

Extract of Les nouvelles technologies pour l'enseignement des mathématiques

<http://revue.sesamath.net/spip.php?article293>

Calcul formel et géométrie dynamique : comment les associer ?

- N°23 - Janvier 2011 -

Publication date: dimanche 12 décembre 2010

Description:

La géométrie dynamique apporte le graphisme au calcul formel, mais le calcul formel apporte la précision à la géométrie. Comment associer les deux ?

Copyright © Les nouvelles technologies pour l'enseignement des
mathématiques - Tous droits réservés

CAS dans la géométrie

Une première solution consiste à choisir un logiciel de calcul formel (CAS en anglais, pour *Computer Algebra System*) et à l'incorporer au logiciel de géométrie dynamique. Ce qui est possible

- techniquement si les deux logiciels sont écrits dans le même langage de programmation (par exemple, [java \(langage\)](#));
- légalement, si les deux logiciels sont libres (par exemple tous les deux sous la license GPL).

C'est le cas (le CAS ?) avec GeoGebra, incorporant le logiciel *Yacas*, porté en java sous le nom *MathPiper*. La puissance du calcul formel est alors disponible avec la convivialité de GeoGebra :

Exemple 1 : Les fractions

Le fichier ci-dessous a été assez rapide à fabriquer : Deux curseurs nommés *a* et *b*, pour chacun desquels on a utilisé la fonction *FractionTexte()* et il a suffi de rajouter deux textes, représentant respectivement le "+" et le "=" :

```
<applet name="ggbApplet" code="geogebra.GeoGebraApplet" codebase="."/>
archive="http://www.geogebra.org/webstart/geogebra.jar" width="727" height="415"> D'ailleurs, l'activité
GeoGebra ne peut pas démarrer. Assurez-vous que Java 1.4.2 (ou version supérieure) est installée et
activée sur votre navigateur (Cliquez ici pour installer Java maintenant !)
Créez avec GeoGebra
```

C'est bien entendu en manipulant les curseurs qu'on voit l'intérêt que présente la géométrie dynamique dans cet exercice de pur calcul formel !

Cependant, la combinaison de l'utilisation des [curseurs dans un tableur](#) et de la barre d'outils [CmathOOoCAS](#) permet, sous Open Office, de faire la même chose, et [même bien plus](#). C'est en effet l'aspect "engagement direct" qui est utile ici, et celui-ci n'est pas propre à la géométrie dynamique.

Exemple 2 : Factorisation de trinomes

Ce fichier GeoGebra est très classique : Trois curseurs *a*, *b* et *c*, et la fonction

```
f(x)=a*x^2+b*x+c
```

pour avoir la parabole.

Sauf que, à la place, on a entré

```
f(x)=Factoriser[a*x^2+b*x+c]
```

Ce qui change l'affichage de $f(x)$ tel qu'il apparaît dans la "fenêtre algèbre" (à gauche).

```
<applet name="ggbApplet" code="geogebra.GeoGebraApplet" codebase="."/>
archive="http://www.geogebra.org/webstart/geogebra.jar" width="727" height="415"> D'ailleurs, l'activité
GeoGebra ne peut pas démarrer. Assurez-vous que Java 1.4.2 (ou version supérieure) est installée et
activée sur votre navigateur (Cliquez ici pour installer Java maintenant !)
```

Créé avec [GeoGebra](#)

La partie intéressante de la manipulation consiste à triturer les curseurs et regarder l'effet produit, en particulier pourquoi, alors que x^2-1 est factorisé en $(x+1)(x-1)$, $x^2-0,81$ n'est pas factorisé...

Pourtant, en entrant

```
Développer[(x-0.9)*(x+0.9)]
```

on a bien $x^2-0,81$...

Exemple 3 : Analyse

Après avoir créé deux curseurs a et b , on a tapé très classiquement

```
f(x)=a*x^3+b*x
```

pour avoir un représentant d'une famille de fonctions à deux paramètres ; mais ensuite il a suffi d'entrer

```
Dérivée[f]
```

et

```
g(x)=Intégrale[f]
```

pour voir, non seulement les représentations graphiques de la dérivée et de l'intégrale de f , mais également leur expression algébrique (non réduite cependant) :

```
<applet name="ggbApplet" code="geogebra.GeoGebraApplet" codebase="/"  
archive="http://www.geogebra.org/webstart/geogebra.jar" width="727" height="415">
```

Désolé, l'activité GeoGebra ne peut pas démarrer. Assurez-vous que Java 1.4.2 (ou version supérieure) est installée et activée sur votre navigateur ([Cliquez ici pour installer Java maintenant !](#))

Créé avec [GeoGebra](#)

Exemple 4 : Notations ensemblistes

Pour un exemple de ce que permet GeoGebra en matière de réunion, intersection et affichage d'ensembles finis, voir [ce cours de Seconde en ligne](#) (en milieu de page). La figure est dynamique à cause des cases booléennes à cocher pour faire varier les ensembles (ici des événements).

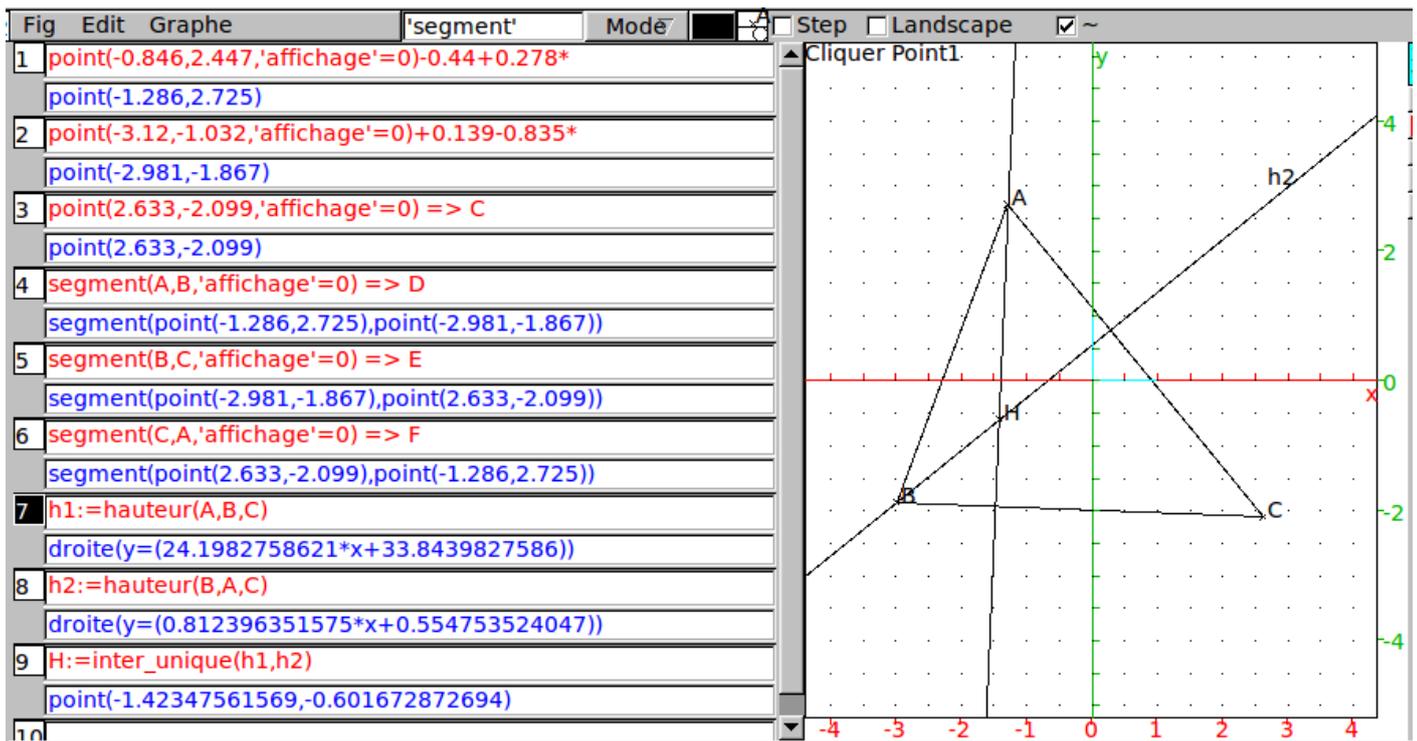
++++

Géométrie dans un CAS

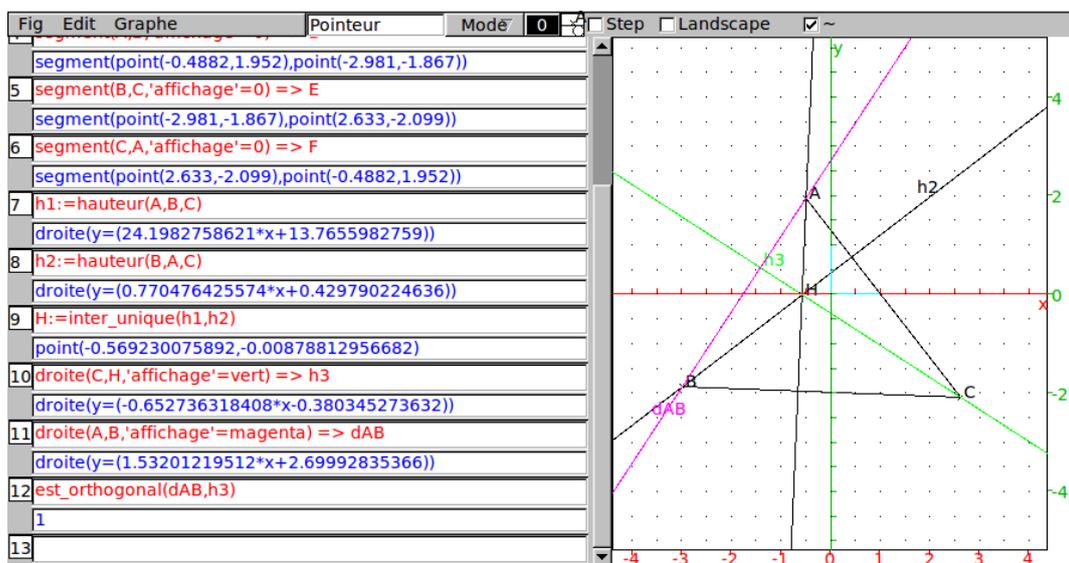
L'opération inverse de celle de l'onglet précédent, c'est celle consistant à incorporer un moteur de géométrie dynamique à l'interface d'un logiciel de calcul formel. C'est ce qu'a fait Bernard Parisse avec [Xcas](#).

Orthocentre

Pour vérifier, puis *démontrer*, que les trois hauteurs d'un triangle sont concourantes, on peut, sous Xcas, créer un affichage 2D, puis créer par clics de souris, les points A, B et C mobiles, puis les segments dessinant le triangle ABC



Après, c'est au clavier qu'on entre "h1 :=hauteur(A,B,C)" pour obtenir la hauteur issue de A, et, de façon analogue, la hauteur h2 issue de B. C'est aussi au clavier qu'on a créé leur intersection H, afin de vérifier que (CH) est perpendiculaire à (AB). D'ailleurs, il suffit, une fois qu'on a créé la droite (AB) (qui ci-dessous s'appelle dAB), d'un test d'orthogonalité :

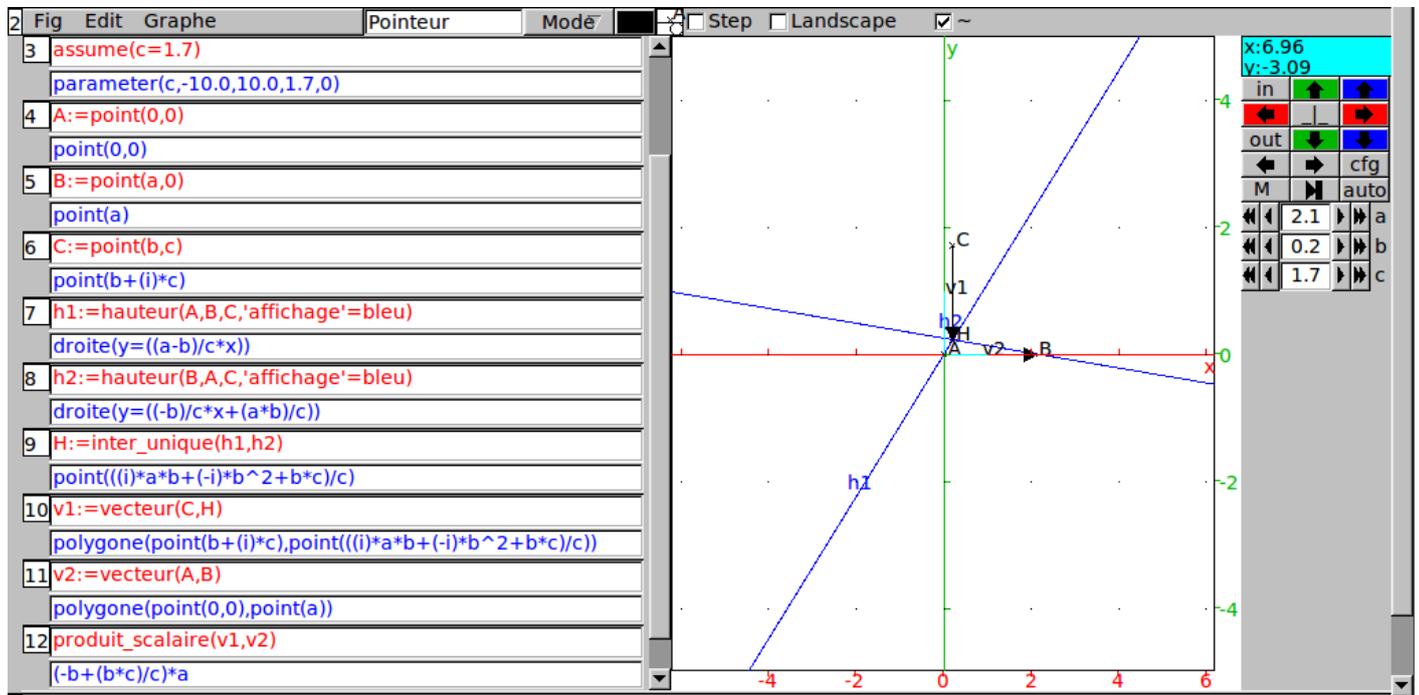


Cependant, comme on le voit sur les équations des droites de la figure, les calculs sont approchés, et pas formels. On fait alors de la géométrie dynamique comme avec n'importe quel logiciel de géométrie dynamique...

En fait on peut avoir du calcul formel dans la figure, mais en manipulant A, B et C par des curseurs. On crée trois

Calcul formel et géométrie dynamique : comment les associer ?

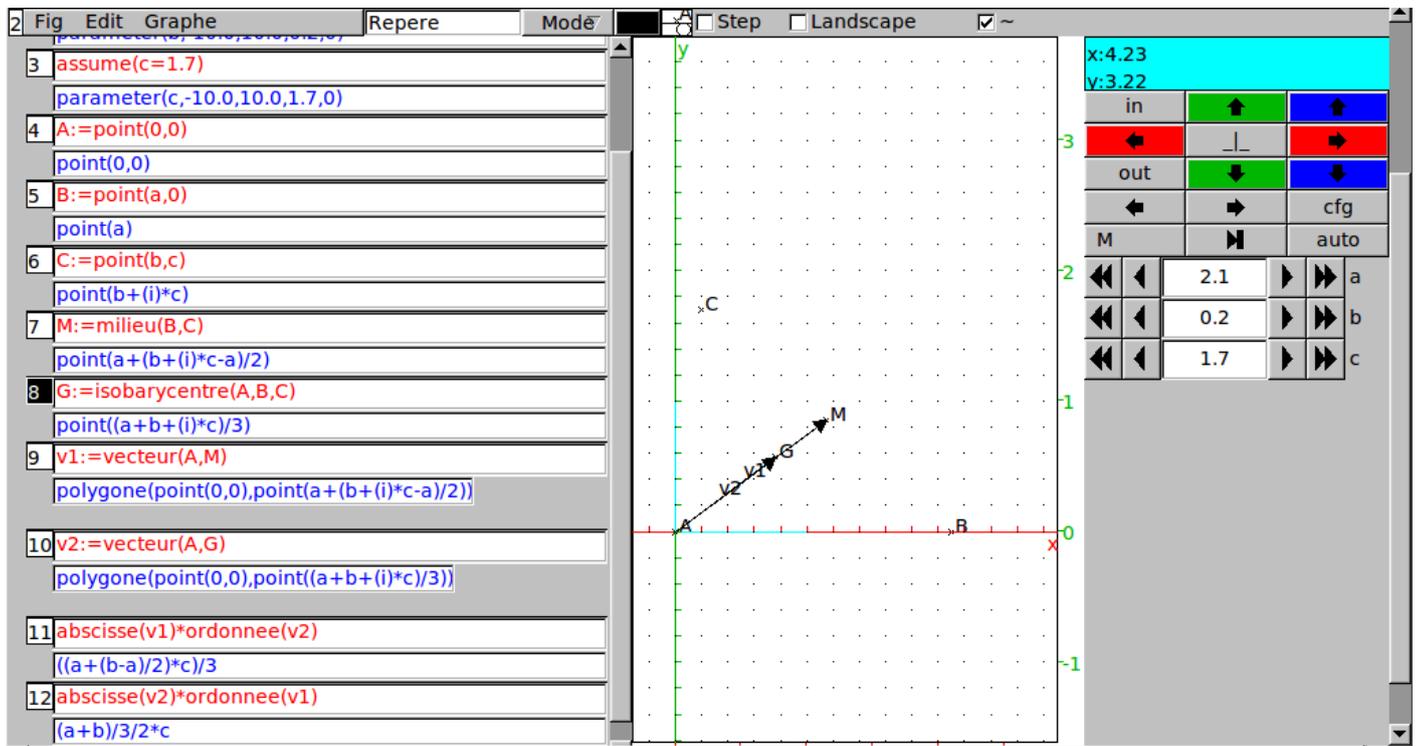
nombre réels a , b et c par "assume($a=3.3$)" etc., ce qui crée immédiatement les curseurs à droite de la figure. Puis les points $A(0;0)$, $B(a,0)$ et $C(b,c)$. De cette manière les coordonnées des points A , B et C ne sont pas numériques et on peut continuer à faire du calcul formel dans la figure. Pour prouver que la droite (CH) est une hauteur par du calcul formel, on définit les vecteurs \overrightarrow{AB} et \overrightarrow{CH} , et on calcule leur produit scalaire :



Centre de gravité

Pour montrer que le centre de gravité du triangle ABC est aligné avec A et le milieu M de $[BC]$, on peut de même utiliser le calcul formel, en calculant des produits en croix, ce qui est faisable en Seconde : Comme ci-dessus, on définit $A(0;0)$, $B(a,0)$ et $C(b,c)$, puis M (milieu de $[BC]$) et G (centre de gravité) par des saisies au clavier, puis, toujours au clavier, les vecteurs \overrightarrow{AM} et \overrightarrow{AG} :

Calcul formel et géométrie dynamique : comment les associer ?



Ensuite, il suffit de calculer les deux produits en croix pour vérifier qu'ils sont égaux, et donc que les deux vecteurs sont colinéaires. Ceci dit, l'égalité des deux produits ci-dessus ne saute peut-être pas aux yeux, il suffit dans ce cas de demander à Xcas de les simplifier :

```
11 simplifier(abcisse(v1)*ordonnee(v2))
(a*c+b*c)/6
12 simplifier(abcisse(v2)*ordonnee(v1))
(a*c+b*c)/6
```

Hyperbole équilatère et orthocentre

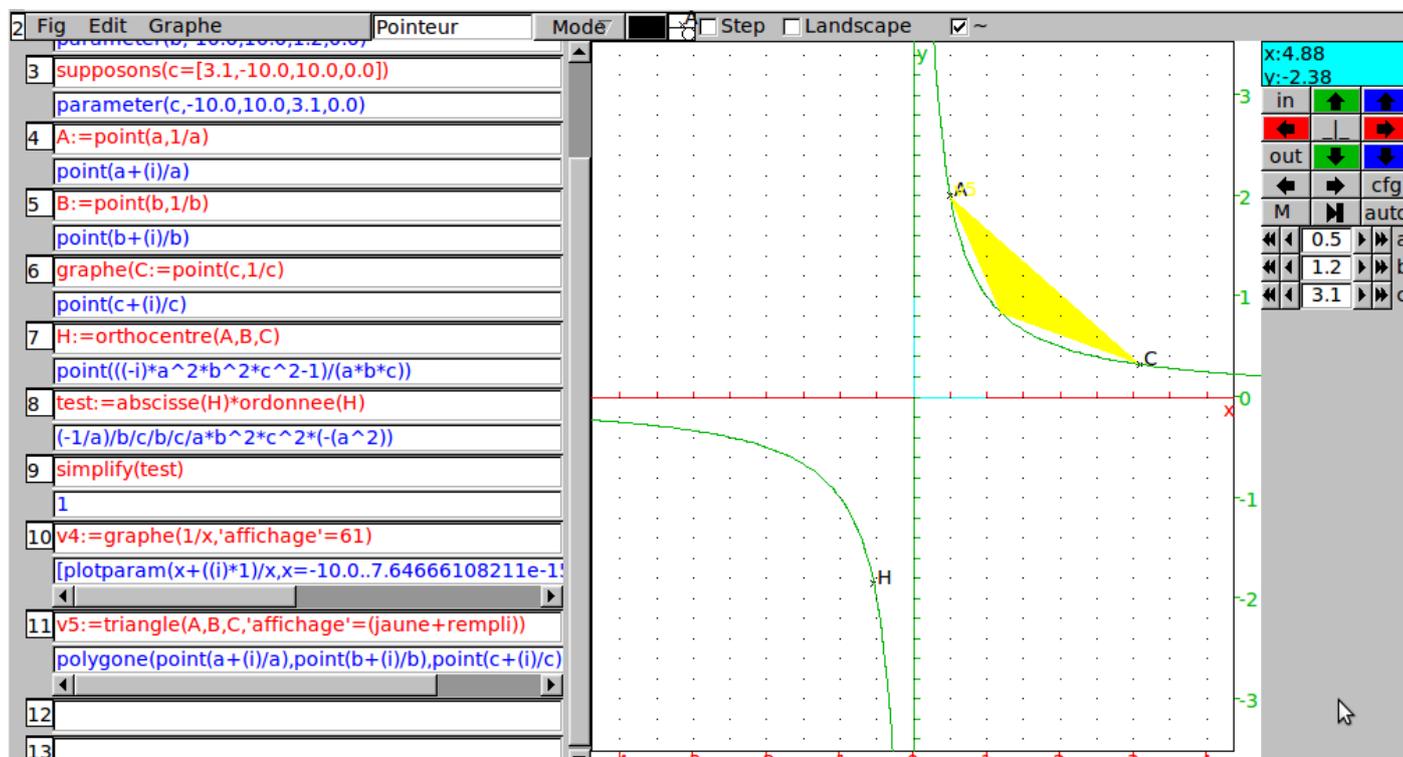
Théorème

[1] :
L'orthocentre H d'un triangle ABC inscrit dans une hyperbole équilatère est lui-même sur l'hyperbole équilatère.

Pour vérifier (et prouver) cela, on peut aussi utiliser Xcas. Un premier problème se pose à propos de l'hyperbole

Calcul formel et géométrie dynamique : comment les associer ?

équilatère, Xcas n'étant capable de construire une hyperbole qu'à partir de ses foyers, ce qui complique la manipulation. De plus, un point ne peut être lié qu'à une droite, un segment ou un cercle. Alors on crée trois réels a , b et c par des curseurs, et on donne aux points A , B et C les coordonnées $A\left(a; \frac{1}{a}\right)$, $B\left(b; \frac{1}{b}\right)$ et $C\left(c; \frac{1}{c}\right)$. Alors, en créant au clavier l'orthocentre H de ABC , on apprend (en lisant son affixe) que ses coordonnées sont $H\left(\frac{-1}{abc}; -abc\right)$:



Si on n'arrive pas à lire les coordonnées de H ou s'il n'est pas évident que leur produit vaut 1, on peut toujours faire effectuer ce produit par Xcas, comme ci-dessus, ce qui a nécessité une simplification.

Les exemples fournis avec Xcas, en général de niveau Terminale S, illustrent également très bien comment le calcul formel permet des démonstrations, en particulier avec les nombres complexes.

++++

Communication entre logiciels

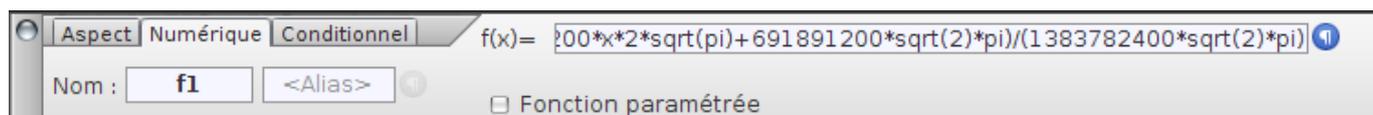
Par suite d'une certaine uniformité dans les langages utilisés par les logiciels pour représenter les expressions algébriques, on peut faire du copier-coller entre Maxima, Xcas, GeoGebra et CaRMetal presque sans problème. Par exemple, calculer une expression formelle avec un logiciel comme Maxima ou Xcas, et la récupérer dans un logiciel de géométrie dynamique pour illustrer une figure. Un [atelier APMEP](#) a déjà été consacré à la communication entre Maxima et CaRMetal. Voici en complément, un autre exemple :

Par exemple, on souhaite manipuler dans CaRMetal la représentation graphique de $\Pi(x)$, définie par $\Pi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ (fonction de répartition d'une variable aléatoire normale centrée réduite), par exemple pour construire un intervalle de confiance. Mais cette fonction n'est pas exprimable avec des fonctions usuelles, on ne peut l'obtenir que par intégration numérique. Alors, faute de mieux, on va la remplacer par son [approximant de Padé](#) à l'ordre 16, calculé en un tour de main par Xcas :

```
2 pade(normal_cdf(0,1,x),x,16,16)
-143 * x15 * sqrt(pi + 2310 * x13 * sqrt(pi - 32760 * x11 * sqrt(pi + 400400 * x9 * sqrt(pi - 4118400 * x7 * sqrt(pi + 34594560 * x5 * sqrt(pi - 230630400 * x3 *
1383782400 * sqrt(2) * pi
```

(on constate en passant que l'approximant est un polynôme)

Le miracle, c'est qu'il suffit de sélectionner sous Xcas l'expression obtenue avec ses racines et ses π , puis "copier". Ensuite, sous CaRMetal, créer la représentation graphique d'une fonction, et effectuer un "coller" dans le cadre réservé à l'expression de la fonction :



Voilà ! C'est aussi simple qu'un copier-coller ! Le miracle, c'est que les auteurs de tous ces logiciels ne se sont pas concertés, et qu'il y a donc convergence de la syntaxe des différents logiciels (à peu de chose près, le logiciel Maxima rajoutant des "%" devant les "pi" ce qui, sur cet exemple précis, le rend moins maniable).

Ensuite il suffit d'ajouter quelques objets propres à la géométrie dynamique comme les flèches manipulables à l'écran et le fameux [LaTeX](#) dynamique de [CaRMetal](#) pour obtenir cette figure :

```
<applet archive="plugins/auto/eva_geometrie/scripts/CaRMetal.jar" code="Main.class" width="808" height="531" mayscript="true">
```

Un exemple de communication dans les deux sens entre Maxima et CaRMetal [a déjà été publié](#). Et l'atelier de l'APMEP cité ci-dessus est relaté [ici](#).

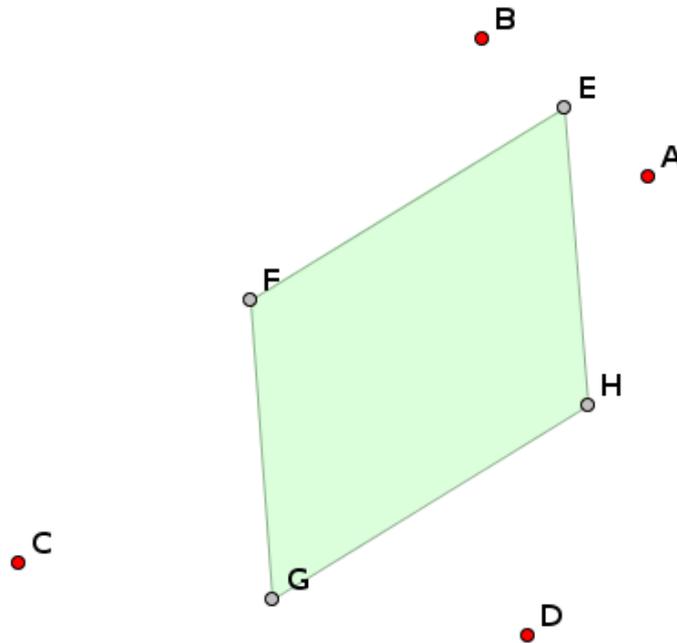
++++
Varignon

Le logiciel [JGex](#) ressemble à un logiciel de géométrie dynamique en [Java \(langage\)](#) comme les autres (opinion personnelle : plutôt moins bon que les autres) mais il possède des [bases de Gröbner](#) qui lui permettent de faire du calcul formel puisqu'il s'agit de polynômes. Le principe est de travailler dans un repère où chaque contrainte géométrique se traduit par une équation polynomiale, et par résolution du système formé, de trouver de nouvelles égalités qui se traduisent par des propriétés géométriques de la figure. JGex permet donc à la fois d'émettre automatiquement des conjectures, et de démontrer des propriétés de la figure. L'exemple suivant a été traité en Seconde, en module d'apprentissage de la rédaction :

Théorème(Varignon) :
 Dans un quadrilatère quelconque ABCD, le quadrilatère formé par les milieux des côtés d'ABCD est un parallélogramme.

Comme les meilleurs élèves de Seconde, JGex est capable de démontrer ce théorème. Pour cela on construit le

quadrilatère ABCD et les milieux de ses côtés :



Première méthode : algébriquement

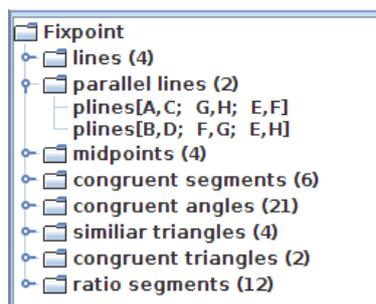
En choisissant (A,B,C) comme repère, on a $A(0;0)$, $B(1;0)$, $C(0;1)$, $D(x,y)$ et donc

$E\left(\frac{1}{2};0\right)$, $F\left(\frac{1}{2};\frac{1}{2}\right)$, $G\left(\frac{x}{2};\frac{y+1}{2}\right)$ et $H\left(\frac{x}{2};\frac{y}{2}\right)$. Par calcul on en déduit

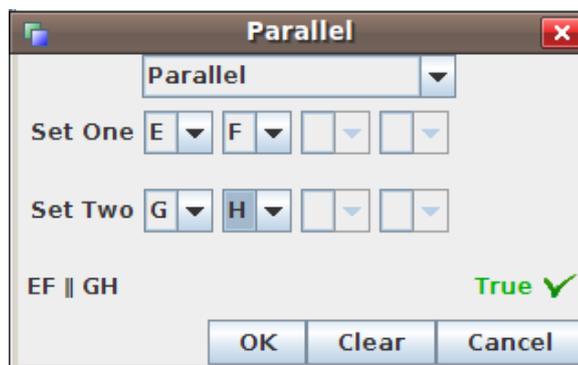
$\overrightarrow{EF}\left(\begin{array}{r} 0 \\ \frac{1}{2} \end{array}\right)$ et $\overrightarrow{HG}\left(\begin{array}{r} 0 \\ \frac{1}{2} \end{array}\right)$:

$\overrightarrow{EF}=\overrightarrow{HG}$ cqfd.

Or JGex "sait" que EFGH est un parallélogramme, en effet à chaque étape de la construction, sa [base de données](#) s'agrandit non seulement des nouveaux objets, mais des théorèmes qu'il démontre à partir des hypothèses de la construction. En sélectionnant cette base de données ("FixPoint"), on voit 21 égalités d'angles, 4 paires de triangles semblables mais aussi, ce qui n'était pas vraiment prévu, des droites parallèles. En y regardant de plus près :



on constate (au cas où on ne l'aurait pas vu sur la figure) qu'EFGH est effectivement un parallélogramme. Mais pour obtenir la preuve, on doit demander explicitement à JGex de prouver que c'est un parallélogramme, ce qui n'a pas été prévu, le mot "parallelogramm" ne figurant pas dans le menu des choses à prouver. On peut alors le faire en deux temps, en montrant que $(EF) \parallel (GH)$ puis en montrant de même que $(EH) \parallel (FG)$. Alors on sélectionne l'entrée "parallel" dans le menu "to prove", et on entre les noms des 4 points :



On constate que le mot "true" apparaît automatiquement, en vert.

Après ça il suffit de demander la démonstration, à choisir parmi 4 méthodes :

Le choix des bases de Gröbner donne l'affichage suivant (avec un choix du repère quelque peu différent) :

```

The Equational Hypotheses:
1: E : midpoint( BA)
2x10 - x4 = 0
2: F : midpoint( CB)
2x12 - x6 - x4 = 0
2x11 - x5 = 0
3: G : midpoint( DC)
2x14 - x8 - x6 = 0
2x13 - x7 - x5 = 0
4: H : midpoint( AD)
2x16 - x8 = 0
2x15 - x7 = 0

The Initial Polynomial Set
2x16 - x8 = 0
2x15 - x7 = 0
2x14 - x8 - x6 = 0
2x13 - x7 - x5 = 0
2x12 - x6 - x4 = 0
2x11 - x5 = 0
2x10 - x4 = 0

The Groebner Basis: GB =
2x16 - x8 = 0
2x15 - x7 = 0
2x14 - x8 - x6 = 0
2x13 - x7 - x5 = 0
2x12 - x6 - x4 = 0
2x11 - x5 = 0
2x10 - x4 = 0
x9 = 0

The Nondegenerate Conditions

The Conclusion:
EF || GH
x11x16 + (-x12 + x10)x15 - x11x14 + (x12 - x10)x13 = 0
    
```

On lit (avec une orthographe quelque peu douteuse) que le parallélisme à prouver se traduit par une équation du second degré à 6 variables, et la réduction du premier membre (non montrée ci-dessus) donne 0, ce qui achève la

démonstration.

Deuxième méthode : Par le *modus ponens*

En sélectionnant la méthode *GDD* (déductive), GJex affiche presque instantanément

♀	1. EF \parallel GH 2. EF \parallel AC 3. GH \parallel AC
♀	2. EF \parallel AC (r35) midp(F, CB) (by HYP) midp(E, BA) (by HYP)
♀	3. GH \parallel AC (r35) midp(H, AD) (by HYP) midp(G, DC) (by HYP)

qui devrait être rédigée de bas en haut (en fait, c'est un algorithme, qui, à partir de faits à prouver, élabore une stratégie de déduction, de façon récursive, en amener de nouveaux faits à démontrer). Pour comprendre le sens de cette démonstration, on peut cliquer dans la liste des "lemmes" sur la règle 35, ce qui a pour effet d'afficher une figure, avec l'énoncé du [théorème des milieux](#).

Un dernier exemple en guise de conclusion, mais surtout pour montrer que l'association entre la géométrie dynamique et le calcul formel en est à ses premiers balbutiements : Pour explorer la [trigonométrie rationnelle](#), [René Grothmann](#) a introduit dans son logiciel [CaR](#) un affichage de fractions.

[1] Voir par exemple le [Bulletin Vert 487 de l'APMEP](#).