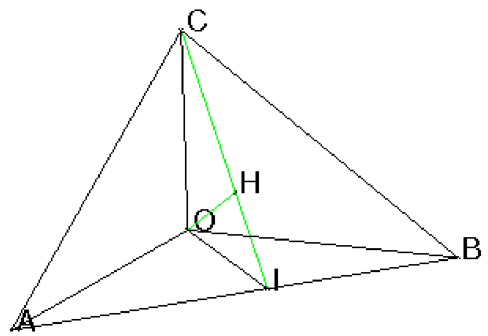
8.2 La solution avec `Xcas`

8.2.1 La figure

On ouvre un niveau de géométrie 3-d (`Alt+h`), on choisit comme repère $(O; \overrightarrow{OA}, \overrightarrow{OB}, \overrightarrow{OC})$ et on tape :

```
O:=point([0,0,0]);
A:=point([1,0,0]);
B:=point([0,1,0]);
C:=point([0,0,1]);
pyramide(O,A,B,C);
I:=projection(droite(A,B),C);
segment(C,I,affichage=2+ligne_tiret);
H:=projection(droite(I,C),O);
segment(O,H,affichage=2+ligne_tiret);
segment(O,I);
D:=translation(O-H,O);
```

On obtient :

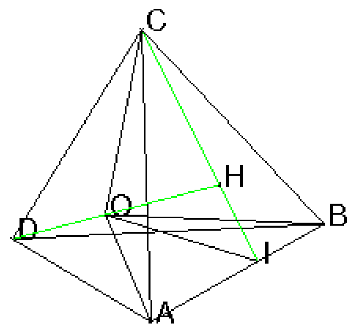


8.2.2 Les réponses aux questions

1. On tape :
   `est_orthogonal(droite(A,B),droite(O,H))`
   On obtient : `1`
   On tape :
   `longueur(I,H),longueur(C,H),longueur(C,I)`
   On obtient : `(sqrt(6))/6,(sqrt(6))/3,(sqrt(6))/2`
   On tape :
   `equation(droite(C,I))`
   On obtient : `1/2*x+(-1)/2*y=0,-1/2*x-1/2*y-1/2*z+1/2=0`
   On tape :
   `equation(plan(O,C,I))`
   On obtient : `-1/2*x-(-1)/2*y=0`
   On tape :
   `B-A,H-O,dot(B-A,H-O)`
   On obtient : `[-1,1,0],[1/3,1/3,1/3],0`

2. On tape :
   `est_equilateral(A,B,C)`
   On obtient : `1`
   On tape :
   `B-C,H-A,dot(B-C,H-A)`
   On obtient : `[0,1,-1],[(-2)/3,1/3,1/3],0`
   On tape :
   `est_orthogonal(droite(C,B),droite(A,H))`
   On obtient : `1`

3. On tape :
   `coordonnees(H)`
   On obtient :
   `[1/3,1/3,1/3]`
   On tape :
   `longueur(O,H)`
   On obtient : `(sqrt(3))/3`
4. On complète la figure en tapant :
   ```
   D:=translation(O-H,O);
   segment(D,H,affichage=2+ligne_tiret);
   pyramide(D,A,B,C,affichage=epaisseur_ligne_2)
   ```
   On obtient :

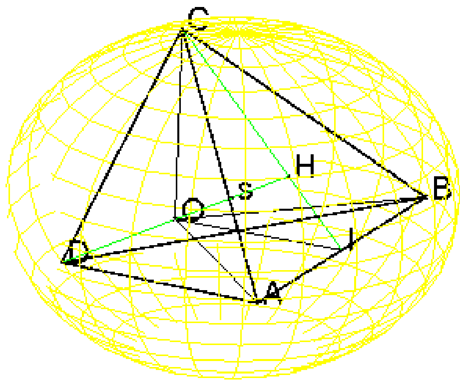


   On tape :
   `longueur(A,B),longueur(B,C),longueur(C,A)`
   On obtient : `sqrt(2),sqrt(2),sqrt(2)`
   On tape :
   `longueur(A,D),longueur(B,D),longueur(C,D)`
   On obtient : `sqrt(2),sqrt(2),sqrt(2)`
5. On tape, si on sait que le centre s de la sphère circonscrite au tétraèdre régulier $ABCD$ est l'isobarycentre des points A, B, C, D :
   `c:=coordonnees(isobarycentre(A,B,C,D))`
   On obtient : `[1/6,1/6,1/6],0]`
   Puis, on tape : `s:=point(c)`

   Si on ne sait pas que Le centre s de la sphère circonscrite à un tétraèdre régulier $ABCD$ est l'isobarycentre des points A, B, C, D, on tape :
   ```
   d:=inter(mediatrice(A,B),mediatrice(A,C));
   s:=inter_unique(mediatrice(A,D),d)
   ```

On obtient : `pnt(pnt[point[1/6,1/6,1/6],0])`

Puis, on tape : `r:=normal(longueur(A,s))`
On obtient : `sqrt(3)/2`
On tape :
`sphere(s,r)`
On obtient :



8.3 La solution en géométrie pure

1. Montrons que les droites OH et AB sont orthogonales.
   Le triangle ABC est équilatéral car les triangles OAB, OAC, OBC sont des triangles rectangles isocèles égaux.
   on a $AB = BC = CA = \sqrt{2}$ AB est perpendiculaire à CI car CI est une hauteur (et aussi médiane) du triangle équilatéral ABC. AB est perpendiculaire à OI car OI est une médiane et donc aussi une hauteur du triangle isocèle OAB de sommet O.
   Donc AB est perpendiculaire au plan COI donc AB est perpendiculaire à toutes les droites du plan COI et en particulier à OH.

2. Montrons que H est l'orthocentre du triangle ABC
   OH est perpendiculaire au plan ABC car orthogonal à AB et à CI qui sont 2 droites de ce plan.
   $OA = OB = OC = 1$ et H est la projection de O sur ABC donc $HA = HB = HC$.

H est donc le point de concours des médiatrices du triangle équilatéral ABC donc H est aussi l'orthocentre du triangle ABC.

3. Calculons OH
Le triangle COI est rectangle en O, $CO = 1$, $CI = \sqrt{2}\sqrt{3}/2 = \sqrt{6}/2$ et $OI = \sqrt{2}/2$ donc $OH = OI \times CO/CI = \sqrt{3}/3$

4. Montrons que le tétraèdre $ABCD$ est régulier.
$DH = 2 * OH = 2\sqrt{3}/3$
$CH = 2/3CI = \sqrt{2}\sqrt{3}/3$
Le triangle DCH étant rectangle en H on en déduit que :
$DC^2 = DH^2 + CH^2 = (12+6)/9 = 2$ donc $DC = \sqrt{2}$ On montre de même que $DA = \sqrt{2}$ et que $DB = \sqrt{2}$.
On a donc $DA = DB = DC = AB = BC = CA = \sqrt{2}$ donc
le tétraèdre $ABCD$ est régulier.
Ou bien on fait comme `Xcas`, on choisit comme repère $(O, \overrightarrow{OA}, \overrightarrow{OB}, \overrightarrow{OC})$ et on cherche les coordonnés de H dans ce repère.
H etant le point de concours des médiatrices du triangle équilatéral ABC de cotés $\sqrt{2}$, on a $HI = CI/3$ et donc $z_H = 1/3$
H est dans le plan ABC d'équation $x + y + z = 1$ et dans le plan COI d'équation $x = y$ donc $x_H + y_H = 2/3$ et $x_H = y_H$ donc H a pour coordonnées $[1/3, 1/3, 1/3]$.
On a $\overrightarrow{OD} = \overrightarrow{HO}$ donc D a pour coordonnées $[-1/3, -1/3, -1/3]$.
On calcule alors $DC^2 = 1/9 + 1/9 + (1 + 1/3)^2 = 2$

5. Calculons le rayon et les coordonnées du centre s de la sphère circonscrite à $ABCD$.

   On suppose que l'on sait que le centre s de la sphère circonscrite au tétraèdre régulier $ABCD$ est l'isobarycentre des points A, B, C, D.
$4\overrightarrow{Os} = \overrightarrow{OA} + \overrightarrow{OB} + \overrightarrow{OC} + \overrightarrow{OD}$ donc puisque $(1 - \frac{1}{3})/4 = 1/6$
$\overrightarrow{Os}$ a pour coordonnées : $[1/6, 1/6, 1/6]$.
Le rayon est donc $r = As = \sqrt{(25 + 1 + 1)/36} = \sqrt{3}/2$.

   Si on ne sait pas que Le centre s de la sphère circonscrite à un tétraèdre régulier $ABCD$ est l'isobarycentre des points A, B, C, D, on cherche un point s équidisdant de A, B, C, D. s est sur le segment DH car tous les points de ce segment sont équidisdant de A, B, C. On pose $Hs = l$ et on a :
$AH = \sqrt{2} * \sqrt{3}/3 * 2/3 = \sqrt{6}/3$ car H est le centre de gravité du triangle équlatéral ABC de coté $\sqrt{2}$.
$As^2 = AH^2 + l^2 = 2/3 + l^2$ car le triangle HAs est rectangle en H.
$DH^2 = AD^2 - AH^2 = 2 - 2/3 = 4/3$ car le triangle HAD est rectangle en H.
donc $Ds^2 = (DH - l)^2 = As^2 = 2/3 + l^2$
$2l * DH = 2/3$ et $DH = 2/\sqrt{3} = 2\sqrt{3}/3$
$Hs = l = 1/(3DH) = \sqrt{3}/6 = DH/4$
s est donc le milieu de OH et $r = Ds = 3DH/4 = \sqrt{3}/2$.

Chapitre 9

Un exemple traité avec un programme iteratif puis récursif

9.1 La suite des triangles semblables à ABC

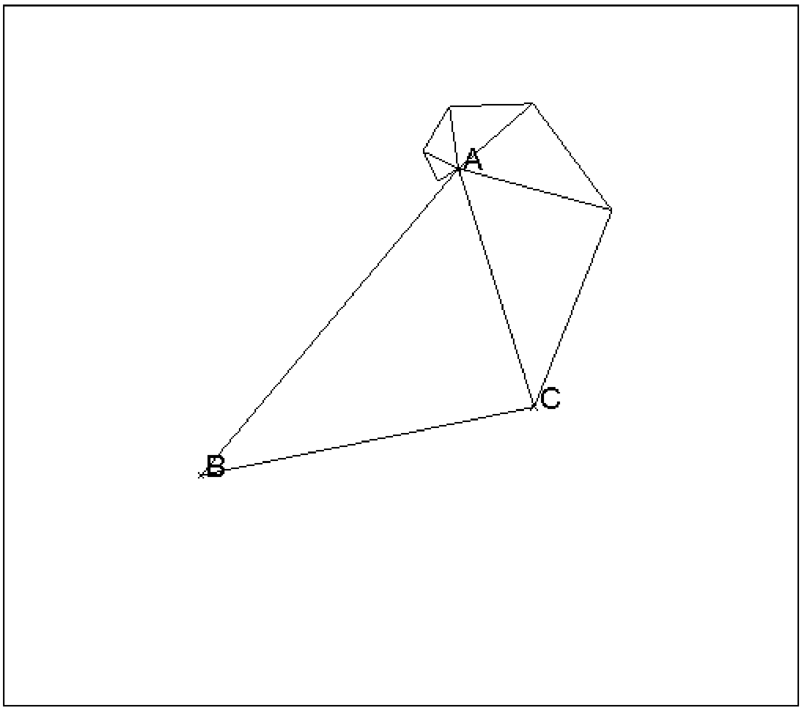
9.1.1 Avec un programme iteratif

À partir d'un triangle ABC, on trace le triangle ACC1 semblable au triangle ABC. Puis, on recommence le même processus avec le triangle ACC1 on obtient le triangle AC1C2 etc... Écrire un programme Xcas qui déssine la suite des n triangles :
ACC1, AC1C2,...AC(n-1)Cn.
On dessine tout d'abord le triangle ABC, puis on utilise une boucle qui calcule à chaque étape le nouveau point B (l'ancien point C) et le nouveau point C (celui obtenu dans la similitude de centre A de rapport k:=longueur(A,C)/longueur(A,B) et d'angle t:=angle(A,B,C)).
Attention
On ne suppose rien sur le triangle ABC car t:=angle(A,B,C) renvoie la mesure de l'angle oriené ($\rightarrow$ AB, $\rightarrow$ AC).
On tape :

```
//a partir d'un triangle ABC on trace son semblable
//sur AC etc...
//k=rapport et t=angle de la similitude
//n=nombre de triangles a construire
spirale(A,B,C,n):={
local k,t,j,L;
L:=triangle(A,B,C);
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
for (j:=1;j<=n;j++){
 B:=C;
 C:=similitude(A,k,t,B);
 L:=L,triangle(A,B,C);
 }
 retourne L;
```

```
}:;
```

On clique sur 3 points `ABC`, puis on tape :
`spirale(A,B,C,5)`
On obtient la suite de 6 triangles :



On peut écrire un programme légèrement diffèrent pour ne pas faire deux fois le même trait.
On trace le triangle `ABC` puis on ne trace dans la boucle que les 2 segments qui vont définir le nouveau triangle :

```
//meme dessin que spirale(A,B,C,n)
//mais sans repasser sur le meme trait
spirales(A,B,C,n):={
local k,t,L;
L:=triangle(A,B,C);
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
for (j:=1;j<=n;j++){
 B:=C;
 C:=similitude(A,k,t,B);
 L:=L,segment(B,C),segment(A,C);
}
retourne L;
}:;
```

9.1.2 Avec un programme récursif

On peut voir la récusivité de 2 façons :
1/ la procédure récursive `spiraler(A,B,C,n)`
`spiraler(A,B,C,0)` est le triangle(A,B,C) et
si n>0, `spiraler(A,B,C,n)` est formé du triangle(A,B,C) et de
`spiraler(A,C,C1,n-1)` avec `C1=similitude(A,k,t,C)`.
On écrit donc :

```
//Le dessin obtenu a partir d'un triangle ABC peut
//etre decrit de facon recursive si on a C=similitude(A,k,t,B),
//soit  C1=similitude(A,k,t,C) :
//spiraler(A,B,C,0)=triangle(A,B,C)
// si n>0, spiraler(A,B,C,n)=triangle(A,B,C) puis
//spiraler(A,C,C1,n-1)
//n=nombre de triangles a construire=nombre d'appels recursifs
spiraler(A,B,C,n):={
local k,t,L;
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
if (n>0) {
 L:=triangle(A,B,C);
 B:=C;
 C:=similitude(A,k,t,B);
 L:=L,spiraler(A,B,C,n-1);
   } else
 L:=triangle(A,B,C);
 retourne L;
}:;
```

On peut écrire un programme légèrement diffèrent pour ne pas faire deux fois le même trait.
`spiralers(A,B,C,0)` c'est le triangle(A,B,C) si n>0, `spiralers(A,B,C,n)` est formé du segment(A,B), du segment(B,C) et de `spiralers(A,C,C1,n-1)` si `C1=similitude(A,k,t,C)`.
On écrit donc :

```
//meme dessin que spiraler(A,B,C,n)
//mais sans repasser sur le meme trait
//si on a C=similitude(A,k,t,B), soit  C1=similitude(A,k,t,C)
//spiralers(A,B,C,0)=triangle(A,B,C) et si n>0,
//spiralers(A,B,C,n)=segment AB et BC puis spiralers(A,C,C1,n-1)
spiralers(A,B,C,n):={
local k,t,L;
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
if (n>0) {
 L:=segment(A,B),segment(B,C);
 B:=C;
 C:=similitude(A,k,t,B);
```

```
 L:=L,spiralers(A,B,C,n-1);
   } else
 L:=L,triangle(A,B,C);
 retourne L;
}:;
```

2/ la procédure récursive `spiraler1s(A,B,C,n)` commence à dessiner la spirale par la fin...
`spiraler1s(A,B,C,0)` c'est le triangle(A,B,C) et
si n>0, `spiraler1s(A,B,C,n)` est formé du triangle(A,C(n-1),Cn) (ou des segments C(n-1)Cn et ACn si on ne veut pas repasser sur le même trait) et
de `spiraler1s(A,B,C,n-1)` si `C(n-1)=similitude(A,k^n,n*t,B)`
et `Cn=similitude(A,k^n,n*t,C)=similitude(A,k,t,C(n-1))`.

```
//meme dessin que spiraler(A,B,C,n) mais autre facon de voir
//la recursivite spiraler1(A,B,C,0)=triangle(A,B,C) si n>0,
//spiraler1(A,B,C,n)=dernier triangle AMN et spiraler1(A,B,C,n-1)
//sans repasser sur le meme trait :
//spiraler1s(A,B,C,0)=triangle(A,B,C) et si n>0,
//spiraler1s(A,B,C,n)=segments AN et MN et spiraler1s(A,B,C,n-1)
spiraler1s(A,B,C,n):={
local k,t,L;
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
if (n>0) {
 M:=similitude(A,k^n,n*t,B);
 N:=similitude(A,k,t,M);
 L:=segment(M,N),segment(A,N);
 L:=L,spiraler1s(A,B,C,n-1);
   } else
 L:=triangle(A,B,C);
}:;
```

9.2 La double suite des triangles semblables à ABC

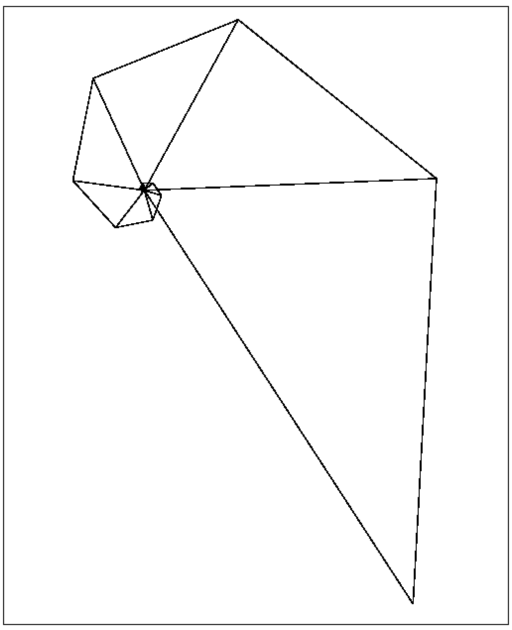
À partir d'un triangle `ABC`, on trace le triangle `ACC1` semblable au triangle `ABC`. Puis, on recommence le même processus avec le triangle `ACC1` on obtient le triangle `AC1C2` etc... On trace aussi le triangle `ABB1` semblable au triangle `ABC`. Puis, on recommence le même processus avec le triangle `ABB1` on obtient le triangle `AB1B2` etc..
Écrire un programme `Xcas` qui déssine la suite des `n` triangles :
`ACC1`, `AC1C2`,...`AC(n-1)Cn`,ABB1, `AB1B2`,...`AB(n-1)Bn`

9.2.1 Avec un programme iteratif

```
//a partir d'un triangle ABC on trace son semblable sur
//AC etc...n fois, on trace aussi le semblable du triangle ABC
// sur AB etc...(aussi n fois).
```

```
spirale2(A,B,C,n):={
local k,t,B0,C0,L;
L:=triangle(A,B,C);
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
B0:=B;
C0:=C;
for (j:=1;j<=n;j++){
 B:=C;
 C:=similitude(A,k,t,B);
 L:=L,triangle(A,B,C);
};
B:=B0;
C:=C0;
for (j:=1;j<=n;j++){
 C:=B;
 B:=similitude(A,1/k,-t,C);
 L:=L,triangle(A,B,C);
}
retourne L;
}:;
```

On clique sur 3 points `ABC` puis on tape :
`spirale2(A,B,C,5)`
On obtient la suite de 11 triangles :



pour faire le même dessin mais sans repasser sur le même trait, on tape :

```
//meme dessin que spirale2(A,B,C,n) mais sans repasser sur le meme tra
spirales2(A,B,C,n):={
local k,t,B0,C0,L,j;
L:=triangle(A,B,C);
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
B0:=B;
C0:=C;
for (j:=1;j<=n;j++){
 B:=C;
 C:=similitude(A,k,t,B);
 L:=L,segment(B,C),segment(A,C);
 };
B:=B0;
C:=C0;
for (j:=1;j<=n;j++){
 C:=B;
 B:=similitude(A,1/k,-t,C);
 L:=L,segment(B,C),segment(A,B);
 };
 retourne L;
}:;
```

9.2.2 Avec un programme récursif

Cette fois on a une seule façon de voir le dessin récursif : `spiralers2(A,B,C,0)` c'est le triangle(A,B,C) si n>0, `spiralers2(A,B,C,n)` est formé des segments ACn et C(n-1)Cn, de `spiraler1s(A,B,C,n-1)` puis dessegments ABn et B(n-1)Bn.
On tape :

```
//meme dessin que spirale2(A,B,C,n) mais en recursif
//cette fois on a une seule facon de voir le dessin
//(analogue a spiraler1s)
//et sans repasser sur le meme trait
//spiralers2(A,B,C,0)=triangle(A,B,C) si n>0,
//spiralers2(A,B,C,n)=segments AN et MN et spiraler1s(A,B,C,n-1)
//et segments AQ et PQ (M=C(n-1),N=Cn,P=B(n-1),Q=Bn)
spiralers2(A,B,C,n):={
local k,t,M,N,P,Q,L;
k:=longueur(A,C)/longueur(A,B);
t:=angle(A,B,C);
if (n>0) {
 M:=similitude(A,k^n,n*t,B);
 N:=similitude(A,k,t,M);
 L:=segment(M,N),segment(A,N);
 L:=L,spiralers2(A,B,C,n-1);
 P:=similitude(A,1/k^n,-n*t,C);
 Q:=similitude(A,1/k,-t,P);
```

```
 L:=L,segment(P,Q),segment(A,Q);
   } else
 L:=triangle(A,B,C);
 retourne L;}
:;
```

Remarques Ainsi on a :
`spiralers2(A,B,C,5)= spiralers(A,B,C,5);spiralers(A,C,B,5);`

On peut aussi rajouter un paramètre `s` supplémentaire qui donnera le sens de la spirale par rapport au signe de `t:=angle(A,B,C)` : `s=1` si la spirale tourne dans le même sens que les points `A,B,C` et `s=-1` sinon.
On tape :

```
//spiraler2s(A,B,C,1,3);spiraler2s(A,B,C,-1,3)
//equivalent a spiralers2(A,B,C,3)
//s= sens de la spirale par rapport au signe
// de l'angle oriente (AB,AC)
spiraler2s(A,B,C,s,n):={
 local k,t,L;
 if (s==1) {
  k:=longueur(A,C)/longueur(A,B);
 } else{
  k:=longueur(A,B)/longueur(A,C);
 }
 t:=angle(A,B,C);
 if (n>0) {
   L:=triangle(A,B,C);
   B:=similitude(A,k,s*t,B);
   C:=similitude(A,k,s*t,C);
   L:=L,spiraler2s(A,B,C,s,n-1);
 } else {
   L:=triangle(A,B,C);
 }
 retourne L;
}:;
```

On clique sur 3 points `ABC` (triangle `ABC` direct) puis on tape :
`spiraler2s(A,B,C,1,5);spiraler2s(A,B,C,-1,5)`
On obtient la suite de 11 triangles.

Chapitre 10

Quelques exemples de récursivité

10.1 Récursivité ayant un seul appel récursif

On commence par des exemples simples.

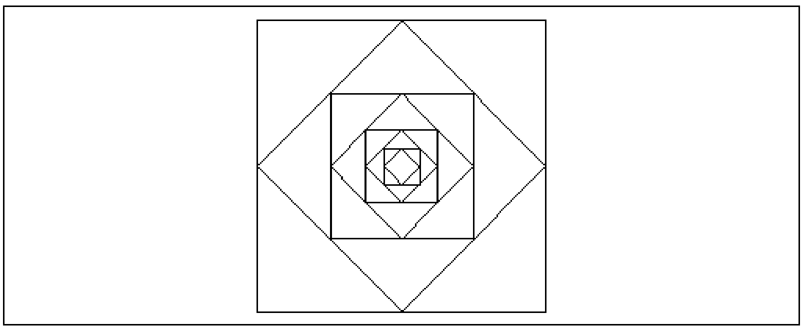
10.1.1 Les carrés

On trace un carré puis le carré qui joint les milieux des cotés etc... on s'arrête quand les segments à dessiner deviennent trop petits ou quand on a un dessin de profondeur n (n est le nombre d'étapes necessaires pour réaliser le dessin).

On tape dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` :

```
carres(A,B):={
local L;
L:=carre(A,B);
if (longueur2(A,B)>0.01) {
  L:=L,carres(A+(B-A)/2,B+(B-A)*i/2);
}
return L;
};
```

On tape :
`carres(point(-1),point(1))` :
On obtient :



On obtient le dessin des carrés avec le tracé qui est fait du plus grand au plus petit : le dessin du carré (-1,1,1+2*i,-1+2*i) puis du carré (0,1+i,i,-1+i)....

`bf Remarque` si on tape

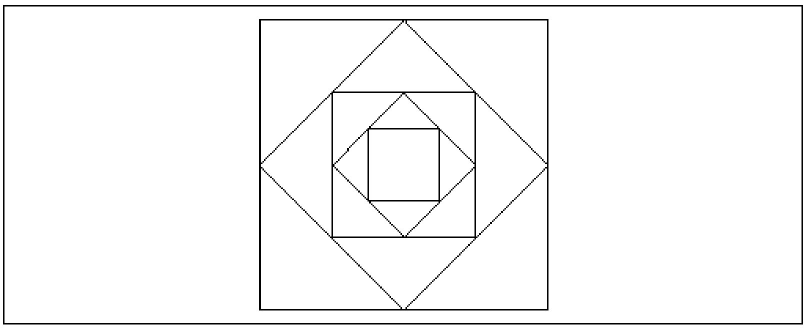
```
carres2(A,B):={
local L;
if (longueur2(A,B)>0.01) {
  L:=L,carres2(A+(B-A)/2,B+(B-A)*i/2);
}
L:=L,carre(A,B);
return L;
};
```

puis :
`carres2(-1.0,1.0)`
le dessin des carrés ne se fera pas dans le même ordre et se fera du plus petit au plus grand.

Autre test d'arrêt On peut avoir besion de connaitre le nombre de n de fois que l'on fait le ou les appels récursifs pour avoir un dessin de "profondeur" n. On rajoute pour cela un paramètre qui sera la profondeur.
Dans l'exemple ci-dessus, on tape dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` le programme :

```
carrep(A,B,n):={
local L;
L:=carre(A,B);
 if (n==0) return NULL;
 L:=L,carrep(A+(B-A)/2,B+(B-A)*i/2,n-1);
return L;
};
```

On tape :
`carrep(-1.0,1.0,5)`
On obtient le dessin des carrés du plus grand au plus petit et de profondeur 5

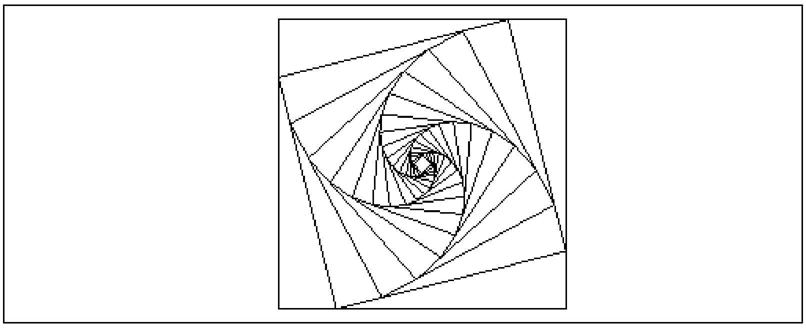


Généralisation
On trace un carré $ABCD$, puis le carré $MNPQ$ avec :

$\overrightarrow{AM} = a * \overrightarrow{AB}$,
$\overrightarrow{BN} = a * \overrightarrow{BC}$,
$\overrightarrow{CO} = a * \overrightarrow{CD}$,
$\overrightarrow{DP} = a * \overrightarrow{DA}$,
où a est un nombre réel entre 0 et 1.

```
carresp(A,B,a):={
local L;
L:=carre(A,B);
if (longueur2(A,B)>0.01) {
L:=L,carresp(A+(B-A)*a,B+(B-A)*i*a,a);
}
return L;
}:;
```

On tape par exemple :
`carresp(-1.0,1.0,0.2)` On obtient :

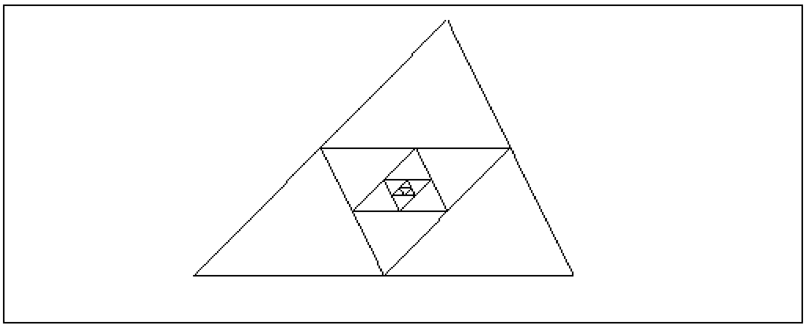


10.1.2 Les triangles

On trace un triangle puis le triangle qui joint les milieux des cotés etc...on s'arrête quand les segments à dessiner deviennent trop petits.
On tape dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` :

```
triangles(A,B,C):={
local L;
L:=triangle(A,B,C);
if (longueur2(A,B)>0.01) {
   L:=L,triangles(A+(B-A)/2,B+(C-B)/2,C+(A-C)/2);
}
return L};
```

On tape :
`triangles(-2.0,1,2*i)`
On obtient le dessin des triangles du plus grand au plus petit :le dessin du triangle (-2,1,2*i) puis du triangle (-0.5,0.5+i,-1+i,)....



Remarque si on tape

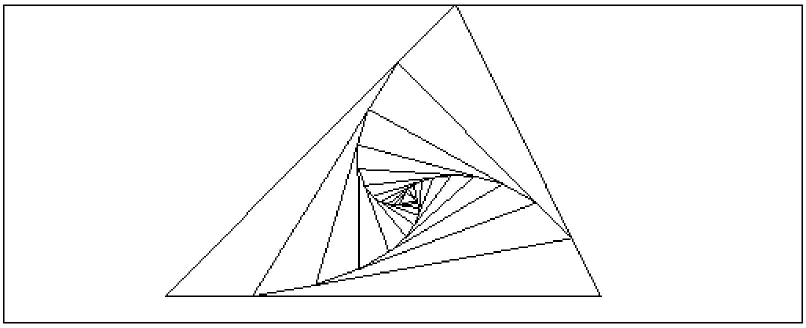
```
trianglesp(A,B,C):={
local L;
if (longueur2(A,B)>0.01) {
  L:=L,trianglesp(A+(B-A)/2,B+(C-B)/2,C+(A-C)/2);
}
L:=L,triangle(A,B,C);
return L};
```

On tape :
`trianglesp(-2.0,1,2*i)`
On obtient le même dessin, mais le tracé des triangles ne se fera pas dans le même ordre et se feradu plus petit au plus grand.

Généralisation
On trace un triangle ABC, puis le triangle MNP avec :
$\overrightarrow{AM} = a * \overrightarrow{AB}$,
$\overrightarrow{BN} = a * \overrightarrow{BC}$,
$\overrightarrow{CO} = a * \overrightarrow{CD}$,
où a est un nombre réel entre 0 et 1.

```
trianglea(A,B,C,a):={
local L;
L:=triangle(A,B,C);
if (longueur2(A,B)>0.01) {
L:=L,trianglep(A+(B-A)*a,B+(C-B)*a,C+(A-C)*a,a);
}
return L}:;
```

On tape par exemple :
`trianglea(-2.0,1.0,2*i,0.2)`
On obtient :



10.2 Récursivité ayant plusieurs appels récursifs

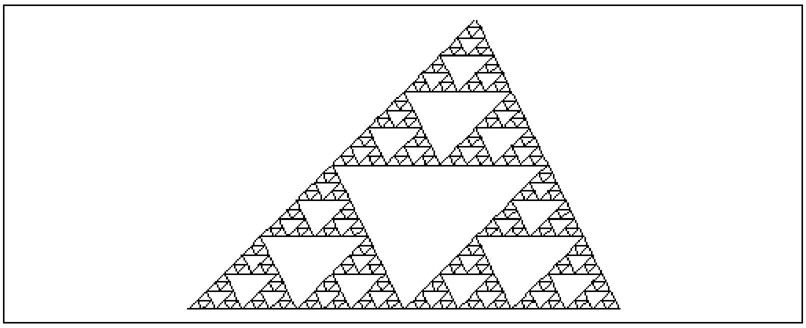
Voici des exemples encore assez simples.

10.2.1 Les triangles

On trace un triangle puis on joint les milieux des cotés.
On obtient ainsi 4 petits triangles semblables au précédent.
On recommence le même processus avec les trois triangles qui ont un angle commun avec le grand triangle et ainsi de suite.....on s'arrête quand les segments à dessiner deviennent trop petits.
On tape dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` :

```
triangle3(A,B,C):={
local L;
L:=triangle(A,B,C);
 if (longueur2(A,B)<0.005) return NULL;
 L:=L,triangle3(A,A+(B-A)/2,C+(A-C)/2);
 L:=L,triangle3(A+(B-A)/2,B,B+(C-B)/2);
 L:=L,triangle3(C+(A-C)/2,B+(C-B)/2,C);
return L;
};
```

On tape par exemple :
`triangle3(-2.0,1.0,2*i)`
On obtient :



Remarque
Le tracé du triangle ne peut se faire qu'a la fin car il suffit de tracer les derniers petits triangles on écrit donc dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` :

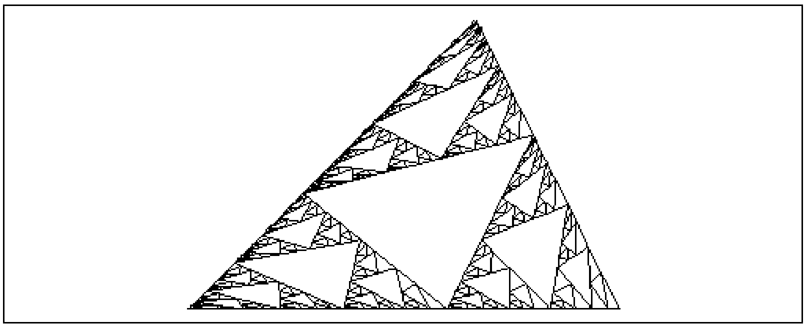
```
trianglep(A,B,C):={
local L:=NULL;
if (longueur2(A,B)<0.01) {return triangle(A,B,C);}
L:=L,trianglep(A,A+(B-A)/2,C+(A-C)/2);
L:=L,trianglep(A+(B-A)/2,B,B+(C-B)/2);
L:=L,trianglep(C+(A-C)/2,B+(C-B)/2,C);
return L;
};
```

On tape par exemple :
`trianglep(-2.0,1.0,2*i)`
On obtient le même dessin.

Généralisation
On trace un triangle ABC, puis le triangle MNP avec :
$\overrightarrow{AM} = a * \overrightarrow{AB}$,
$\overrightarrow{BN} = a * \overrightarrow{BC}$,
$\overrightarrow{CO} = a * \overrightarrow{CD}$,
où a est un nombre réel entre 0 et 1.

```
triangle3p(A,B,C,a):={
local L:=NULL;
if (longueur2(A,B)<0.02) {return triangle(A,B,C);}
L:=L,triangle3p(A,A+(B-A)*a,C+(A-C)*a,a);
L:=L,triangle3p(A+(B-A)*a,B,B+(C-B)*a,a);
L:=L,triangle3p(C+(A-C)*a,B+(C-B)*a,C,a);
return L;
}:;
```

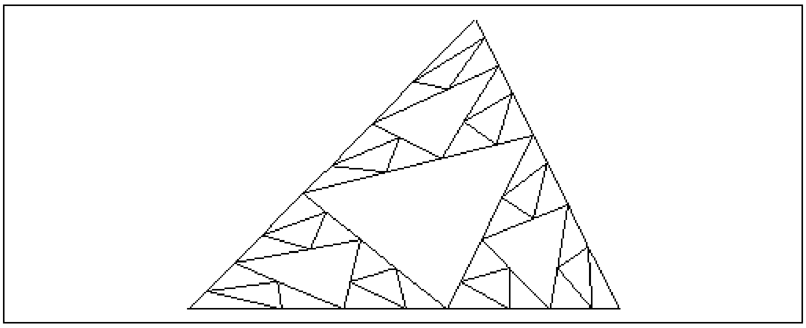
On tape par exemple :
`triangle3p(-2.0,1.0,2*i,0.6)`
On obtient :



Et avec un autre test d'arrêt en utilisant la profondeur n du dessin :

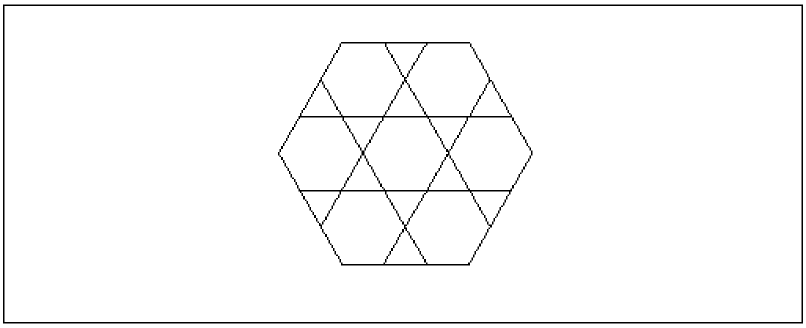
```
triangle3an(A,B,C,a,n):={
local L;
if (n==0) {return triangle(A,B,C);}
L:=L,triangle3an(A,A+(B-A)*a,C+(A-C)*a,a,n-1);
L:=L,triangle3an(A+(B-A)*a,B,B+(C-B)*a,a,n-1);
L:=L,triangle3an(C+(A-C)*a,B+(C-B)*a,C,a,n-1);
return L;
}:;
```

On tape par exemple :
`triangle3an(-2.0,1.0,2*i,0.6,3)`
0n obtient un dessin de profondeur 3 :



10.2.2 Les hexagones

On considère un hexagone de coés de longueur l, on remplace cet hexagone par 7 hexagones de cotés de longueur $l/3$ qui sont :les 6 hexagones ayant un angle commun avec l'hexagone de départ et un septième hexagone se trouvant au centre comme sur la figure :



On obtient ainsi 7 petits hexagones semblables au précédent et on recommence le même processus.

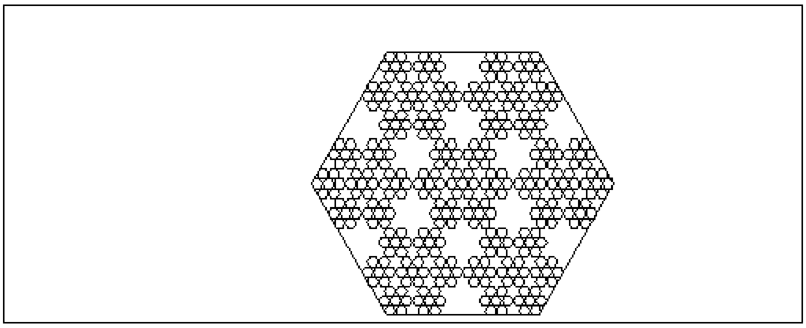
On tape la fonction `hexago` ou on utilise la commande `hexagone` de `Xcas` :

```
// dessin d'un hexagone
hexago(x,y):={
local a,b,c,L;
a:=x;
b:=y;
L:=NULL;
for (j:=1;j<=6;j++) {
c:=a+(b-a)*exp(evalf(i*pi*2/3));
L:=L,segment(a,c);
b:=a;
a:=c;
}
return L;
}:;
```

puis on fait un premier appel récursif correspondant à l'hexagone du centre et avec la même itération que dans la fonction `hexago`, on fait un appel récursif au lieu de tracer un segment pour les hexagones des angles, on tape :

```
hexagones(a,b,n):={
local j,c,L;
L:=NULL;
if (n==0) {return hexago(a,b);}
c:=a+(b-a)*2/3*exp(evalf(i*pi/3));
// dessin de l'hexagone central
L:=L,hexagones(c,c+(b-a)/3, n-1);
//dessin des 6 hexagones dans les angles
for (j:=1;j<=6;j++) {
c:=a+(b-a)*exp(evalf(i*pi*2/3));
L:=L,hexagones(c,c+(a-c)/3,n-1);
b:=a;
a:=c;
}
return L;};
```

On tape :
`hexagone(point(-1),point(1)),hexagones(-1,1,3)`
On obtient :



10.2.3 Les polygones réguliers

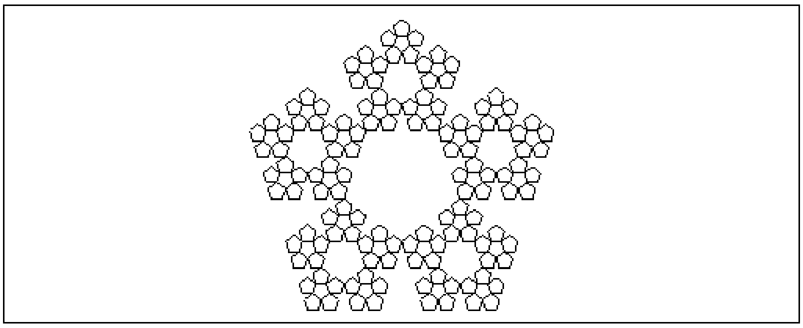
On part d'un polygone régulier P à k sommets et on le remplace par k polygones réguliers à k sommets de façon à ce que ces k polygones aient chacun un angle commun avec P et de façon à ce qu'ils ne se chevauchent pas. Puis on contine le processus et on ne dessine que les derniers petits polygones.
Par exemple un hexagone H de coté a est remplacé par 6 hexagones de côtés $a/3$ obtenus par homothétie de rapport 1/3 et de centre les sommets de l'hexagone H.
Pour écrire une procédure générale il faut faire un peu de trigonométrie.
Le calcul du coté h du petit polygone k-régulier doit vérifier :
- si k=3, 4 on a $h = (b - a)/2$
- si k=5, 6, 7, 8 on a $h + h\cos(2\pi/k) = (b - a)/2$
- si k=9, 10, 11, 12, 13, 14, 15, 16 on a :

$h + h\cos(2\pi/k) + h\cos(4\pi/k) = (b - a)/2$ - si p=`iquo`(k-1,4), on a :
$h\sum_{l=0}^{p} cos(2p\pi/k) = (b - a)/2$
On a donc :
$s = \sum_{l=0}^{p} cos(2p\pi/k) = (\sin((2p + 1)\pi/k) + \sin(\pi/k))/(2\sin(\pi/k))$
donc $h = (b - a)/2/s$
et on tape :

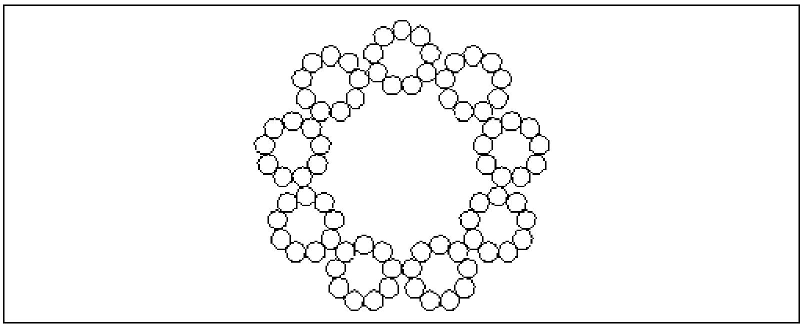
```
//napperon de Cantor ou de Sierpinski k=3,4...
//utilise isopolygone(a,b,k) k>0
//ex polyserp(-1-2*i,1-2*i,5,3); polyserp(-2*i,1-2*i,9,2)
polyserp(a,b,k,n):={
local c,h,j,q,p,s,L;
if (n==0) {return isopolygone(a,b,k);}
//pour k=3 ou 4  h:=(b-a)/3;
//pour k=5,6,7,8 h:=(b-a)/2/(cos(evalf(2*pi/k))+1);
//pour autre k il faut calculer s avec la trigo ou avec
//s:=1;for (l:=1;l<=iquo(k-1,4);l++){s:=s+cos(2*l*evalf(pi)/k);}
p:=iquo(k-1,4);
s:=(sin(evalf(pi)/k)+sin((2*p+1)*evalf(pi)/k))/
```

```
    2/sin(evalf(pi)/k);
for (j:=1;j<=k;j++) {
h:=(b-a)/2/s;
L:=L,polyserp(a,a+h,k,n-1);
c:=a+(b-a)*exp(evalf(i*pi*(k-2)/k));
b:=a;
a:=c;
}
retourne L;
};
```

On tape :
`polyserp(-1-2*i,1-2*i,5,3)`
On obtient :



On tape :
`polyserp(-2*i,1-2*i,9,2)`
On obtient :



Le programme du dessin d'un polygone régulier à k cotés peut vous aider á comprendre le programme précédent.

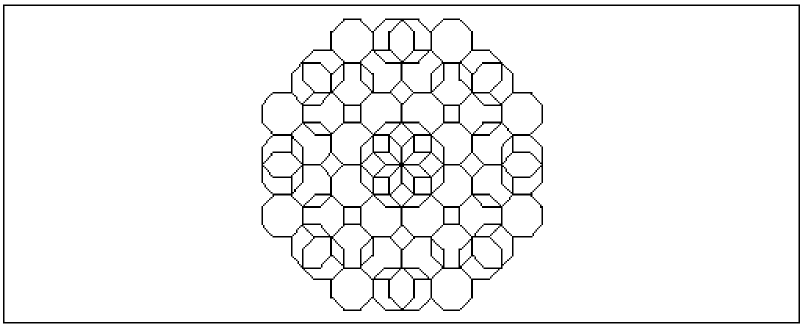
```
// dessin d'un polygone regulier de k cotes
polyreg(x,y,k):={
local a,b,c;
a:=x;
b:=y;
DispG();
for (j:=1;j<=k;j++) {
```

```
c:=a+(b-a)*exp(evalf(i*pi*(k-2)/k));
segment(a,c);
b:=a;
a:=c;
}
};
```

Dans Xcas, pour tracer un polygone régulier, on utilise la commande isopolygone.
Remarque
Une faute de signe peut vous faire voir de jolis dessins pour k=7,8...et n=2

```
//utilise polyreg(a,b,k) k>0 k=nb de cotes
//ex polyserr(-2*i,1-2*i,8,2); polyserr(-2*i,1-2*i,9,2)
polyserr(a,b,k,n):={
local c,h,j,q,p,s,L;
L:=NULL;
if (n==0) return isopolygone(a,b,k);
//if (n==0) {return polyreg(a,b,k);}
p:=iquo(k-1,4);
s:=(sin(evalf(pi)/k)-sin((2*p+1)*evalf(pi)/k))/
    2/sin(evalf(pi)/k);
for (j:=1;j<=k;j++) {
if ( s!=0) h:=(b-a)/2/s; else h:=(b-a)/3;
L:=L,polyserr(a,a+h,k,n-1);
c:=a+(b-a)*exp(evalf(i*pi*(k-2)/k));
b:=a;
a:=c;
}
return L;
};
```

On tape :
polyserr(-2*i,1-2*i,8,2)
On obtient :

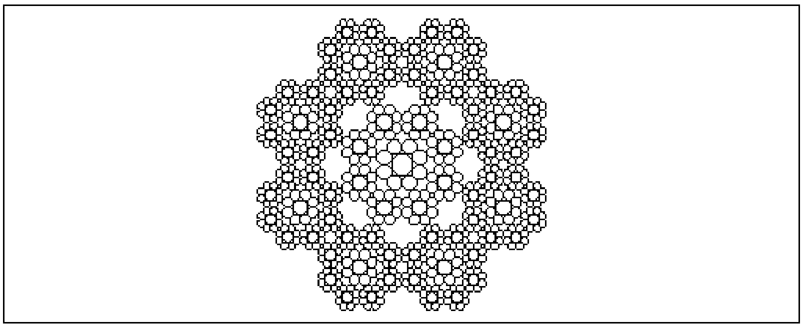


Un autre dessin avec des octogones en traçant un octogone au centre et 8 octogones dans les angles :

```
polyserp8(a,b,n):={
```

```
local c,h,j,q,p,s,k,L;
if (n==0) return isopolygone(a,b,8);
k:=8;
//if (n==0) {return polyreg(a,b,k);}
p:=iquo(k-1,4);
s:=(sin(evalf(pi)/k)+sin((2*p+1)*evalf(pi)/k))/
    2/sin(evalf(pi)/k);
h:=(b-a)/2/s;
L:=polyserp8(a+h+i*h*(sqrt(2)+1),a+h*(sqrt(2)+1)+i*h*(sqrt(2)+1),n-1);
for (j:=1;j<=k;j++) {
L:=L,polyserp8(a,a+h,n-1);
c:=a+(b-a)*exp(evalf(i*pi*(k-2)/k));
b:=a;
a:=c;
h:=(b-a)/2/s;
}
return L;
};
```

On tape :
`polyserp8(-2*i,1-2*i,3)`
On obtient :



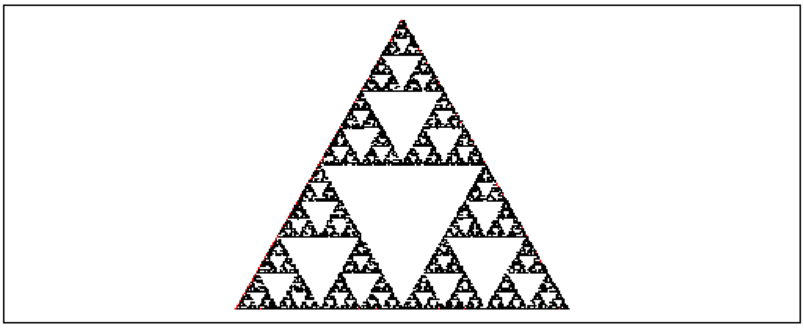
10.2.4 Le napperon de Cantor ou de Sierpinski avec des points aléatoires

On tape :

```
Sierpinski(n):={
local T,j,N,x,y,r;
T:=triangle_equilateral(-1/2-i*sqrt(3)/4,1/2-i*sqrt(3)/4,affichage=rou
z:=0;
N:=[0$n];
N[0]=<point(0,affichage=point_point)
pour j de 1 jusque n faire
r:=alea(3);
si r==0 alors
```

```
  x:=(-1/2+x)/2.;
  y:=(-sqrt(3)/4+y)/2.;
sinon
  si r==1 alors
    x:=(1/2+x)/2.;
    y:=(-sqrt(3)/4+y)/2.;
  sinon
    x:=x/2.;
    y:=(sqrt(3)/4+y)/2.;
  fsi;
fsi;
N[j]=<point(x,y,affichage=point_point)
fpour
retourne(T,N);
}
:;
```

On tape :
`Sierpinski(10000)`
On obtient :

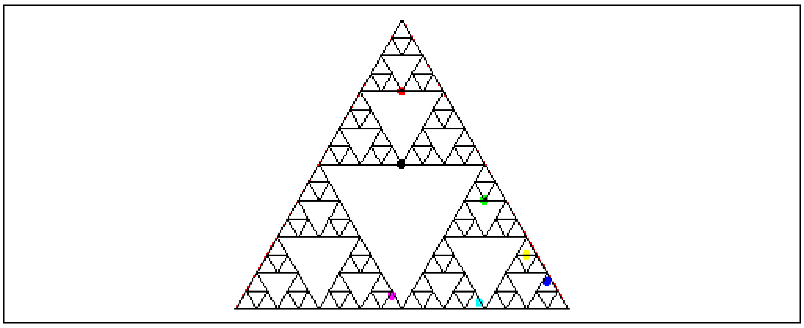


Pour comprendre on reprend le programme :

```
polyserp(a,b,k,n):={
local c,h,j,q,p,s,L;
if (n==0) {return isopolygone(a,b,k);}
//pour k=3 ou 4  h:=(b-a)/3;
//pour k=5,6,7,8 h:=(b-a)/2/(cos(evalf(2*pi/k))+1);
//pour autre k il faut calculer s avec la trigo ou avec
//s:=1;for (l:=1;l<=iquo(k-1,4);l++){s:=s+cos(2*l*evalf(pi)/k);}
p:=iquo(k-1,4);
s:=(sin(evalf(pi)/k)+sin((2*p+1)*evalf(pi)/k))/
    2/sin(evalf(pi)/k);
for (j:=1;j<=k;j++) {
h:=(b-a)/2/s;
L:=L,polyserp(a,a+h,k,n-1);
c:=a+(b-a)*exp(evalf(i*pi*(k-2)/k));
b:=a;
a:=c;
```

```
}
retourne L;
}:;
```

et on tape :
`Sierpinski(6),polyserp(-1/2-i*sqrt(3)/4,1/2-i*sqrt(3)/4,3,4)`
avec pour `Sierpinski(6)` 7 gros points aléatoires de couleurs 0,1...6. On a fait 6 tirages qui sont : 2, 1, 1, 1, 0 ,0.
On obtient :

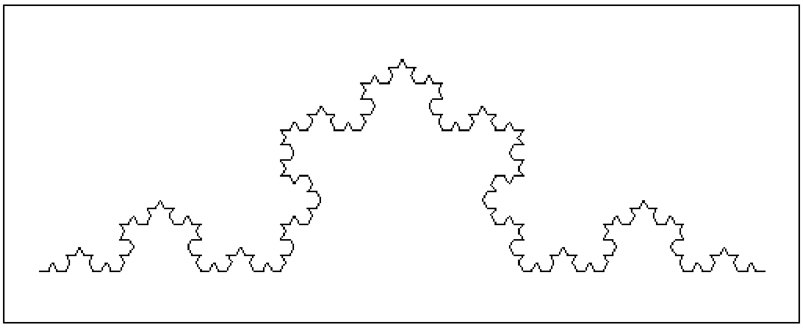


10.2.5 Le flocon de Koch

Tous les programmes qui sont dans cette section se trouve dans le fichier : `flocon.cas` On considère un segment AB et on place les points PQR tels que :
$3 * \overrightarrow{AP} = \overrightarrow{AB}$
$3 * \overrightarrow{BQ} = \overrightarrow{BA}$
et le triangle PQR est équilatèral direct.
On remplace alors le tracé du segment AB par le tracé $APRQB$.
On continue en faisant subir le même traitement aux 4 segments AP,PR,RQ,QB...on s'arrête quand la longueur des segments devient trop petite ou quand la profondeur est nulle.
Il y a donc 4 appels récursifs.

```
floconc(A,B):={
local L;
L:=NULL;
if (longueur2(A,B)<0.005) {return segment(A,B);}
L:=L,floconc(A,A+(B-A)/3);
L:=L,floconc(A+(B-A)/3,A+(B-A)/3*(1+exp(i*pi/3)));
L:=L,floconc(A+(B-A)/3*(1+exp(i*pi/3)),A+2*(B-A)/3);
L:=L,floconc(A+2*(B-A)/3,B);
return L;
}:;
```

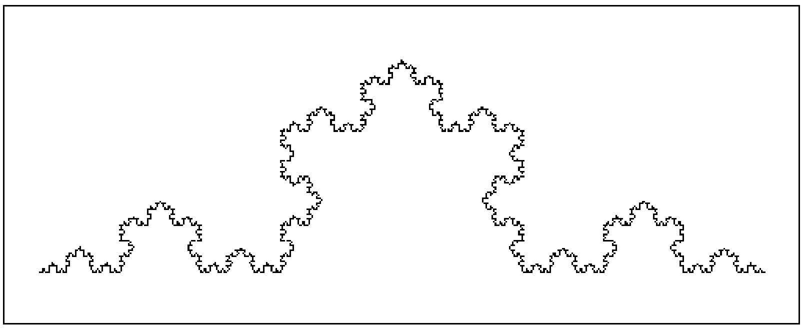
On tape par exemple :
`floconc(-1.0,1.0)`
On obtient une courbe de Koch :



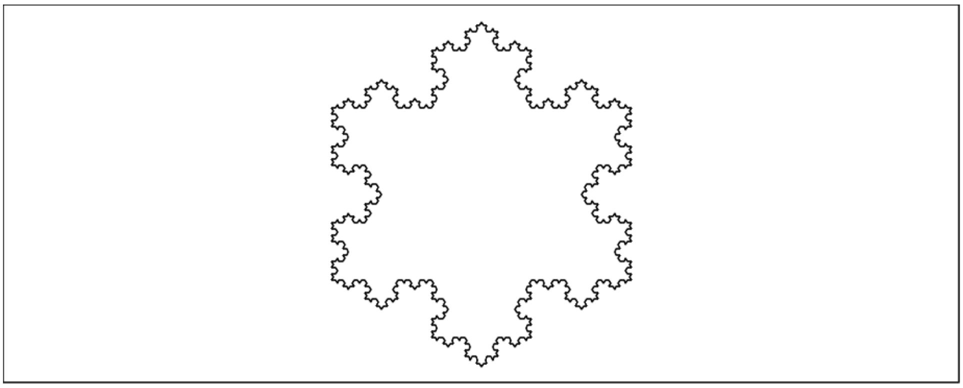
ou avec la profondeur :

```
floconp(A,B,n):={
local h,L;
L:=NULL;
if (n==0) {return segment(A,B);}
h:=(B-A)/3;
L:=L,floconp(A,A+h,n-1);
L:=L,floconp(A+h,A+h*(1+exp(i*pi/3)),n-1);
L:=L,floconp(A+h*(1+exp(i*pi/3)),A+2*h,n-1);
L:=L,floconp(A+2*h,B,n-1);
return L;
}:;
```

On tape par exemple :
`floconp(-2.0,2.0,5)`
On obtient une courbe de Koch :



Le flocon de Koch est formée par 3 courbes de Koch formant un triangle équilatéral.
On tape :
`floconp(1,0,4),floconp(0,1/2+i*sqrt(3)/2,4),`
`floconp(1/2+i*sqrt(3)/2,1,4)`
On obtient un flocon de Koch :

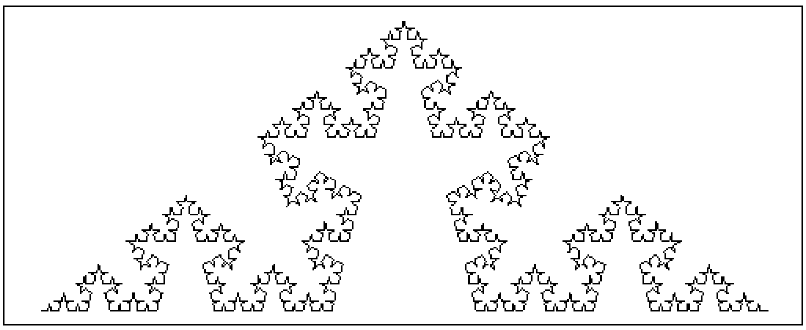


Généralisation

On coupe le segment initial AB en trois en placant P et Q de façon à avoir :
$\overrightarrow{AP} = a * \overrightarrow{AB}$
$\overrightarrow{BQ} = a * \overrightarrow{BA}$
avec $0.25 < a < 0.5$. puis on construit un triangle isocéle PQR tel que $PR = QR = AP$. On remplace le segment AB par les segments $APRQB$ et on recommence le processus. Il y a donc 4 appels récursifs.

```
flocong(A,B,a,n):={
local h,t,L;
L:=NULL;
if (n==0) {return segment(A,B);}
t:=acos((0.5-a)/a);
h:=(B-A)*a;
L:=L,flocong(A,A+h,a,n-1);
L:=L,flocong(A+h,A+h*(1+exp(i*t)),a,n-1);
L:=L,flocong(A+h*(1+exp(i*t)),B-h,a,n-1);
L:=L,flocong(B-h,B,a,n-1);
return L;
}:;
```

On tape par exemple :
`flocong(-2.0,2.0,0.4,4)`
On obtient :

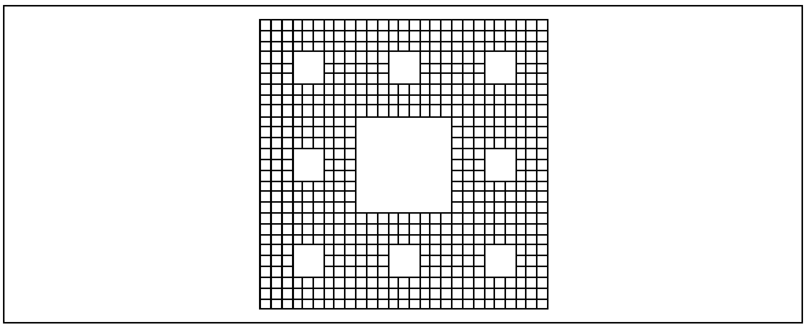


10.2.6 Le tapis carré

Ces programmes se trouvent dans `examples/recur/carre.cxx`.
On trace un carré puis puis on partage les cotés de ce carré en trois partie égales. On obtient ainsi 9 carrés.
On recommence le même processus avec les 8 carrés qui ont un coté commun avec le grand carré et ainsi de suite.....on s'arrête quand les segments à dessiner deviennent trop petits. Il y a donc 8 appels récursifs.
On tape dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` :

```
carre8(A,B):={
local h,L;
L:=carre(A,B);
if (longueur2(A,B)<0.005) return NULL;
h:=(B-A)/3;
L:=L,carre8(A,A+h);
L:=L,carre8(A+h,A+2*h);
L:=L,carre8(A+2*h,B);
L:=L,carre8(A+i*h,A+i*h+h);
L:=L,carre8(A+i*h+2*h,B+i*h);
L:=L,carre8(A+2*i*h,A+2*i*h+h);
L:=L,carre8(A+2*i*h+h,A+2*i*h+2*h);
L:=L,carre8(A+2*i*h+2*h,B+2*i*h);
return L;
}:;
```

On tape par exemple :
`carre8(-1.0,1.0)`
On obtient :



Autre test d'arrêt
On peut avoir besion de connaitre le nombre de n de fois que l'on fait le ou les appels récursifs pour avoir un dessin de "profondeur" n. On rajoute pour cela un paramètre qui sera la profondeur.
Dans l'exemple ci-dessus, on tape dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` le programme :

```
carre8p(A,B,n):={
local h,L;
h:=(B-A)/3;
L:=carre(A,B);
if (n==0)  return NULL;
h:=(B-A)/3;
L:=L,carre8p(A,A+h,n-1);
L:=L,carre8p(A+h,A+2*h,n-1);
L:=L,carre8p(A+2*h,B,n-1);
L:=L,carre8p(A+i*h,A+i*h+h,n-1);
L:=L,carre8p(A+i*h+2*h,B+i*h,n-1);
L:=L,carre8p(A+2*i*h,A+2*i*h+h,n-1);
L:=L,carre8p(A+2*i*h+h,A+2*i*h+2*h,n-1);
L:=L,carre8p(A+2*i*h+2*h,B+2*i*h,n-1);
return L;
}:;
```

On tape par exemple :
`carre8p(-1.0,1.0,4)` On obtient le même dessin

10.3 Quelques dessins doublement récursifs

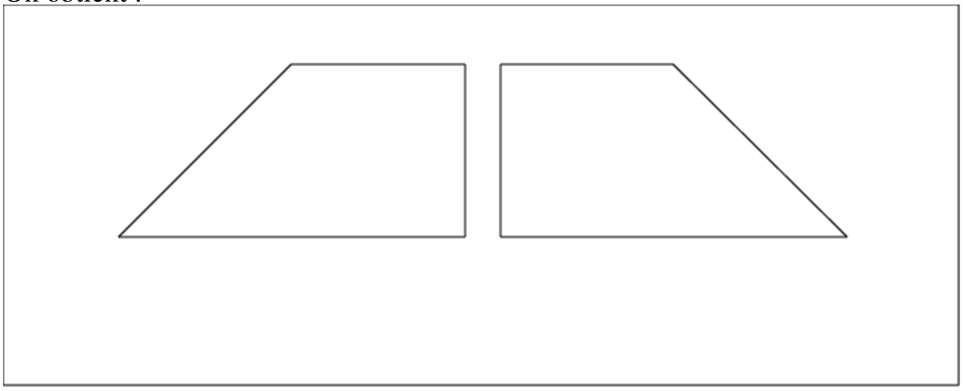
Dans ce chapitre on va faire des dessins qui obligent à écrire plusieurs procédures récursives qui s'appellent l'une l'autre.

10.3.1 Les trapèzes

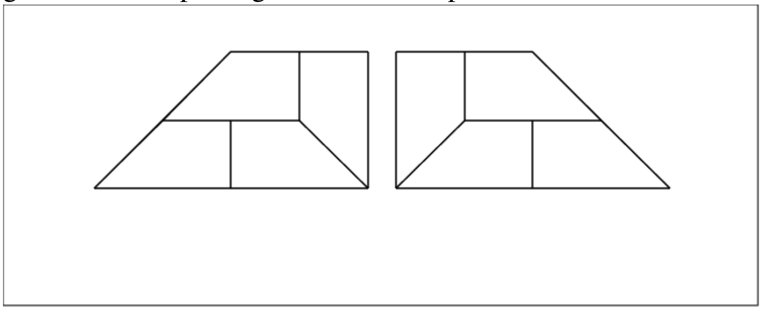
On considère les trapèzes $ABCD$ rectangle en A et D et tel que $AB = 2DC = 2AD$. On dira que l'on a un trapèze droit si l'angle $\overrightarrow{AB}, \overrightarrow{AD} = +\pi/2$ et sinon ce sera un trapèze gauche. Voici le programme du dessin du trapèze gauche et du trapèze droit : (le paramètre a est l'affixe de A et b est l'affixe de B)

```
trapd(a,b):={
local c,d,L;
L:=segment(a,b);
L:=L,segment(a,a+i*(b-a)/2);
L:=L,segment(a+i*(b-a)/2,(a+b)/2+i*(b-a)/2);
L:=L,segment(b,(a+b)/2+i*(b-a)/2);
retourne L;
}:;
trapg(a,b):={
local c,d,L;
L:=segment(a,b);
L:=L,segment(a,a-i*(b-a)/2);
L:=L,segment(a-i*(b-a)/2,(a+b)/2-i*(b-a)/2);
L:=L,segment(b,(a+b)/2-i*(b-a)/2);
retourne L;
}:;
```

On tape :
`trapg(0,10),trapd(10,20)`
On obtient :



On partage le trapèze droit en 3 trapèzes droits et un trapèze gauche et le trapèze gauche en 3 trapèzes gauches et un trapèze droit :

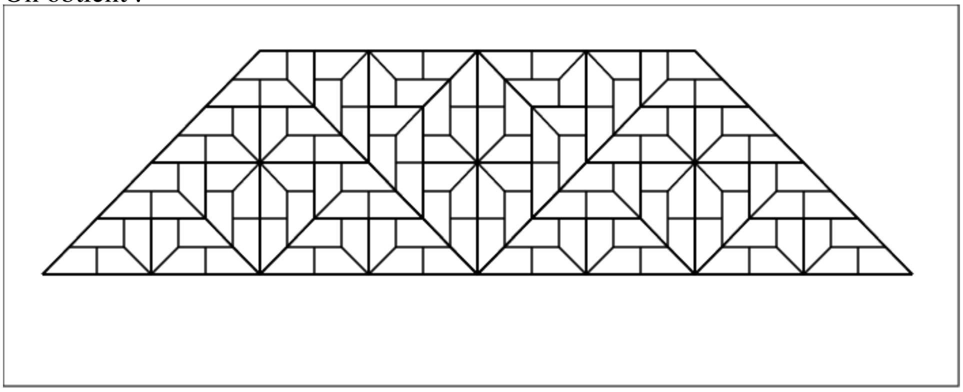


Puis on continue le même processus, on tape : (a et b sont les affixes de A et de B, $n = 1$ trace un trapèze, $n = 2$ dessine le partage (soit 4 trapèzes), $n = 3$ dessine $16 = 4^2$ trapèzes et $n = 4$ en dessine $64 = 4^3$ etc...)

```
trapdr(a,b,n):={
local L;
L:=NULL;
si n==0 alors retourne NULL; fsi;
L:=L,trapd(a,b);
L:=L,trapgr((a+b)/2,a,n-1);
L:=L,trapdr(a+i*(b-a)/2,a,n-1);
L:=L,trapdr(a+(b-a)/4+i*(b-a)/4,a+3*(b-a)/4+i*(b-a)/4,n-1);
L:=L,trapdr((a+b)/2,b,n-1);
return L;
}:;
trapgr(a,b,n):={
local L;
L:=NULL;
si n==0 alors return NULL; fsi;
L:=trapg(a,b);
L:=L,trapdr((a+b)/2,a,n-1);
L:=L,trapgr(a-i*(b-a)/2,a,n-1);
L:=L,trapgr(b+3*(a-b)/4+i*(a-b)/4,b+(a-b)/4+i*(a-b)/4,n-1);
L:=L,trapgr((a+b)/2,b,n-1);
```

```
return L;
}:;
```

On tape : `trapgr(10,0,4);trapdr(10,20,4);`
On obtient :

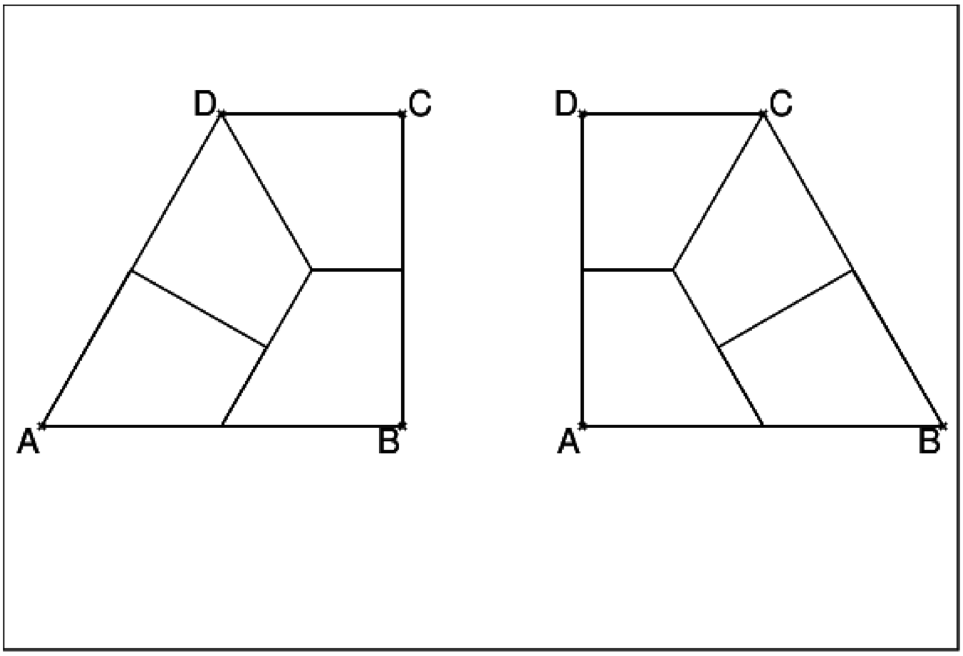


10.3.2 D'autres trapèzes

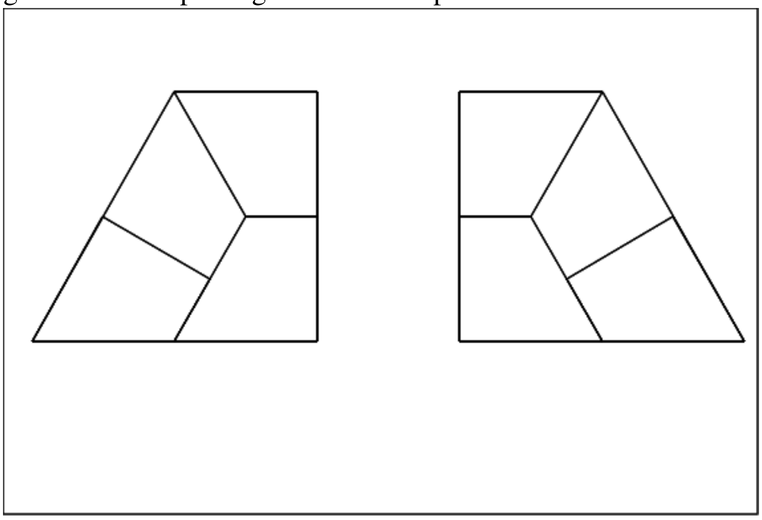
On considère le trapèze droit $ABCD$ rectangle en A et D et tel que $AD = AB\sqrt{3}/2 = BC\sqrt{3}/2 = DC\sqrt{3}$ et $\overrightarrow{AB}, \overrightarrow{AD} = +\pi/2$.
Le symétrique du traèze droit par rapport à une droite perpendiculaire à AB donne le trapèze gauche gauche. Voici le programme des dessins du trapèze gauche et du trapèze droit : (le paramètre a est l'affixe de A et b est l'affixe de B). On tape :

```
traped(a,b):={
local c,d,L;
L:=NULL;
L:=segment(a,b);
L:=L,segment(b,a+(b-a)/2+i*(b-a)*sqrt(3)/2);
L:=L,segment(a+(b-a)/2+i*(b-a)*sqrt(3)/2,a+i*(b-a)*sqrt(3)/2);
L:=L,segment(a+i*(b-a)*sqrt(3)/2,a);
retourne L;
}:;
trapeg(a,b):={
local c,d,L;
L:=NULL;
L:=segment(a,b);
L:=L,segment(b,a+(b-a)/2-i*(b-a)*sqrt(3)/2);
L:=L,segment(a+(b-a)/2-i*(b-a)*sqrt(3)/2,a-i*(b-a)*sqrt(3)/2);
L:=L,segment(a-i*(b-a)*sqrt(3)/2,a);
retourne L;
}:;
```

On tape :
`trapg(0,10),trapd(10,20)`
On obtient :



On partage le trapèze droit en 2 trapèzes droits et 2 trapèzes gauches et le trapèze gauche en 2 trapèzes gauches et 2 trapèzes droits :



Puis on continue le même processus. Pour faire les programmes récursifs de ces deux trapèzes, on peut comme précédemment prendre comme paramètres les affixes a et b de A et B, ou prendre comme paramètres : l'affixe a de A, la longueur l du segment AB et la valeur t de l'angle que fait AB avec l'axe Ox.
Attention le trapèze droit est rectangle en A et l'angle $\widehat{CBA} = \pi/3$ et le trapèze gauche est rectangle en B et l'angle $\widehat{DAB} = \pi/3$. On tape :

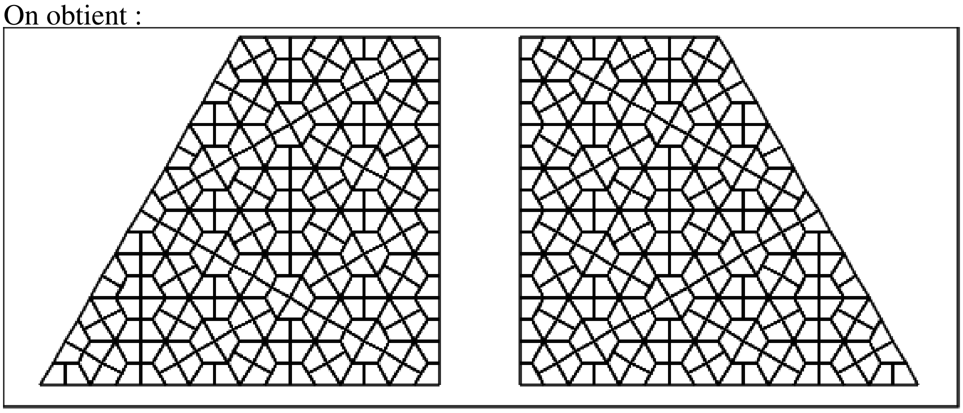
```
traperd(a,l,t):={
  local L;
  L:=NULL;
  si l<0.5 alors
    L:=L,polygone(a,a+l*exp(i*t),a+l*exp(i*t)+l*exp(i*(t+2*pi/3)),a+sqrt(3)/2
   sinon
   L:=L,traperd(a,l/2,t);
```

```
   L:=L,traperg(a+l*exp(i*t),l/2,t+2*pi/3);
   L:=L,traperd(a+l*exp(i*t)+l/2*exp(i*(t+2*pi/3)),l/2,t+2*pi/3);
   L:=L,traperg(a+l*exp(i*t)+l*exp(i*(t+2*pi/3)),l/2,t+pi);
 fsi;
   retourne L;
   }:;
 traperg(a,l,t):={
  local L;
  L:=NULL;
  si l<0.5 alors
    L:=L,polygone(a,a+l*exp(i*t),a+l*exp(i*t)+sqrt(3)/2*l*exp(i*(t+pi/
  sinon
  L:=L,traperd(a+l/2*exp(i*(t+pi/3)),l/2,t-2*pi/3);
  L:=L,traperg(a+l*exp(i*(t+pi/3)),l/2,t-2*pi/3);
  L:=L,traperg(a+l/2*exp(i*t),l/2,t);
  L:=L,traperd(a+l*exp(i*t)+l*sqrt(3)/2*exp(i*(t+pi/2)),l/2,t+pi);
  fsi;
   retourne L;
   }:;
```

On tape :

```
traperd(0,5,0)
```

On obtient :



Si on veut mettre de la couleur, on rajoute c comme paramètre :

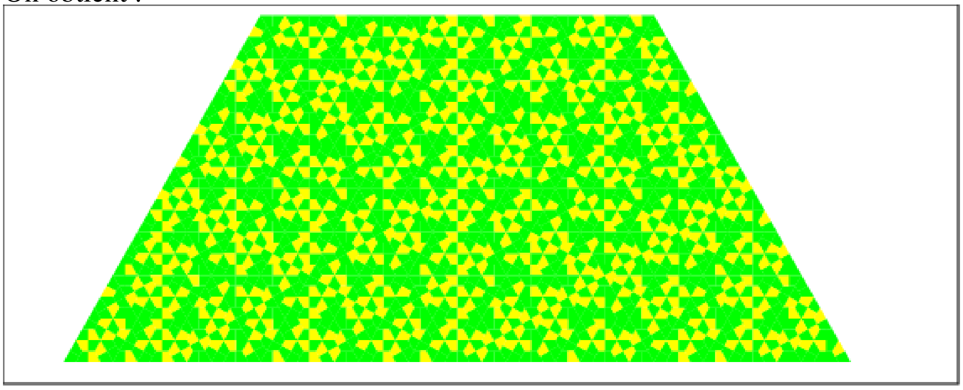
```
 traperdc(a,l,t,c):={
  local L;
  L:=NULL;
  si l<0.5 alors
    L:=L,affichage(polygone(a,a+l*exp(i*t),a+l*exp(i*t)+l*exp(i*(t+2*p
   sinon
   L:=L,traperdc(a,l/2,t,c);
   L:=L,trapergc(a+l*exp(i*t),l/2,t+2*pi/3,2);
   L:=L,traperdc(a+l*exp(i*t)+l/2*exp(i*(t+2*pi/3)),l/2,t+2*pi/3,3);
   L:=L,trapergc(a+l*exp(i*t)+l*exp(i*(t+2*pi/3)),l/2,t+pi,2);
 fsi;
   retourne L;
```

```
   }:;
 trapergc(a,l,t,c):={
  local L;
  L:=NULL;
  si l<0.5 alors
    L:=L,affichage(polygone(a,a+l*exp(i*t),a+l*exp(i*t)+sqrt(3)/2*l*exp(i*(t+
  sinon
  L:=L,traperdc(a+l/2*exp(i*(t+pi/3)),l/2,t-2*pi/3,c);
  L:=L,trapergc(a+l*exp(i*(t+pi/3)),l/2,t-2*pi/3,2);
  L:=L,trapergc(a+l/2*exp(i*t),l/2,t,2);
  L:=L,traperdc(a+l*exp(i*t)+l*sqrt(3)/2*exp(i*(t+pi/2)),l/2,t+pi,3);
  fsi;
   retourne L;
   }:;
```

On tape :

`trapergc(-10,10,0,3),traperdc(0,10,0,2)`

On obtient :



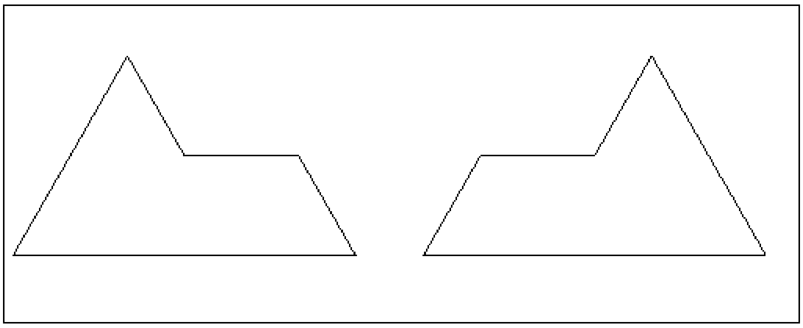
10.3.3 Les sphinx

Voici un sphinx droit et un sphinx gauche :

```
sphinxd(x,y):={
local z,u,t,L;
L:=NULL;
z:=x+2*(y-x)/3*exp(evalf(pi)*i/3);
t:= y+(x-y)/3*exp(-evalf(pi)*i/3);
u:=t+(x-y)/3;
L:=L,segment(x,z);
L:=L,segment(z,u);
L:=L,segment(u,t);
L:=L,segment(t,y);
L:=L,segment(y,x);
return L;
}:;
sphinxg(x,y):={
local z,u,t,L;
L:=NULL;
```

```
z:=y+2*(x-y)/3*exp(-evalf(pi)*i/3);
t:= x+(y-x)/3*exp(evalf(pi)*i/3);
u:=t+(y-x)/3;
L:=L,segment(y,z);
L:=L,segment(z,u);
L:=L,segment(u,t);
L:=L,segment(t,x);
L:=L,segment(x,y);
return L;
}:;
```

On tape :
`sphinxd(0,1),sphinxg(1.2,2.2)`
On obtient :

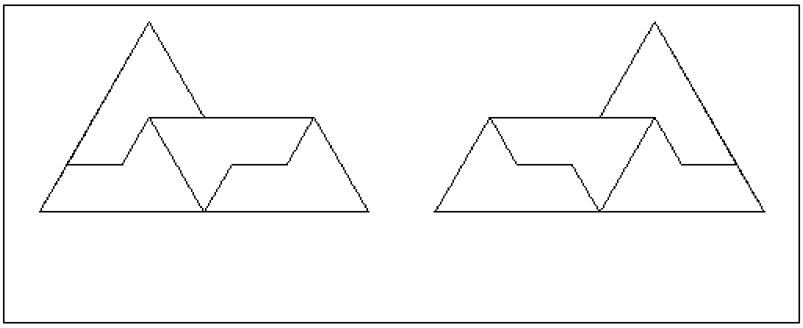


Voici un sphinx droit et ses 4 petits (composé de trois sphinx gauches et d'un sphinx droit) et un sphinx gauche et ses 4 petits (composé d'un sphinx gauches et de trois sphinx droit) :

```
sphinxd4(x,y):={
local z,u,t,L;
L:=NULL;
z:=x+2*(y-x)/3*exp(3.14*i/3);
t:= y+(x-y)/3*exp(-3.14*i/3);
u:=t+(x-y)/3;
L:=L,segment(x,z);
L:=L,segment(z,u);
L:=L,segment(u,t);
L:=L,segment(t,y);
L:=L,segment(y,x);
L:=L,sphinxg(x,(x+y)/2);
L:=L,sphinxg((x+y)/2,y);
L:=L,sphinxg(t,t+(x-y)/2);
L:=L,sphinxd(z,(3*x+z)/4);
return L;
}:;
sphinxg4(x,y):={
local z,u,t,L;
```

```
L:=NULL;
z:=y+2*(x-y)/3*exp(-evalf(pi)*i/3);
t:= x+(y-x)/3*exp(evalf(pi)*i/3);
u:=t+(y-x)/3;
L:=L,segment(y,z);
L:=L,segment(z,u);
L:=L,segment(u,t);
L:=L,segment(t,x);
L:=L,segment(x,y);
L:=L,sphinxd(x,(x+y)/2);
L:=L,sphinxd((x+y)/2,y);
L:=L,sphinxd(t+(y-x)/2,t);
L:=L,sphinxg((3*y+z)/4,z);
return L;
}:;
```

On tape :
`sphinxd4(0,1),sphinxg4(1.2,2.2)`
On obtient :



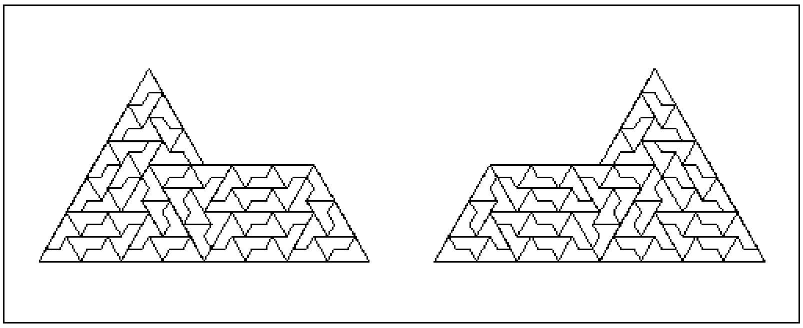
Et voici toute la famille des sphinx droits et toute la famille des sphinx gauches : (`sphinxds(x,y,n)` est une fonction récursive qui utilise la fonction récursive `sphinxgs(x,y,n)` et `sphinxgs(x,y,n)` est une fonction récursive qui utilise la fonction récursive `sphinxds(x,y,n)`).

```
sphinxds(x,y,n):={
local z,u,t,L;
if (n==0) return NULL;
z:=x+2*(y-x)/3*exp(evalf(pi)*i/3);
t:= y+(x-y)/3*exp(-evalf(pi)*i/3);
u:=t+(x-y)/3;
L:=NULL;
L:=L,segment(x,z);
L:=L,segment(z,u);
L:=L,segment(u,t);
L:=L,segment(t,y);
L:=L,segment(y,x);
L:=L,sphinxgs(x,(x+y)/2,n-1);
```

```
L:=L,sphinxgs((x+y)/2,y,n-1);
L:=L,sphinxgs(t,t+(x-y)/2,n-1);
L:=L,sphinxds(z,(3*x+z)/4,n-1);
return L;
}:;

sphinxgs(x,y,n):={
local z,u,t,p,L;
if (n==0) return NULL;
L:=NULL;
z:=y+2*(x-y)/3*exp(-evalf(pi)*i/3);
t:= x+(y-x)/3*exp(evalf(pi)*i/3);
u:=t+(y-x)/3;
L:=L,segment(y,z);
L:=L,segment(z,u);
L:=L,segment(u,t);
L:=L,segment(t,x);
L:=L,segment(x,y);
L:=L,sphinxds(x,(x+y)/2,n-1);
L:=L,sphinxds((x+y)/2,y,n-1);
L:=L,sphinxds(t+(y-x)/2,t,n-1);
L:=L,sphinxgs((3*y+z)/4,z,n-1);
return L;
}:;
```

On tape :
`sphinxds(0,1,4),sphinxgs(1.2,2.2,4)`
On obtient :



ou encore en ne dessinant que la dernière génération du sphinx droit ou la dernière génération du sphinx gauche :
sphindps(-2,2,4) met 0.52s alors que sphinds(-2,2,4) met 0.83s

```
sphinxdps(x,y,n):={
local z,u,t,L;
L:=NULL;
if (n==1) {return sphinxd(x,y);}
z:=x+2*(y-x)/3*exp(evalf(pi)*i/3);
t:= y+(x-y)/3*exp(-evalf(pi)*i/3);
```

```
u:=t+(x-y)/3;
L:=L,sphinxgps(x,(x+y)/2,n-1);
L:=L,sphinxgps((x+y)/2,y,n-1);
L:=L,sphinxgps(t,t+(x-y)/2,n-1);
L:=L,sphinxdps(z,(3*x+z)/4,n-1);
return L;
}:;
sphinxgps(x,y,n):={
local z,u,t,p,L;
L:=NULL;
if (n==1) {return sphinxg(x,y);}
z:=y+2*(x-y)/3*exp(-evalf(pi)*i/3);
t:= x+(y-x)/3*exp(evalf(pi)*i/3);
u:=t+(y-x)/3;
L:=L,sphinxdps(x,(x+y)/2,n-1);
L:=L,sphinxdps((x+y)/2,y,n-1);
L:=L,sphinxdps(t+(y-x)/2,t,n-1);
L:=L,sphinxgps((3*y+z)/4,z,n-1);
return L;
}:;
```

On tape :
`sphinxdps(0,1,4),sphinxgps(1.2,2.2,4)`
On obtient le même dessin que précédemment.

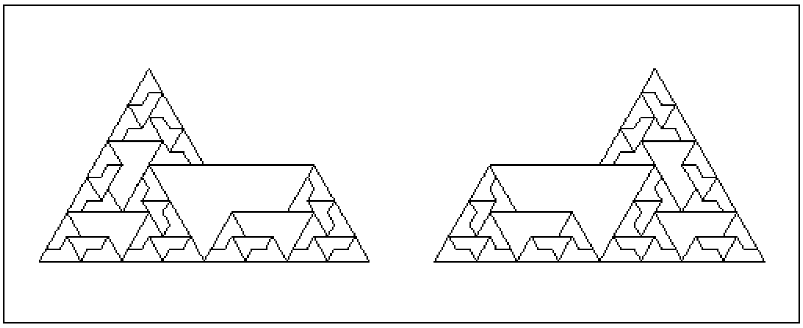
Mais si on remplace dans le sphinx droit, un sphinx gauche par un segment et dans le sphinx gauche, un sphinx droit par un segment on n'obtient pas la même chose ! ! !

```
sphinxdpst(x,y,n):={
local z,u,t,L;
L:=NULL;
if (n==1) return sphinxd(x,y);
z:=x+2*(y-x)/3*exp(evalf(pi)*i/3);
t:= y+(x-y)/3*exp(-evalf(pi)*i/3);
u:=t+(x-y)/3;
L:=L,sphinxgpst(x,(x+y)/2,n-1);
L:=L,sphinxgpst((x+y)/2,y,n-1);
L:=L,segment(t,t+(x-y)/3);
L:=L,sphinxdpst(z,(3*x+z)/4,n-1);
return L;
}:;

sphinxgpst(x,y,n):={
local z,u,t,p,L;
L:=NULL;
if (n==1) return sphinxg(x,y);
z:=y+2*(x-y)/3*exp(-evalf(pi)*i/3);
t:= x+(y-x)/3*exp(evalf(pi)*i/3);
```

```
u:=t+(y-x)/3;
L:=L,sphinxdpst(x,(x+y)/2,n-1);
L:=L,sphinxdpst((x+y)/2,y,n-1);
L:=L,segment(t+(y-x)/3,t);
L:=L,sphinxgpst((3*y+z)/4,z,n-1);
return L;
}:;
```

On tape :
`sphinxdpst(0,1,4),sphinxgpst(1.2,2.2,4)`
On obtient :



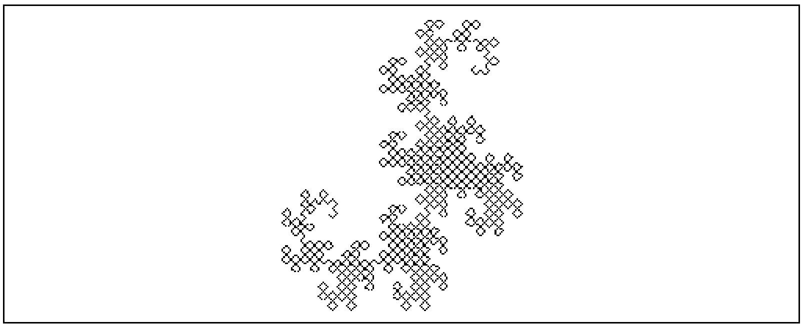
10.3.4 Le dragon

On se donne deux points A et B (ou deux nombres complexes a et b qui sont l'affixe de ces points) et on considère le carré $ACBD$ direct ayant pour diagonale AB.
Le segment AB peut donner naissance à un dragon gauche, pour cela, on remplace le segment AB par les deux côtés AD et DB du carré $ACBD$ situé à gauche du vecteur AB ou,
le segment AB peut donner naissance à un dragon droit, pour cela, on remplace le segment AB par les deux côtés AC et CB du carré $ACBD$ situé à droite du vecteur AB. Pour la fabrication du dragon gauche, ces deux segments sont considérés comme allant donner naissance à un dragon gauche (AD) et à un dragon droit (DB) et
pour la fabrication du dragon droit, ces deux segments sont considérés comme allant donner naissance à un dragon gauche (AC) et à un dragon droit (CB).
On a :
$b - c = i * (a - c)$ et $b - d = -i * (a - d)$ donc :
$c = (b - i * a) * (1 + i)/2$ et $d = (b + i * a) * (1 - i)/2$.
On écrit donc en prenant comme test d'arrêt la profondeur n c'est à dire le nombre de générations.

```
// dessine un dragon dragong(-i,2+i,10)
//x=a,y=b et d=u
 dragong(x,y,n):={
local u,L;
L:=NULL;
if (n==0){return segment(x,y);}
```

```
u:=(y+i*x)*(1-i)/2;
L:=L,dragong(x,u,n-1);
L:=L,dragond(u,y,n-1);
return L;
}:;
// dessine un dragon dragond(-i,2+i,10)
//x=a,y=b et c=u
dragond(x,y,n):={
local u,L;
L:=NULL;
if (n==0){return segment(x,y);}
u:=(y-i*x)*(1+i)/2;
L:=L,dragong(x,u,n-1);
L:=L,dragond(u,y,n-1);
return L;
}:;
```

On tape :
`dragong(-i,2+i,10)`
On obtient :



Remarque
Il est facile d'obtenir la courbe du dragon en prenant une longue bande de papier que l'on plie n fois sur elle même, toujours dans le même sens. Lorsqu'on a pris soin de bien marquer les plis, on obtient un dragon lorsqu'on déplie la bande en disposant les plis à angle droit.
Ce n'est pas tout à fait ce que l'on a programmer car dans le programme à chaque étape on multiplie la longueur du dragon par $\sqrt{2}$.
Sauriez vous programmer le dragon de la bande de papier ?

Voici la solution : on remarquera que le dragon droit est réalisé par la deuxième moitié de la bande de papier et donc la fonction `dragonpapierd` est la fonction `dragonpapierg` en changeant gauche en droite, et en commençant par la dernière instruction.

```
dragonpapierg(x,y,n):={
local u,v,a,b;
DispG();
if (n==0){segment(x,y);return y;}
```

```
u:=x+(y-x)/2;
a:=dragonpapierg(x,u,n-1);
v:=a+(y-x)*i/2;
b:=dragonpapierd(a,v,n-1);
return b
}:;

dragonpapierd(x,y,n):={
local u,v,a,b;
DispG();
if (n==0){segment(x,y); return y;}
v:=x+(y-x)*i/2;
b:=dragonpapierg(x,v,n-1);
u:=b+(y-x)/2;
a:=dragonpapierd(b,u,n-1);
return a;
}:;
```

On tape :
`dragonpapierg(-3.0,13,5)`
On obtient le dragon dans l'écran `DispG` :

Voici une autre solution où on repère l'arrivée et la direction du dernier trait. Dans ce cas on connait le départ et la direction de départ du dragon droit....mais c'est nettement plus compliqué.

```
dragonpaperg(x,y,n):={
local u,v,a,b;
DispG();
if (n==0){segment(x,y); return (x,y);}
u:=x+(y-x)/2;
a:=dragonpaperg(x,u,n-1);
v:=a[1]+abs((y-x)/(a[1]-a[0]))*(a[1]-a[0])*i/2;
b:=dragonpaperd(a[1],v,n-1);
return b;
}:;

dragonpaperd(x,y,n):={
local u,v,a,b;
DispG();
if (n==0){segment(x,y); return (x,y);}
u:=x+(y-x)/2;
a:=dragonpaperg(x,u,n-1);
v:=a[1]-abs((y-x)/(a[1]-a[0]))*(a[1]-a[0])*i/2;
b:=dragonpaperd(a[1],v,n-1);
return b;
}:;
```

On tape par exemple :
`dragonpaperg(-3,13,5)`
On obtient le dragon dans l'écran `DispG`.

10.4 Les courbes de Péano

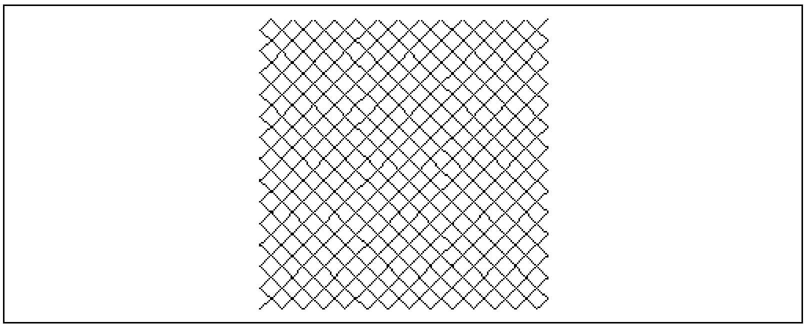
Parmi les nombreuses courbes inventées par Peano en voici quatre.

10.4.1 La première courbe de Péano

Une courbe de Péano passant par tous les points d'un carré.
On trace la diagonale AC d'un carré $ABCD$.
Puis puis on partage les cotés de ce carrés en trois partie égales. On obtient ainsi 9 carrés.
On remplace alors la diagonale du carré précédent par les diagonales des 9 carrés de façon à avoir une ligne continue allant de A à C. On recommence le même processus avec les 9 carrés de façon à avoir une ligne continue allant de A à C, et ainsi de suite..... On s'arrête quand les segments à dessiner deviennent trop petits.
On a choisit comme paramètre les affixes des points A et B et d'utiliser la profondeur comme test d'arrêt.
On tape dans un éditeur de programme (que l'on ouvre avec `Alt+p`), puis on valide avec `OK` :

```
//courbe qui remplit le carre de cote x,y
peano(x,y,n):={
local u,v,L;
u:=(y-x)/3;
v:=i*u;
L:=NULL;
if (n==0) {return segment(x,y+3*v);}
L:=L,peano(x,x+u,n-1);
L:=L,peano(x+u+v,x+u,n-1);
L:=L,peano(x+2*u,y,n-1);
L:=L,peano(y+v,y+2*v,n-1);
L:=L,peano(x+2*(u+v),x+u+2*v,n-1);
L:=L,peano(x+(u+v),x+u+2*v,n-1);
L:=L,peano(x+2*v,x+2*v+u,n-1);
L:=L,peano(x+3*v+u,x+u+2*v,n-1);
L:=L,peano(x+2*(u+v),y+2*v,n-1);
return L;
}:;
```

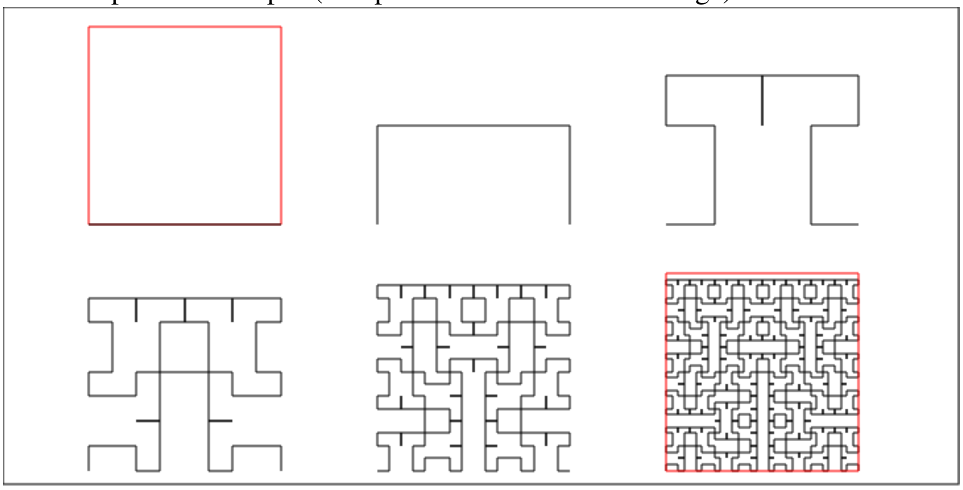
On tape par exemple :
`peano(-i,3-i,3)`
On obtient :



Vous pouvez voir les différentes étapes de la construction en faisant successivement $n = 1, 2, 3, 4$ en utilisant le bouton `STOP` si le tracé est trop long.

10.4.2 La courbe de Péano binaire

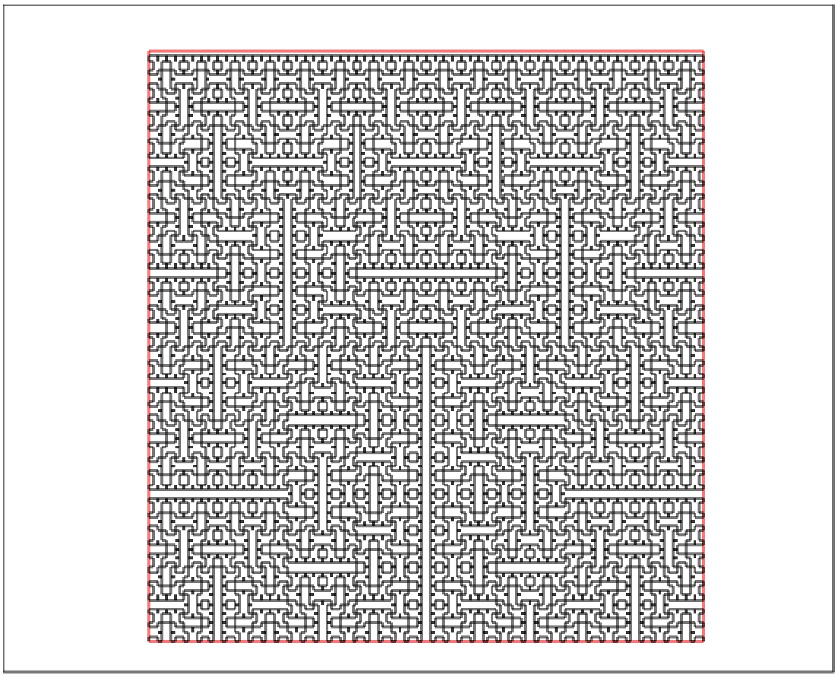
Une autre courbe de Péano passant par tous les points d'un carré que Péano a construit vers 1890.
Voici les premières étapes (l'étape 0 est le côté du carré rouge) :



Voici le programme récursif :

```
peano2(A,B,n):={
local C,D,M,L;
si n==0 alors retourne segment(A,B); fsi;
L:=NULL;rectangle(A,B,1/2,D,C);M:=milieu(D,C);
L:=L,peano2(B,C,n-1);
L:=L,peano2(D,A,n-1);
L:=L,peano2(D,M,n-1);
L:=L,peano2(M,C,n-1);
retourne L;
}:;
```

On tape par exemple :
`carre(0,10,affichage=1),peano2(0,10,6)`
On obtient :

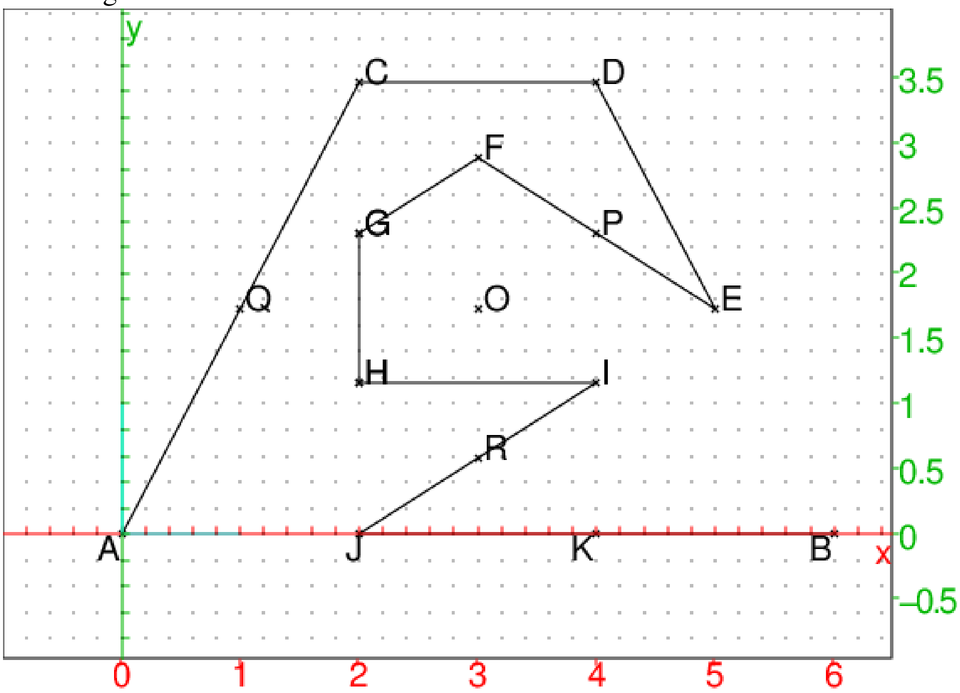


10.4.3 Deux courbes de Péano formée d'arcs de cercle

Parmi les nombreuses courbes inventées par Péano on va en décrire deux qui sont des courbes rècursives formées par des arcs de cercle.
Ces programmes se trouvent dans `examples/recur/peano.cxx`

Une courbe de Péano formée par 13 segments

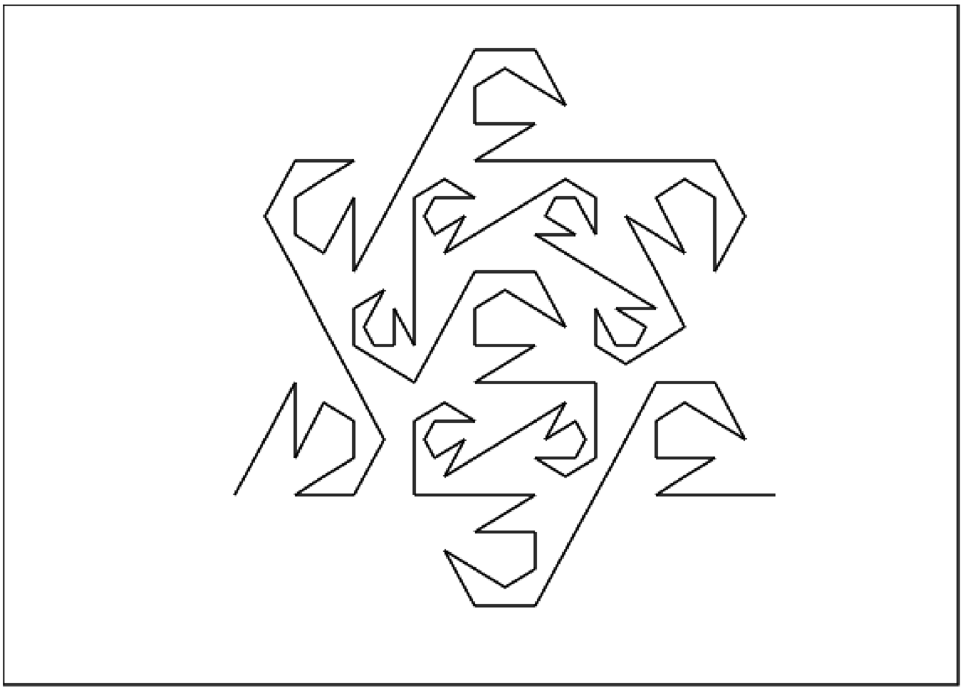
La figure de base :



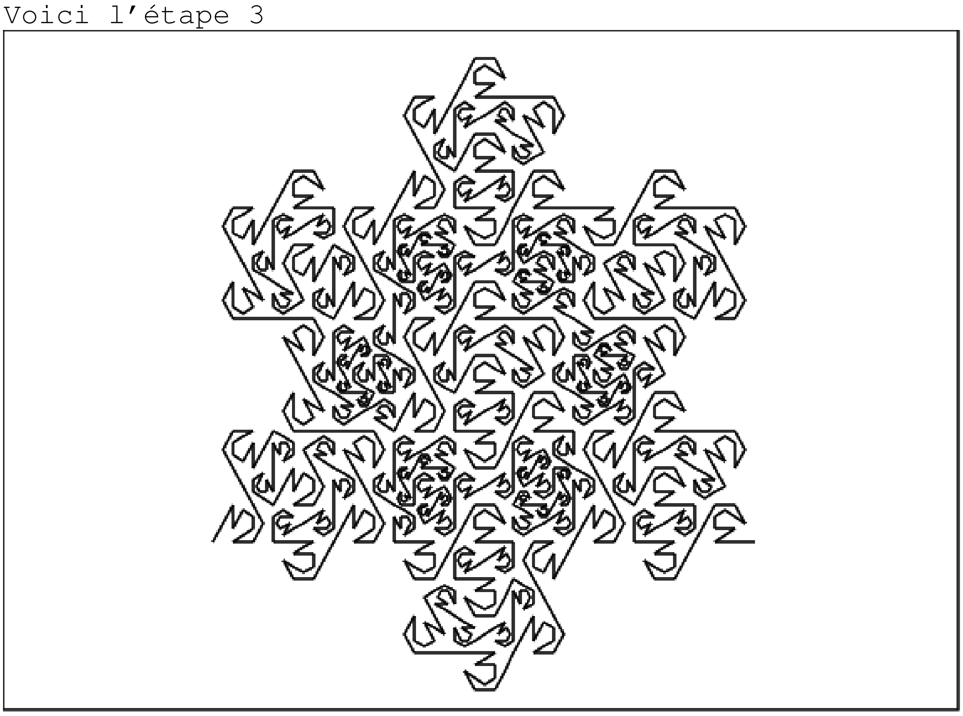
cette figure est obtenue en tapant :

```
supposons(a=[0,-5,5,0.1]);
supposons(b=[6,0,10,0.1]);
A:=point(a);B:=point(b) ;
C:=similitude(A,2/3,pi/3,B);
D:=translation((B-A)/3,C);
E:=rotation(D,2*pi/3,C);
M:=milieu(A,B):;;
N:=milieu(C,D):;;
O:=milieu(M,N);
F:=similitude(O,1/sqrt(3),pi/6,D);
G:=rotation(O,pi/3,F);
H:=rotation(O,2*pi/3,F);
I:=rotation(O,-2*pi/3,F);
J:=point((2A+B)/3);
K:=point((A+2B)/3);
polygone_ouvert(A,C,D,E,F,G,H,I,J,B);
P:=rotation(O,-pi/3,F); ;
G:=rotation(O,pi/3,F);
H:=rotation(O,2*pi/3,F); ;
R:=milieu(I,J);
Q:=milieu(A,C);
```

On a ainsi facilement les affixes des différents points en tapant, par exemple :
`simplify(affixe(C))`
On écrit la procédure récursive `peanoKs(A,B,n)`.
À l'étape $n=0$ on trace le segment AC,
À l'étape $n=1$ on remplace le segment AC par la figure
aux 13 segments, mais attention à l'orientation :
les figures créees par `peanoKs(A,B,1)` et par `peanoKs(B,A,1)`
sont symétriques par rapport au milieu de AB.
On choisit pour ces 13 segments des orientations différentes.
Voici l'étape 2



Voici l'étape 3



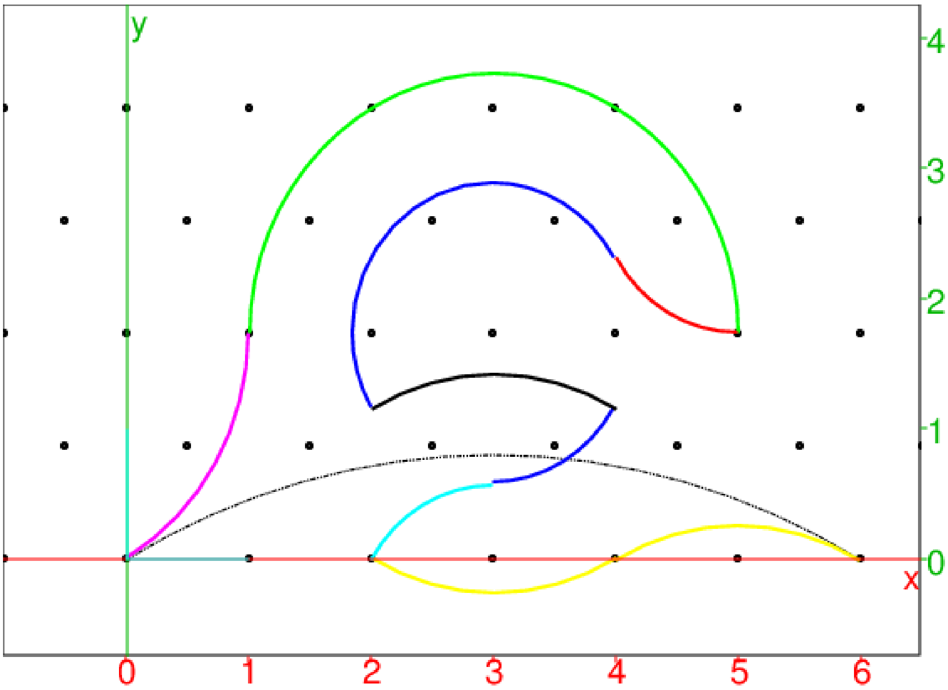
Voici le programme correspondant :

```
peanoKs(A,B,n):={
local a,b,L,C,D,E,F,G,H,I,J,K,O,M,N,P,Q,R;
si n==0 alors retourne segment(A,B); fsi;
a:=affixe(A);
b:=affixe(B);
L:=NULL;
C:=point((((-i)*sqrt(3)+2)*a+(i*sqrt(3)+1)*b)/3);
```

```
Q:=point((b+i*b*sqrt(3)+5*a+(-i)*a*sqrt(3))/6);
L:=L,peanoKs(Q,A,n-1);
L:=L,peanoKs(Q,C,n-1);
D:=point((2*b+i*b*sqrt(3)+a+(-i)*a*sqrt(3))/3);
L:=L,peanoKs(C,D,n-1);
E:=point((5*b+i*b*sqrt(3)+a+(-i)*a*sqrt(3))/6);
L:=L,peanoKs(D,E,n-1);
O:=point((3*b+i*b*sqrt(3)+3*a+(-i)*a*sqrt(3))/6);
F:=point((9*b+5*i*b*sqrt(3)+9*a+(-5*i)*a*sqrt(3))/18);
P:=point((6*b+2*i*b*sqrt(3)+3*a+(-2*i)*a*sqrt(3))/9);
L:=L,peanoKs(E,P,n-1);
L:=L,peanoKs(F,P,n-1);
G:=point((3*b+2*i*b*sqrt(3)+6*a+(-2*i)*a*sqrt(3))/9);
L:=L,peanoKs(G,F,n-1);
H:=point((3*b+i*b*sqrt(3)+6*a+(-i)*a*sqrt(3))/9);
L:=L,peanoKs(H,G,n-1);
I:=point((6*b+i*b*sqrt(3)+3*a+(-i)*a*sqrt(3))/9);
L:=L,peanoKs(H,I,n-1);
J:=point((2A+B)/3);
R:=point((9*b+i*b*sqrt(3)+9*a+(-i)*a*sqrt(3))/18);
L:=L,peanoKs(I,R,n-1);
L:=L,peanoKs(J,R,n-1);
K:=point((A+2B)/3);
L:=L,peanoKs(K,J,n-1);
L:=L,peanoKs(K,B,n-1);
  return L;
}:;
```

Une courbe de Péano formée par 13 arcs

On remplca les segments par des arcs d'angle $\pi/3$ ou d'angle $-\pi/3$.
L'arc AB (en pointillé) d'un cercle de rayon R et d'angle $-\pi/3$ (sur le dessin le centre est d'affixe $3-i3\sqrt{3}$ et $R=6$) est remplacé par 13 arcs d'angle $\pi/3$ ou d'angle $-\pi/3$:

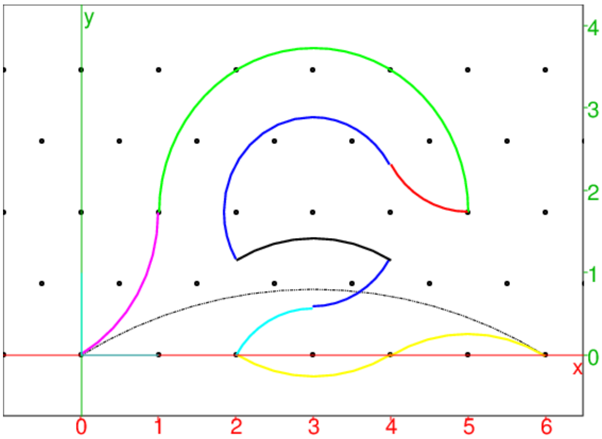


3 arcs d'angle $\pi/3$ se trouvent sur le cercle de centre $3+\sqrt{3}i$ et de rayon $2 = R/3$ (en vert),
1 arc d'angle $\pi/3$ (arc(0,3+i*sqrt(3)-2,pi/3)) se trouve sur le cercle de centre $i * sqrt(3) - 1$ et de rayon $2 = R/3$ (en magenta) 4 arcs , d'angle $\pi/3$ se trouvent sur le cercle de centre $3 + i\sqrt{3}$ et de rayon $2/\sqrt{3} = R\sqrt{3}/9$ (en bleu), 1 arc d'angle $\pi/3$ (arc(0,3+i*sqrt(3)-2,pi/3)) se trouve sur le cercle de centre $3-i*sqrt3/3$ et de rayon $2/\sqrt{3} = R\sqrt{3}/9$ (en cyan) 2 arcs symétriques par rapport au point d'affixe 4, d'angle $\pi/3$ (en jaune) (arc(2,4,pi/3) se trouve sur le cercle de centre $3+i\sqrt{3}$ et de rayon 2 et arc(6,4,pi/3) se trouve sur le cercle de centre $5 + (-i)sqrt3$ et de rayon 2. 1 arc d'angle $\pi/3$ se trouve sur le cercle de centre $3-i\sqrt{3}/3$ et de rayon 2 (en noir) 1 arc d'angle $\pi/3$ se trouve sur le cercle de centre $5+5i\sqrt{3}/3$ et de rayon $2\sqrt{3}/3$.(en rouge) On écrit la fonction arcg (resp arcd) qui dessine des arcs définit par le début de l'arc, la fin de l'arc, et de mesure $\pi/3$ (resp $-\pi/3$).
On tape :

```
papier_pointe(1,pi/3,sqrt(3)/2,x=-3..7,y=-4*sqrt(3)..4*sqrt(3));
affichage(arc(0,6,-pi/3),ligne_tiret_pointpoint);
affichage(arc(0,1+i*sqrt(3),pi/3),5+epaisseur_ligne_2));
affichage(arc(3+i*sqrt(3)-2, 3+i*sqrt(3)+2,-pi),2+epaisseur_ligne_2);
affichage(arc(4,6,-pi/3),3+epaisseur_ligne_2);
affichage(arc(4,2,-pi/3),3+epaisseur_ligne_2);
affichage(arc(3+i*sqrt(3)+2/sqrt(3)*exp(7*i*pi/6),
    3+i*sqrt(3)+2/sqrt(3)*exp(i*pi/6),-pi),4+epaisseur_ligne_2);
```

```
affichage(arc(3+i*sqrt(3)+2/sqrt(3)*exp(-i*pi/6),
    3+i*sqrt(3)-2*i/sqrt(3),-pi/3),4+epaisseur_ligne_2);
affichage(arc(3+i*sqrt(3)-2*i/sqrt(3), 2,pi/3),6+epaisseur_ligne_2);
affichage(arc(3+i*sqrt(3)+2/sqrt(3)*exp(i*pi/6),
    5+i*sqrt(3),pi/3),1+epaisseur_ligne_2);
affichage(arc(3+i*sqrt(3)+2/sqrt(3)*exp(-i*pi/6),
    3+i*sqrt(3)+2/sqrt(3)*exp(7*i*pi/6),pi/3), epaisseur_ligne_2) ;
```

On obtient le dessin :



Pour le faire le programme récursif, on tape :

```
//arc x y  de mesure +pi/3
arcg(x,y):={
return arc(x,y,pi/3);
}:;
//arc x y de de mesure -pi/3
arcd(x,y):={
return arc(x,y,-pi/3);
}:;
```

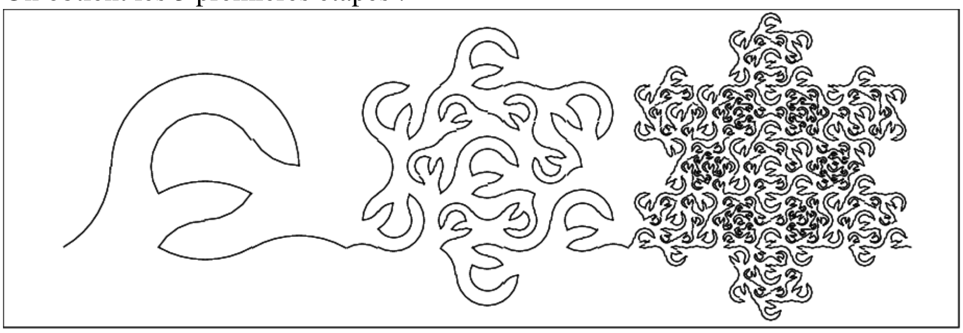
Puis on écrit la fonction peanog (resp peanod) :
Soient deux points A d'affixe a et B d'affixe b.
Pour la fonction peanog, on débute par l'arc AB de mesure $\pi/3$, situé sur le cercle de centre Cg d'affixe $cg = (b - a * exp(i * \pi/3)) * (1 - exp(-i * \pi/3))$ que l'on appellera arcg.
Pour la fonction peanod, on débute par l'arc AB de mesure $-\pi/3$, situé sur le cercle de centre Cd d'affixe $cd = (b - a * exp(-i * \pi/3)) * (1 - exp(i * \pi/3))$ que l'on appellera arcd.
On remplace ensuite arcg (resp arcd) par 13 arcs de mesure $\pi/3$ ou de mesure $-\pi/3$ selon le dessin que l'on obtient en tapant :
peanog(-2-2*i,2-2*i,1) (resp peanod(-2-2*i,2-2*i,1)).
Ces deux figures sont symétriques.
Et on continue en appliquant le même traitement à chacun de ses 13 arcs en remplacant les arcg (resp arcd) par 13 arcs.

```
// courbe de peano avec 13 arcs
//par ex peanod(-2-2*i,2-2*i,3)
peanod(x,y,n):={
local c1,b,c,d,e1,f,g,h,i1,j,k,l,m,L;
L:=NULL;
if (n==0) {return arc(x,y,-pi/3);}
c1:=x+(y-x)*exp(evalf(pi)*2*i/3)/3;
b:=x+(y-x)/3*exp(evalf(pi)*i/3);
c:=x+(y-x)/3*2*exp(evalf(pi)*i/3);
d:=c+(y-x)/3;
e1:=b+2*(y-x)/3;
f:=c1+(y-x)*(15+i*sqrt(3))/18;
g:=c1+(y-x)*(6+i*sqrt(3))/9;
h:=f-(y-x)/3;
i1:=h-i*(y-x)/9*sqrt(3);
j:=i1+(y-x)/3;
k:=g-i*2*(y-x)/9*sqrt(3);
l:=x+(y-x)/3;
m:=x+2*(y-x)/3;
L:=L,peanog(x,b,n-1);
L:=L,peanod(b,c,n-1);
L:=L,peanod(c,d,n-1);
L:=L,peanod(d,e1,n-1);
L:=L,peanod(e1,f,n-1);
L:=L,peanog(f,g,n-1);
L:=L,peanog(g,h,n-1);
L:=L,peanog(h,i1,n-1);
L:=L,peanod(i1,j,n-1);
L:=L,peanod(j,k,n-1);
L:=L,peanog(k,l,n-1);
L:=L,peanog(l,m,n-1);
L:=L,peanod(m,y,n-1);
return L;
}:;
peanog(x,y,n):={
local c1,b,c,d,e1,f,g,h,i1,j,k,l,m,L;
L:=NULL;
if (n==0) {return arc(x,y,pi/3);}
c1:=x+(y-x)*exp(evalf(-2*pi)*i/3)/3;
b:=x+(y-x)/3*exp(evalf(-pi)*i/3);
c:=x+(y-x)/3*2*exp(evalf(-pi)*i/3);
d:=c+(y-x)/3;
e1:=b+2*(y-x)/3;
f:=c1+(y-x)*(15-i*sqrt(3))/18;
g:=c1+(y-x)*(6-i*sqrt(3))/9;
h:=f-(y-x)/3;
i1:=h+i*(y-x)/9*sqrt(3);
j:=i1+(y-x)/3;
```

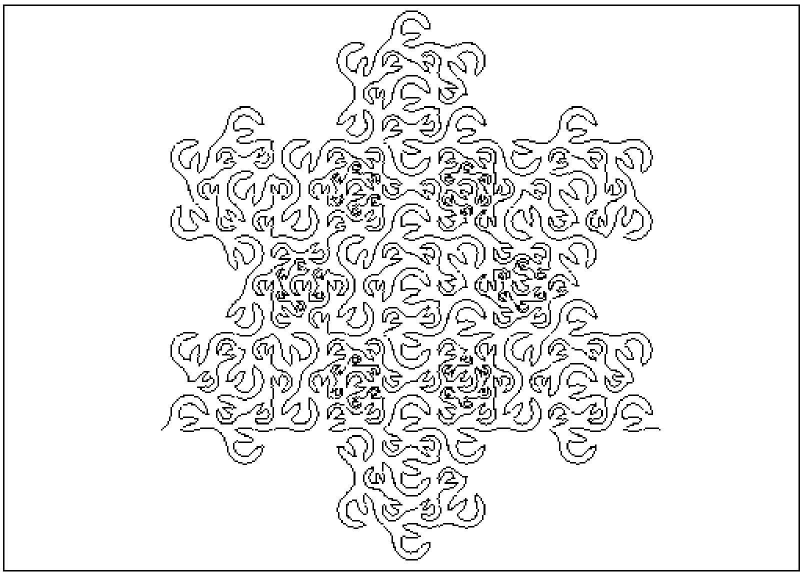
```
k:=g+i*2*(y-x)/9*sqrt(3);
l:=x+(y-x)/3;
m:=x+2*(y-x)/3;
L:=L,peanod(x,b,n-1);
L:=L,peanog(b,c,n-1);
L:=L,peanog(c,d,n-1);
L:=L,peanog(d,e1,n-1);
L:=L,peanog(e1,f,n-1);
L:=L,peanod(f,g,n-1);
L:=L,peanod(g,h,n-1);
L:=L,peanod(h,i1,n-1);
L:=L,peanog(i1,j,n-1);
L:=L,peanog(j,k,n-1);
L:=L,peanod(k,l,n-1);
L:=L,peanod(l,m,n-1);
L:=L,peanog(m,y,n-1);
return L;
}:;
```

```
On tape :
peanod(0,6,1),peanod(6,12,2),peanod(12,18,3
```

On obtient les 3 premières étapes :



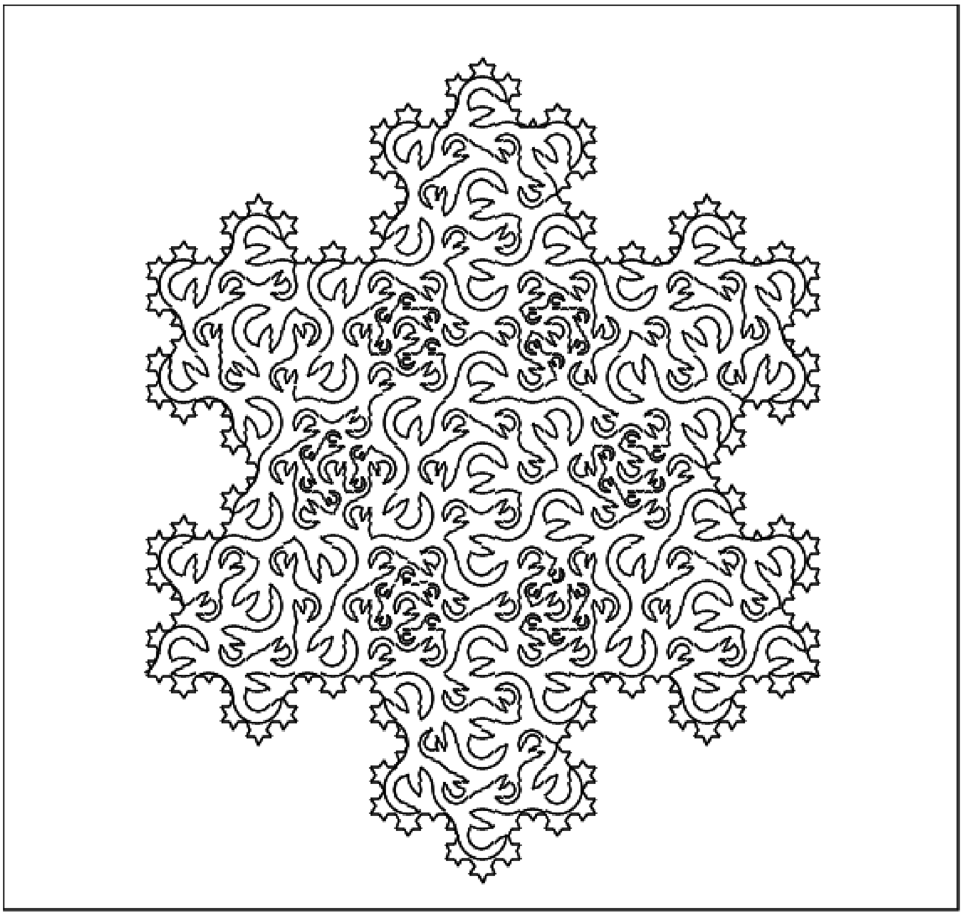
On tape :
`peanod(-2-2*i,2-2*i,3)`
On obtient :



On tape :

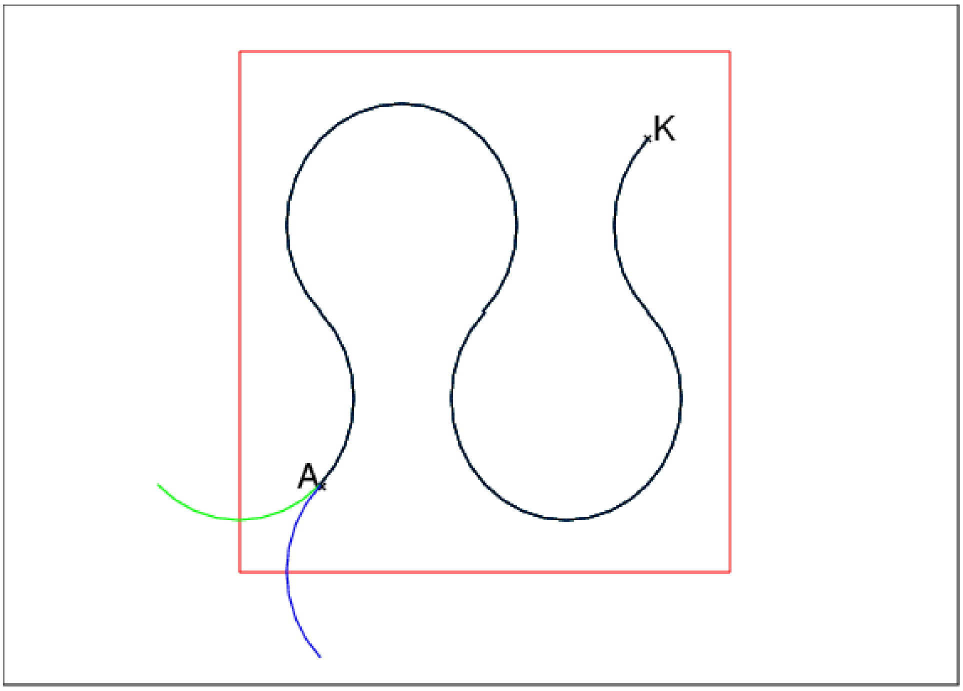
```
floconp(1,0,4),floconp(0,1/2+i*sqrt(3)/2,4),
floconp(1/2+i*sqrt(3)/2,1,4),peanod(0,1,3)
```

On obtient la courbe de péano qui rempli le flocon de Koch :

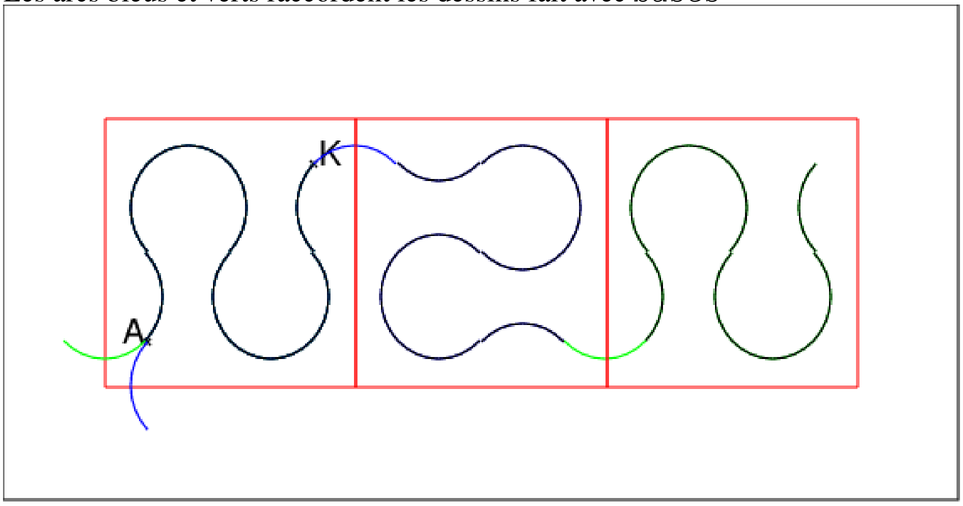


La courbe $\mathcal{C}^0$ de Péano ternaire

On va maintenant décrire la courbe de Péano ternaire qui remplit un carré direct de coté le segment XY, en étant $\mathcal{C}^0$.
Soient x est l'affixe du point X et y est l'affixe du point Y.
Le dessin de base est obtenu en tapant :
`bases(x,y)`
Il est composé de 8 arcs de cercle de mesure $\pi/2$ ou de mesure $-\pi/2$ et débute au point A d'affixe $a := x + (y - x)/6 * (1 + i)$ et se termine au point K d'affixe k, symétrique de A par rapport au centre du carré.
Á partir de ce dessin de base on fait la figure un :
`un(x,y)`
qui est composée d'un arc de mesure $\pi/2$ commençant au point d'affixe :
$a - (y - x)/3$ et se terminant au point A suivi du dessin de base.
Á partir de ce dessin de base on fait la figure deux :
`deux(x,y)`
qui est composée d'un arc de mesure $\pi/2$ commençant au point d'affixe :
$a - i * (y - x)/3$ et se terminant au point A suivi du dessin de base.



en rouge `carre(x,y)`, en noir `bases(x,y)`,
en vert + noir `un(x,y)`, en bleu + noir `deux(x,y)`
Les arcs bleus et verts raccordent les dessins fait avec `bases`



On partage le carré en 9 petits carrés et dans chacun des petits carrés on trace la figure de base ou la figure de baseun ou la figure de basedeux de façon à obtenir une ligne continue etc....
On écrit ensuite les procédures :
`peano0` qui débute par la figure `bases`,
`peano1` qui débute par la figure `baseun`,
`peano2` qui débute par la figure de `basedeux`,
Taper par exemple `peano0(-1,2,1)` pour voir l'étape 1.

```
//motif de base
bases(x,y):={
local a,b,c,d,e1,f,g,h,i1,k,L;
```

```
L:=NULL;
h:=(y-x)/3;
a:=x+h/2+i*h/2;
b:=a+i*h;
c:=b+i*h;
d:=c+h
e1:=b+h;
f:=a+h;
g:=f+h;
k:=e1+h;
i1:=d+h;
L:=L,arc(a,b,pi/2);
L:=L,arc(c,b,pi/2);
L:=L,arc(d,c,pi/2);
L:=L,arc(e1,d,pi/2);
L:=L,arc(e1,f,pi/2);
L:=L,arc(f,g,pi/2);
L:=L,arc(g,k,pi/2);
L:=L,arc(i1,k,pi/2);
return L;
}:;
//un arc et le motif de base
un(x,y):={
local h,a,L;
L:=NULL;
h:=(y-x)/3;
a:=x+h/2+i*h/2;
L:=L,arc(a-h,a,pi/2);
L:=L,bases(x,y);
return L;
}:;
//un  autre arc et le motif de base
deux(x,y):={
local h,a,L;
L:=NULL;
h:=(y-x)/3;
a:=x+h/2+i*h/2;
L:=L,arc(a,a-h*i,pi/2);
L:=L,bases(x,y);
return L;}:;

//courbe qui remplit un carre debute par le motif bases
// ex peano0(-1,2,1) ou  peano0(-1,2,3)
// utilise bases un deux peano1 peano2
peano0(x,y,n):={
local a,h,L;
if (n==0) {return bases(x,y);}
L:=NULL;
```

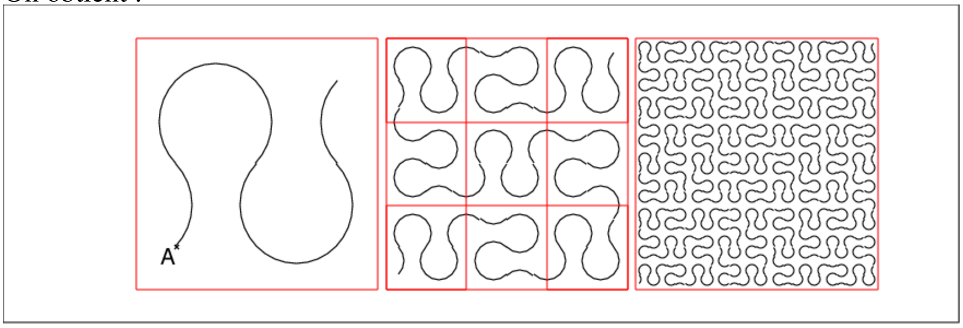
```
h:=(y-x)/3;
a:=x+h;
L:=L,peano0(x,a,n-1);
L:=L,peano2(a+i*h,a,n-1);
L:=L,peano1(a+h,y,n-1);
L:=L,peano1(y+i*h,y+2*i*h,n-1);
L:=L,peano1(a+h+2*i*h,a+2*i*h,n-1);
L:=L,peano2(a+i*h,a+2*i*h,n-1);
L:=L,peano2(x+2*i*h,a+2*i*h,n-1);
L:=L,peano2(a+3*i*h,a+2*i*h,n-1);
L:=L,peano1(a+h+2*i*h,y+2*i*h,n-1);
return L;
}:;
//courbe qui remplit un carre debute par le motif un
// ex peano1(-1,2,1)
// utilise bases un deux peano2
peano1(x,y,n):={
local a,h,L;
if (n==0) {return un(x,y);}
L:=NULL;
h:=(y-x)/3;
a:=x+h;
L:=L,peano1(x,a,n-1);
L:=L,peano2(a+i*h,a,n-1);
L:=L,peano1(a+h,y,n-1);
L:=L,peano1(y+i*h,y+2*i*h,n-1);
L:=L,peano1(a+h+2*i*h,a+2*i*h,n-1);
L:=L,peano2(a+i*h,a+2*i*h,n-1);
L:=L,peano2(x+2*i*h,a+2*i*h,n-1);
L:=L,peano2(a+3*i*h,a+2*i*h,n-1);
L:=L,peano1(a+h+2*i*h,y+2*i*h,n-1);
return L;
}:;
//courbe qui remplit un carre debute par le motif deux
// ex peano2(-1,2,1)
// utilise bases un deux peano1
peano2(x,y,n):={
local a,h,L;
if (n==0) {return deux(x,y);}
L:=NULL;h:=(y-x)/3;
a:=x+h;
L:=L,peano2(x,a,n-1);
L:=L,peano2(a+i*h,a,n-1);
L:=L,peano1(a+h,y,n-1);
L:=L,peano1(y+i*h,y+2*i*h,n-1);
L:=L,peano1(a+h+2*i*h,a+2*i*h,n-1);
L:=L,peano2(a+i*h,a+2*i*h,n-1);
L:=L,peano2(x+2*i*h,a+2*i*h,n-1);
```

```
L:=L,peano2(a+3*i*h,a+2*i*h,n-1);
L:=L,peano1(a+h+2*i*h,y+2*i*h,n-1);
return L;
}:;
```

On tape :

```
carre(-1,2,affichage=1);peano0(-1,2,0);
peano0(2+1/10,5+1/10,1);carre(2+1/10,5+1/10,affichage=1);
peano0(5+1/5,8+1/5,2);carre(5+1/5,8+1/5,affichage=1);
A:=point(-1+1/2*(1+i))
```

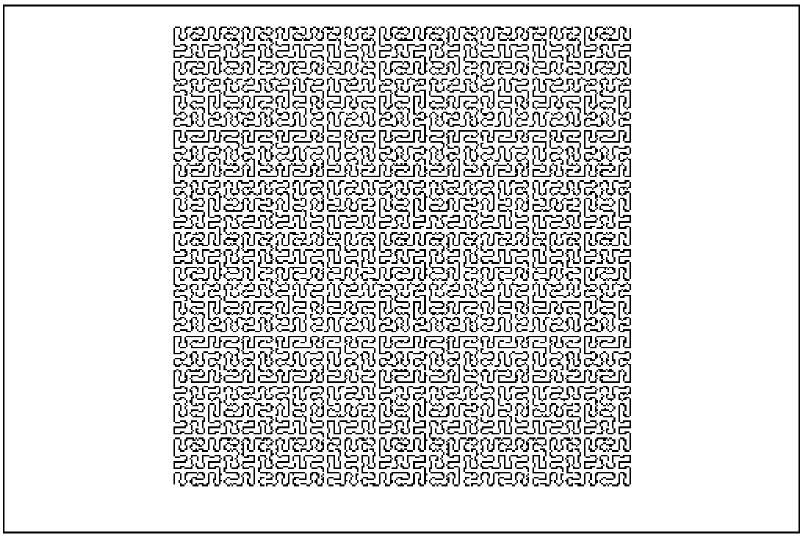
On obtient :



On tape :

```
peano0(-1,2,3)
```

On obtient :



10.5 Autres courbes : des programmes sans commentaires

10.5.1 La courbe de Hibert

Voici les programmes composées de 4 fonctions, qui dessine la courbe de Hilbert dans l'écran `DispG` :

```
//courbe de hilbert par exemple hilg(-2,0,4) est
//compos\'e par 4 morceaux hilg hild berg et berd
//ou hild(0,2,4) ou berg(2,2-2*i,4) ou berd(2,2-2*i,4)
//le morceau 1
hilg(x,y,n):={
local u,v,a,b;
DispG();
if (n==0) {segment(x,y);return 0;}
u:=(y-x)/2;
v:=u*i;
hild(x,x+v,n-1);
hilg(x+v,x+v+u,n-1);
a:=berg(x+v+u,x+u,n-1);
b:=berd(a,a+u,n-1);
};
//le morceau 2
hild(x,y,n):={
local u,v,a,b;
DispG();
if (n==0) {segment(x,y);return 0;}
u:=(y-x)/2;
v:=u*i;
hilg(x,x-v,n-1);
hild(x-v,x-v+u,n-1);
b:=berd(x-v+u,x+u,n-1);
a:=berg(b,b+u,n-1);
};
//le morceau 3
berg(x,y,n):={
local u,v,b;
DispG();
if (n==0) {segment(x,y);return y;}
v:=(x-y)/2;
u:=-v*i;
hild(x,x+v,n-1);
hilg(x+v,x+v+u,n-1);
b:=berg(x+v+u,x+u,n-1);
hild(b,b-v,n-1);
return(b-v);
};
//le morceau 4
berd(x,y,n):={
local u,v,a;
DispG();
if (n==0) {segment(x,y);return y;}
v:=(x-y)/2;
u:=-v*i;
hilg(x,x+v,n-1);
```

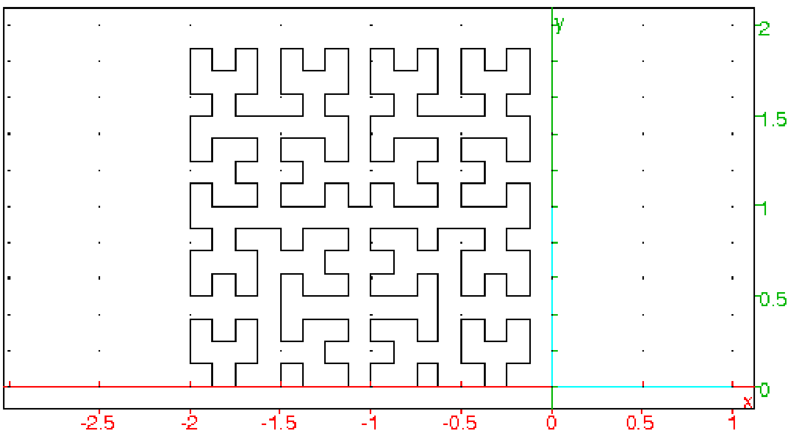
```
hild(x+v,x+v-u,n-1);
a:=berd(x+v-u,x-u,n-1);
hilg(a,a-v,n-1);
return a-v;
};
```

Voici les programmes composées de 4 fonctions, qui dessine la courbe de Hilbert dans l'écran de réponse :

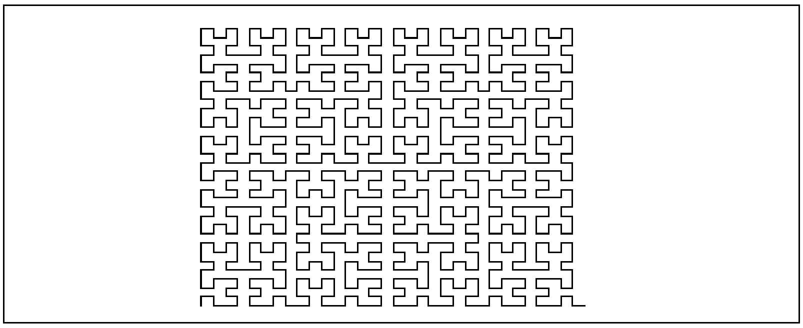
```
hilg(x,y,n):={
local u,v,a,b,L;
L:=NULL;
if (n==0) {return segment(x,y);}
u:=(y-x)/2;
v:=u*i;
L:=L,hild(x,x+v,n-1);
L:=L,hilg(x+v,x+v+u,n-1);
a:=berg(x+v+u,x+u,n-1);
L:=L,tail(a);
a:=a[0];
b:=berd(a,a+u,n-1);
L:=L,tail(b);
b:=b[0];
return L;
}:;
//le morceau 2
hild(x,y,n):={
local u,v,a,b,L;
L:=NULL;
if (n==0) {return segment(x,y);}
u:=(y-x)/2;
v:=u*i;
L:=L,hilg(x,x-v,n-1);
L:=L,hild(x-v,x-v+u,n-1);
b:=berd(x-v+u,x+u,n-1);
L:=L,tail(b);
b:=b[0];
a:=berg(b,b+u,n-1);
L:=L,tail(a);
a:=a[0];
return L;
}:;
//le morceau 3
berg(x,y,n):={
local u,v,b,L;
L:=NULL;
if (n==0) {return y,segment(x,y);}
v:=(x-y)/2;
u:=-v*i;
```

```
L:=L,hild(x,x+v,n-1);
L:=L,hilg(x+v,x+v+u,n-1);
b:=berg(x+v+u,x+u,n-1);
L:=L,tail(b);
b:=b[0];
L:=L,hild(b,b-v,n-1);
return (b-v),L;
}:;
//le morceau 4
berd(x,y,n):={
local u,v,a,L;
L:=NULL;
if (n==0) {return y,segment(x,y);}
v:=(x-y)/2;
u:=-v*i;
L:=L,hilg(x,x+v,n-1);
L:=L,hild(x+v,x+v-u,n-1);
a:=berd(x+v-u,x-u,n-1);
L:=L,tail(a);
a:=a[0];
L:=L,hilg(a,a-v,n-1);
return a-v,L;
}
:;
```

On tape :
`hilg(-2,0,4)`
On obtient :



On tape :
`hilg(-2,0,5)`
On obtient :



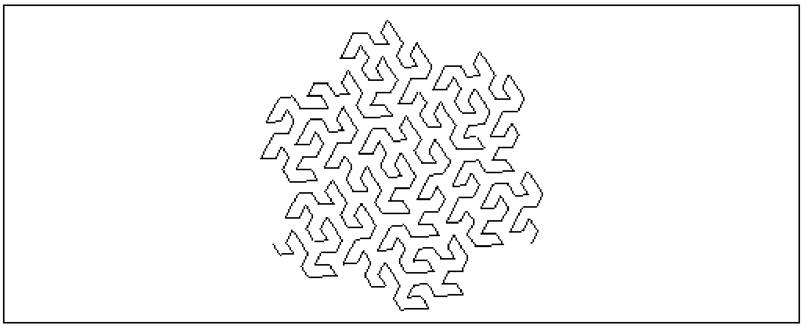
10.5.2 La courbe de Gosper

```
//gosper(-2-2*i,2-2*i,2)ou gosper(-2-2*i,2-2*i,3)
gosper(x,y,n):={
local a,b,c,d,f,g,L;
L:=NULL;
if (n==0) return segment(x,y);
a:=x+(y-x)/sqrt(7)*exp(evalf(-i*acos(5*sqrt(7)/14)));
c:=x+(a-x)*exp(evalf(i*pi/3));
b:=c+a-x;
d:=c+(a-x)*exp(evalf(2*i*pi/3));
f:=d+2*(a-x);
g:=(d+f)/2;
L:=L,gosper(x,a,n-1);
L:=L,gosper(b,a,n-1);
L:=L,gosper(c,b,n-1);
L:=L,gosper(c,d,n-1);
L:=L,gosper(d,g,n-1);
L:=L,gosper(g,f,n-1);
L:=L,gosper(y,f,n-1);
};
```

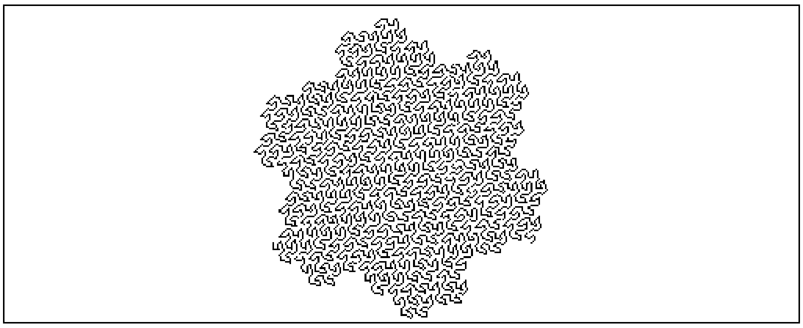
On tape :

`gosper(-2-2*i,2-2*i,3)`

On obtient :



On tape :
`gosper(-2-2*i,2-2*i,4)`
On obtient :

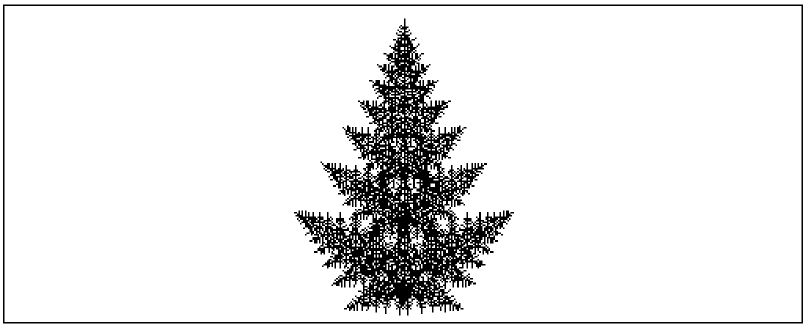


10.5.3 Des plantes et des arbres

Voici des sapins :

```
//Voici des sapins....sapin(0,2*i)
sapin(x,y):={
 local L;
 L:=NULL;
 if (abs(x-y)<0.05) {return segment(x,y);}
 L:=L,sapin(x,x+(y-x)*0.5*exp(i));
 L:=L,sapin(x,x+(y-x)*0.5*exp(-i));
 L:=L,segment(x,(3*x+y)/4);
 L:=L,sapin((3*x+y)/4,y);
return L;
}:;
```

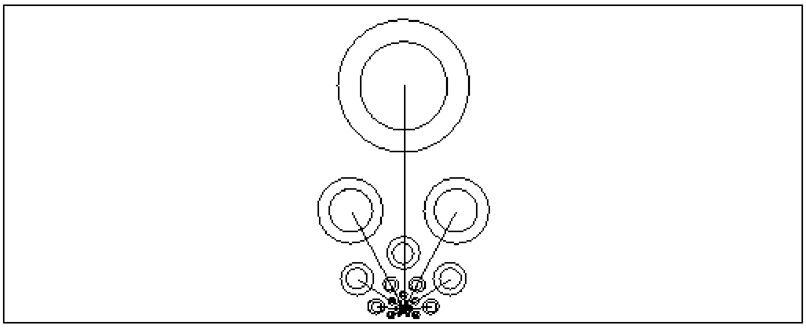
On tape : `sapin(0,2*i)`
On obtient :



Voici des fleurs :

```
//Voici des fleurs....fleur(0,2*i)
fleur(x,y):={
 local L;
 L:=NULL;
 if (abs(x-y)<0.05) {return segment(x,y),cercle(y,(y-x)*0.3);}
 L:=L,segment(x,y),cercle(y,(y-x)*0.3),cercle(y,(y-x)*0.2);
 L:=L,fleur(x,x+(y-x)*0.5*exp(i*0.5));
 L:=L,fleur(x,x+(y-x)*0.5*exp(-i*0.5));
}:;
```

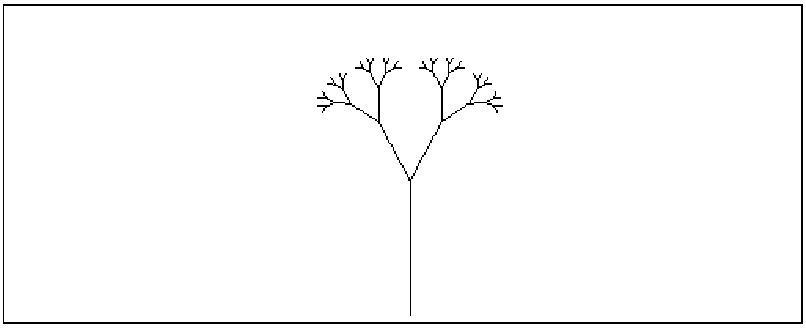
On tape : `fleur(0,2*i)`
On obtient :



Voici des arbres :

```
arbre1(x,y):={
 local L;
 L:=NULL;
 if (abs(x-y)<0.1) {return segment(x,y);}
 L:=L,segment(x,(x+y)/2);
 L:=L,arbre1((x+y)/2,(x+y)/2+(y-x)*0.5*exp(i*0.5));
 L:=L,arbre1((x+y)/2,(x+y)/2+(y-x)*0.5*exp(-i*0.5));
return L;
}:;
```

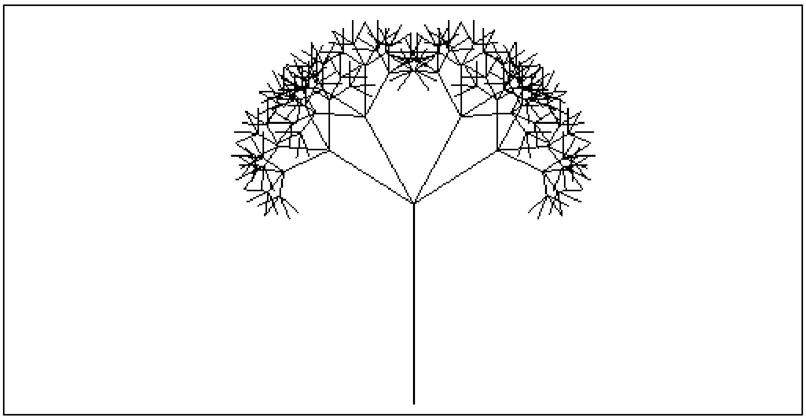
On tape : `arbre1(0,2*i)`
On obtient :



et des arbres moins déplumés :

```
arbre2(x,y):={
 local L;
 L:=NULL;
 if (abs(x-y)<0.1) {return segment(x,y);}
 L:=L,segment(x,(x+y)/2);
 L:=L,arbre2((x+y)/2,(x+y)/2+(y-x)*0.5*exp(i*0.5));
 L:=L,arbre2((x+y)/2,(x+y)/2+(y-x)*0.5*exp(-i*0.5));
 L:=L,arbre2((x+y)/2,(x+y)/2+(y-x)*0.5*exp(i));
 L:=L,arbre2((x+y)/2,(x+y)/2+(y-x)*0.5*exp(-i));
return L;
}:;
```

On tape : `arbre2(0,2*i)`
On obtient :

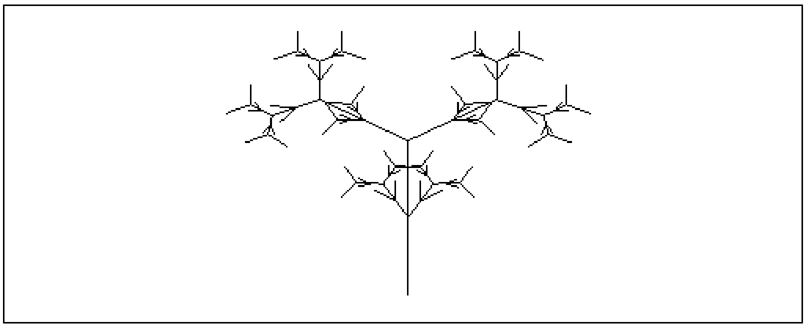


et un epineux :

```
arbre3(x,y):={
 local L;
 L:=NULL;
 if (abs(x-y)<0.1) {return segment(x,y);}
 L:=L,segment(x,(x+y)*0.5);
```

```
 L:=L,arbre3((3*x+y)/4,(3*x+y)/4+(y-x)*0.25*exp(i*0.5));
 L:=L,arbre3((3*x+y)/4,(3*x+y)/4+(y-x)*0.25*exp(-i*0.5));
 L:=L,arbre3((x+y)/2,(x+y)/2+(y-x)*0.5*exp(i));
 L:=L,arbre3((x+y)/2,(x+y)/2+(y-x)*0.5*exp(-i));
return L;
}:;
```

On tape : `arbre3(0,2*i)`
On obtient :



Voici des fougères :

```
//une fougere par ex fougere(-2*i,2*i)
fougere(x,y):={
local a,L;
L:=NULL;
 if (abs(x-y)<0.2) {return segment(x,y);}
a:=x+(y-x)*0.15*exp(-i*0.2);
L:=L,segment(x,a);
L:=L,fougere(a,a+(y-x)*0.33*exp(i*1.2));
L:=L,fougere(a,a+(y-x)*0.33*exp(-i*1.2));
L:=L,fougere(a,a+(y-x)*0.85*exp(-i*0.2));
return L;
}:;
//par ex fougeres(-2*i,2*i,0.05,6)
fougeres(x,y,t,n):={
local a,L;
if (n==0) {return segment(x,(x+y)/2);}
//a:=x+(y-x)*0.15*exp(-i*t);
a:=x+(y-x)*0.15;
L:=NULL;
L:=L,segment(x,a);
L:=L,fougeres(a,a+(y-x)*0.33*exp(i*1.2),t,n-1);
L:=L,fougeres(a,a+(y-x)*0.33*exp(-i*1.2),t,n-1);
L:=L,fougeres(a,a+(y-x)*0.85*exp(-i*t),t,n-1);
return L;
}:;
```

On tape : `fougere(-2*i,2*i)`
On obtient :

On tape : `fougeres(-2*i,2*i,0.05,6)`
On obtient :

Et enfin, le bouquet final :

```
//et le bouquet final bouquet(0,2*i)
bouquet(x,y):={
 local L;
 L:=NULL;
 if (abs(x-y)<0.1) {return segment(x,y);}
 L:=L,segment(x,(x+y)*0.5);
 L:=L,bouquet((3*x+y)/4,(3*x+y)/4+(y-x)*0.25*exp(i*0.5));
 L:=L,bouquet((3*x+y)/4,(3*x+y)/4+(y-x)*0.25*exp(-i*0.5));
 L:=L,bouquet((x+y)/2,(x+y)/2+(y-x)*0.5*exp(i));
 L:=L,bouquet((x+y)/2,(x+y)/2+(y-x)*0.5*exp(-i));
 L:=L,bouquet((x+y)/2,(x+y)/2+(y-x)*0.5);
```

```
 return L;
}:;
```

On tape : `bouquet(0,2*i)`
On obtient :

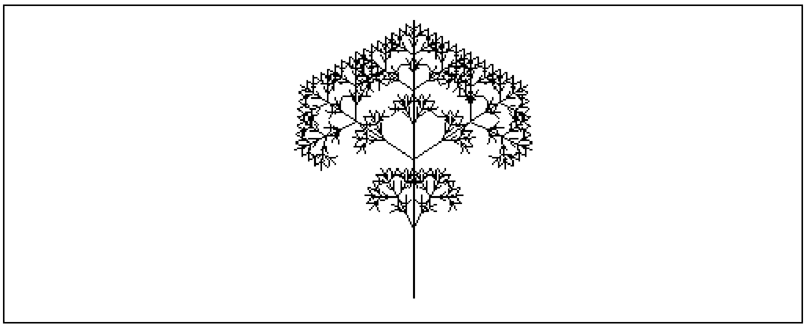


Table des matières