GeoGebra
Fonctionnalités de Raisonnement automatisé

Un tutoriel.

Zoltán Kovács, Tomás Recio et M. Pilar Vélez

1 Introduction.

Le logiciel GeoGebra (`https://www.geogebra.org`) est capable d'épauler l'enseignement des théorèmes d'Euclide en géométrie plane à l'aide de calculs symboliques. Certaines fonctionnalités pour la validation automatique et la découverte de théorèmes géométriques ont été développées pour cela.
Les nouvelles technologies sont encore en phase expérimentale dans les salles de classe. Ce document résume les possibilités techniques en présentant quelques exemples.

2 Démarrer en GeoGebra.

GeoGebra est utilisable sur de nombreuses plateformes, à savoir :

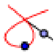
- ordinateurs de bureau ou portables avec différents systèmes d'exploitation installés ;
- tablettes, et
- téléphones intelligents (smartphones).

Les appliquettes GeoGebra peuvent être imbriquées dans des pages web, l'espace dédié étant la plateforme « Espace Ressources GeoGebra » (`https://www.geogebra.org/materials/`) proposant des millions de ressources pédagogiques disponibles gratuitement.

Les outils disponibles sur les différentes plateformes peuvent cependant être différents. De même, l'expérience de l'utilisateur sur les différentes plateformes peut être différente : les calculs symboliques peuvent nécessiter une quantité élevée de calculs et les composants du matériel ou les ressources du logiciel peuvent ne pas prendre en charge complètement certaines étapes.
Le support informatique utilisé en classe peut varier. Les résultats les plus rapides peuvent être obtenus par les ordinateurs de bureau rapides (ou portables), mais dans ce cas le logiciel doit être téléchargé et installé par l'utilisateur. Quelques exemples de ce tutoriel ne peuvent fonctionner uniquement que dans la version « bureau » (créée pour les ordinateurs de bureau et les portables, avec système d'exploitation Microsoft Windows, Apple Macintosh et Linux). D'autre part, la version "web" ne nécessite pas d'installation par l'utilisateur : elle fonctionnera dans un navigateur "web" moderne, et l'enseignant pourra préparer une liste d'exemples avec des appliquettes GeoGebra à l'avance avant l'utilisation en classe via, par exemple, l'« Espace Ressources GeoGebra ». La version "web" est cependant plus lente : les calculs symboliques peuvent être extrêmement lents.
GeoGebra fonctionne aussi, depuis peu, sur tablettes et smartphones. Dans certains cas, ces plateformes fournissent une expérience utilisateur plus rapide que la version "web", mais la taille d'écran plus petite peut empêcher les utilisateurs d'étudier les théorèmes géométriques en détail. Les enseignants sont encouragés à faire des expériences en utilisant ces types de dispositifs modernes, mais leur utilisation pour le raisonnement automatisé est encore expérimentale.
Il y a un travail continu sur les fonctionnalités de raisonnement automatisé de GeoGebra. Une habitude conseillée est de toujours utiliser la dernière version. Il faut tabler sur une mise à jour hebdomadaire pour toutes les versions, sauf pour la version Mac App Store pour laquelle le rythme de mise à jour est mensuel. La liste des modifications récentes est consultable à l'adresse `http://dev.geogebra.org/trac/timeline` destinée essentiellement aux utilisateurs avancés et aux développeurs.

3 Fonctionnalités de Raisonnement automatisé.

Les fonctionnalités de raisonnement automatisé sont une collection d'outils et commandes GeoGebra prêts à conjecturer, découvrir, affiner et prouver des résultats géométriques dans une construction géométrique dynamique.
Tout d'abord, l'utilisateur doit construire une figure géométrique en utilisant certains outils répertoriés par défaut au sommet de la fenêtre « Graphique » de GeoGebra. Après, GeoGebra peut, de nombreuses façons, mettre en avant la recherche des propriétés géométriques par divers outils et paramètres :

1. En déplaçant les objets libres, les objets qui en dépendent sont visualisés ;
2. L'outil  *Relation* aide à comparer des objets et/ou à obtenir des relations les reliant ;
3. En activant ou non la trace d'un objet construit, son mouvement sera visualisé quand ses parents sont modifiés ;
4. L'outil  *Lieu* affiche la trace d'un objet pour toutes les positions possibles d'un de ses parents (se déplaçant sur un chemin) ;
5. En validant les commandes `Relation` ou `Lieu` dans le champ de saisie de GeoGebra des informations plus précises peuvent être obtenues.

Ces méthodes sont généralement bien connues par la communauté GeoGebra, et donc elles sont bien documentées et de nombreux exemples peuvent être trouvés sur l'« Espace Ressources GeoGebra » . En particulier, on peut y consulter mon GgbBook « Recueil de fichiers utilisant Relation ». D'autre part, GeoGebra propose actuellement des fonctionnalités de raisonnement automatisé symboliques pour généraliser les propriétés géométriques observées / conjecturées :

1. Les outil et commande `Relation` peuvent être utilisés pour recalculer les résultats symboliquement ;
2. La commande `EquationLieu` complète le résultat de la commande `Lieu` en affichant l'équation de sa réponse graphique ;
3. La commande `EquationLieu` peut s'appliquer sur des lieux implicites ;
4. La commande `Enveloppe` calcule l'équation d'une courbe tangente à un ensemble d'objets lorsqu'un certain parent de l'objet se déplace sur un chemin.

3.1 Fonctionnalités de « bas / haut niveau ».

GeoGebra fournit les méthodes «de haut niveau» ci-dessus pour approfondir l'étude des théorèmes géométriques. Les outils, par la présence de leurs icônes, sont considérés comme «de haut niveau» du fait de leur facilité d'utilisation et ainsi ils peuvent être montrés directement dans les salles de classe. Ils permettent aussi d'autres façons d'en apprendre davantage sur l'environnement mathématique ou simplement pour aider à dépanner. Les méthodes «de bas niveau» sont listées en Annexe, et ne sont pas conseillées pour un usage direct en présence d'étudiants.
Évidemment, certaines des méthodes énumérées sont plus faciles, et d'autres sont plus difficiles. L'utilisation en ligne de commande dans le champ de saisie de GeoGebra peut être considérée comme un moyen plus difficile pour la plupart des utilisateurs. Il peut être conseillé à un enseignant de montrer d'abord les méthodes les plus faciles, et de montrer, plus tard, les autres manières quand les élèves ont fait assez d'expérimentations .

3.2 Fonctionnalités à support symbolique.

Comme mentionné ci-dessus quelques fonctionnalités de raisonnement automatisé sont assises sur un support symbolique. Cette caractéristique permet de vérifier d'une manière mathématiquement rigoureuse les énoncés de géométrie élémentaire qui ont été conjecturés par l'utilisateur.
Une remarque générale pour l'utilisateur est de démarrer GeoGebra en « Calculatrice Graphique ». Cela revient à afficher les étiquettes sur chaque nouvel objet ajouté - ce qui peut être crucial pour les outil et commande `Relation` lors de l'élaboration de résultats pour diverses configurations.

Dans la plupart des situations, les axes ne sont pas nécessaires : leur affichage peut être désactivé lorsque l'outil *Déplacer* est actif (c'est l'icône la plus à gauche qui montre un curseur de flèche), par exemple, en cliquant avec le bouton droit de la souris dans « *Graphique* » puis désactivant « *Axes* ».

Dans certains cas, « *Algèbre* » ne doit pas nécessairement être affichée - sauf si les équations des courbes implicites sont à étudier en détail, cela peut cependant se faire aussi en changeant l'étiquette de l'objet pour en afficher la valeur. (Pour ce faire, en cliquant avec le bouton droit de la souris sur l'objet, en choisissant « *Propriétés...* », l'utilisateur doit affecter « *Valeur* » à « *Afficher l'étiquette* » dans l'onglet « *Basique* »).
Les appliquettes GeoGebra peuvent être utilisées aisément si elles sont téléversées sur l'« Espace Ressources GeoGebra ». Si « *Algèbre* » est affichée, il peut être judicieux d'augmenter sa largeur avant de téléverser une appliquette sur l'« Espace Ressources GeoGebra ». Sinon, il ne sera pas facile à l'utilisateur de taper la commande appropriée. Après le téléversement, dans l'« Espace Ressources GeoGebra », il est conseillé que, dans les « *Paramètres avancés …* » pour l'appliquette, les options « *Afficher la Barre d'Outils* » et « *Afficher le champ de saisie* » soient activées. La définition d'une taille appropriée peut être aussi obligatoire.

3.2.1 Les outil et commande Relation.

Les outil et commande *Relation* de GeoGebra affiche un message indiquant à l'utilisateur des informations sur la relation entre deux ou plusieurs objets. (NdLN : Le vocable « ligne » désigne segment, demi-droites ou droites.). Ceux ci permettent à l'utilisateur de vérifier numériquement (c'est-à-dire, pour la construction de dessin avec les coordonnées assignées) si

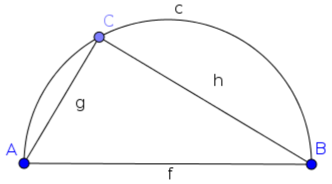
- deux lignes sont perpendiculaires ;
- deux lignes sont parallèles ;
- deux (parfois plusieurs) objets (points, segments) sont égaux ;
- deux polygones ont la même aire ;
- un point appartient à une ligne ou une conique ;
- une ligne est tangente ou sécante à une conique ;
- trois points sont alignés ;
- trois lignes sont concourantes (ou parallèles) ;
- quatre points sont cocycliques (ou alignés).

Certaines de ces vérifications peuvent également être effectuées symboliquement, c'est-à-dire que l'énoncé peut être vérifié rigoureusement pour le cas général (avec des coordonnées arbitraires) et non seulement pour la construction géométrique telle qu'elle est représentée.
Avec l'outil *Relation*, l'utilisateur pointe sur deux objets pour afficher le message. Sinon, deux, trois ou quatre objets peuvent être sélectionnés par le rectangle de sélection pour afficher le message. Pour empêcher l'utilisateur de sélectionner des objets parasites, il est également possible d'empêcher leur sélection, en désactivant « *Sélectionnable* » dans l'onglet « *Avancé* » après clic droit sur l'objet, dans ses « *Propriétés …* ».
Avec la commande `Relation`, l'utilisateur valide une des syntaxes suivantes dans le champ de saisie :

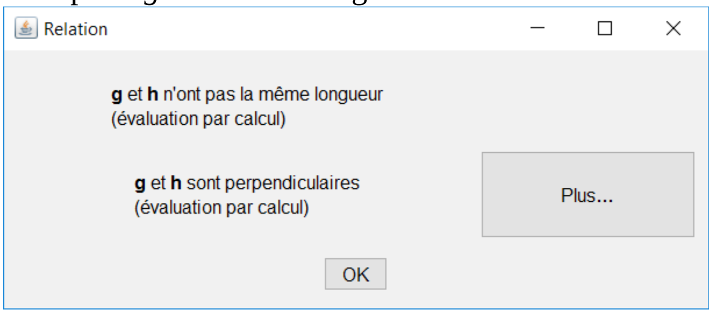
- `Relation[ <Objet>, <Objet> ]`
- `Relation[ { <Objet>, <Objet> } ]`
- `Relation[ { <Objet>, <Objet>, <Objet> } ]`
- `Relation[ { <Objet>, <Objet>, <Objet>, <Objet> } ]`

Quand le message est affiché, avec une ou plusieurs affirmations (évaluées numériquement) reliant les objets, peut être affiché un bouton « *Plus...* » si l'affirmation est éligible à une évaluation symbolique. Après un clic sur ce bouton, l'affirmation numérique sera rapidement mise à jour par une évaluation symbolique.

Exemple (Théorème de l'angle inscrit dans un demi-cercle)

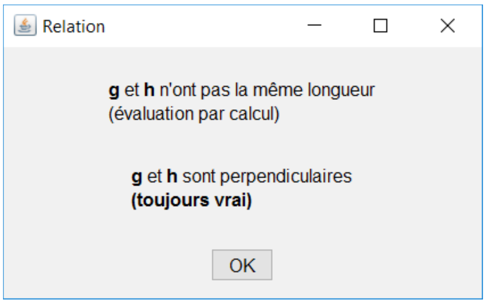


1. Avec l'outil *Segment*, créer un segment [AB].
2. Avec l'outil *Demi-cercle*, créer l'arc *c*.
3. Avec l'outil *Point*, créer un point *C* sur *c*.
4. Avec l'outil *Segment*, créer les segments [AC] et [BC] qui vont être nommés respectivement *g* et *h*.
5. Avec l'outil *Relation* et cliquant sur *g* et *h* avec la souris, ou valider `Relation[g,h]` dans « *Saisie* » pour comparer *g* et *h*. Le message suivant doit s'afficher :



(Avertissement pour les utilisateurs de Windows 10, il arrive actuellement, bien souvent, que le message passe en arrière-plan de la fenêtre GeoGebra, la réduire, la déplacer … pour le lire.)

6. Cliquer sur « *Plus…* », le message doit être alors modifié ainsi :



Remarquer que le message Relation (étape 5) fait apparaître des relations entre *g* et *h* à partir des coordonnées et équations utilisées pour la construction. Alors qu'en cliquant sur « *Plus…* » (étape 6) il est vérifié que *g* et *h* sont perpendiculaires quels que soient les points *A* et *B* choisis à l'étape 1.

La relation entre certains objets ne peut être vraie aussi que sous certaines conditions, ce qui n'est donc pas «toujours vrai». Dans de tels cas, si possible, certaines conditions suffisantes sont affichées. Sinon GeoGebra fait remarquer que la déclaration est vraie «sous certaines conditions». Cela doit être interprété que la déclaration est «généralement vraie», mais dans certains cas secondaires (qui sont «un nombre minimal de cas» comparativement aux cas généraux) l'énoncé peut échouer.
Le résultat symbolique de *Relation* peut être négatif alors que l'évaluation numérique était positive. Par exemple, en définissant deux points *P=(0,0)* et *Q=(0,0), Relation* assure leur égalité par évaluation numérique, mais l'évaluation symbolique va tempérer « *P* et *Q* sont égaux (mais ce n'est pas vrai en règle générale) ».
Un aperçu complet des différents résultats de *Relation* se trouve en Annexe 6.1.3 en page 14.

3.2.2 La commande `EquationLieu`.

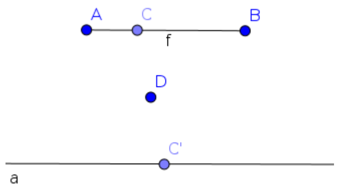
Cette commande calcule l’équation d’un lieu et le représente en tant que courbe implicite.
Consulter mon GgbBook « Travaux EquationLieu ».

Il y a deux manières d’utilisation :

- **Lieu explicite** : Étant donné un point $\mathscr{I}$ sur un chemin $\mathscr{P}$, quelques étapes de construction, et un point $\mathscr{O}$ en découlant. L’objectif est de déterminer l’équation $\mathscr{E}$ de $\mathscr{O}$ quand $\mathscr{I}$ se déplace sur $\mathscr{P}$, et de représenter $\mathscr{E}$. $\mathscr{I}$ est le point mobile, $\mathscr{O}$ est le point du lieu. $\mathscr{E}$ est l’équation du lieu, et la représentation graphique est le lieu.

  La syntaxe de la commande est
  `EquationLieu[ <Point Lieu>, <Point mobile> ]`.

Exemple :

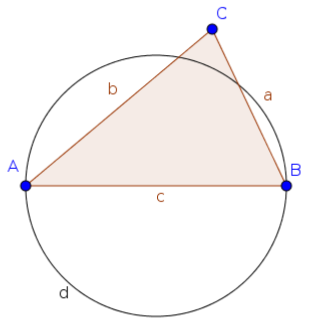


1. Avec l’outil *Segment*, construire un segment [AB], nommé automatiquement *f ;*
2. Avec l’outil *Point*, placer un point *C* sur *f ;*
3. Avec l’outil *Point*, créer un point *D ;*
4. Avec l’outil *Symétrie centrale*, construire le symétrique *C’* de *C* par rapport au point *D ;*
5. Dans « *Saisie* », valider `EquationLieu[C',C]`. Alors une courbe implicite *a* est calculée et tracée. Mais, nous obtenons une droite alors que le symétrique de f est un segment. En effet, pour des raisons de géométrie algébrique, GeoGebra a besoin d’assimiler le segment *f* à sa droite support, dont la symétrique sera aussi une droite ;
6. Avec l’outil *Déplacer*, glisser chaque objet déplaçable. Visualiser que le symétrique d’un segment par rapport à un point est toujours un segment lui étant parallèle.

- **Lieu implicite :** Étant donné un point $\mathscr{I}$, soit libre, soit sur un chemin $\mathscr{P}$, quelques étapes de construction. L'utilisateur applique une condition booléenne $\mathscr{C}$ à des objets de la construction. L’objectif est de déterminer l’équation $\mathscr{E}$ telle que pour tous les points $\mathscr{I}'$ de it, si $\mathscr{I} = \mathscr{I}'$, alors $\mathscr{C}$ est vérifiée. Là encore, $\mathscr{E}$ est l’équation du lieu, et la représentation graphique est le lieu.

La syntaxe de la commande est
`EquationLieu[ <Booléen>, <Point mobile> ]`.

Exemple :



1. Avec l’outil *Polygone*, construire le triangle *ABC*. Ses côtés sont nommés *a*, *b* et *c*.
2. Dans « *Saisie* », valider `EquationLieu[a²+b²==c²,C]`. Alors la courbe implicite *d* est calculée et tracée avec l’apparence d’un cercle.
   Noter que deux signes d’égalité doivent être utilisés pour écrire la condition ; une autre possibilité serait d’utiliser le symbole $\stackrel{?}{=}$ (après avoir cliqué sur $\boxed{\alpha}$ à droite dans « *Saisie* », ou encore, en insérant ce symbole par Copier/Coller à partir d’une autre application).
3. Avec l’outil *Déplacer*, glisser chaque objet déplaçable. Visualiser que si *C* est sur le cercle de diamètre [*AB*] alors, du fait du théorème de l’angle inscrit dans un demi-cercle et du théorème de Pythagore, la relation $a^2 + b^2 = c^2$ en découle.

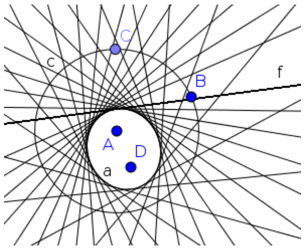
Une expression booléenne peut être :

- Une équation sur les noms de segments, par ex. `a²+b²==c²`.
- Une égalité entre deux objets géométriques, par ex. `A==B`.
  - Une alternative, `SontÉgaux[A,B]` pour l’expression booléenne complète.
- Un test pour savoir si deux objets géométriques sont isométriques, par ex. `SontIsométriques[c,d]`.
- Un test pour savoir si un point est sur un chemin, par exemple, sur une ligne ou sur un cercle, par ex. `A∈c`.
- Un test pour savoir si deux lignes ou segments sont parallèles, par ex. `p∥q`.
  - Une alternative, `SontParallèles[p,q]`.
- Un test pour savoir si deux lignes ou segments sont perpendicular, par ex. `p⊥q`.
  - Une alternative, `SontPerpendiculaires[p,q]`.
- `SontAlignés[A,B,C]` teste si les points *A*, *B* et *C* sont alignés.
- `SontConcourantes[d,e,f]` teste si les lignes *d*, *e* et *f* sont concourantes.
- `SontCocycliques[A,B,C,D]` teste si les points *A*, *B*, *C* et *D* sont cocycliques.

3.2.3 La commande `Enveloppe`.

Cette commande calcule l’équation d’une courbe tangente à un ensemble d’objets lorsqu’un de leurs parents se déplace sur un chemin.
Plus précisément, étant donné un point $\mathscr{I}$ sur un chemin $\mathscr{P}$, quelques étapes de construction, et un chemin $\mathscr{O}$ en découlant, soit une ligne, soit un cercle. L’objectif est de déterminer l’équation $\mathscr{E}$ de la courbe $\mathscr{C}$ tangente à $\mathscr{O}$, lorsque $\mathscr{I}$ se déplace sur $\mathscr{P}$, et de représenter ensuite $\mathscr{E}$. $\mathscr{I}$ est le point mobile. $\mathscr{E}$ est l’équation de l’enveloppe, et la représentation graphique est l’enveloppe.

Exemple :



1. Avec l’outil *Cercle(centre-point)*, construire le cercle *c* de centre *A* passant par *B ;*
2. Avec l’outil *Point*, placer un point *C* sur *c ;*
3. Avec l’outil *Point*, créer un point *D* quelconque à l’intérieur de *c ;*
4. Avec l’outil *Médiatrice*, construire la médiatrice f du segment [CD] en cliquant sur ses extrémités ;
5. Dans « *Saisie* », *v*alider `Enveloppe[f,C]`. Alors une courbe implicite *a* est calculée et tracée avec l’apparence d’une ellipse.

3.3 Notes techniques.

Les notes suivantes listent d’importantes restrictions pour chacune des fonctionnalités de raisonnement automatisé de GeoGebra dans leurs utilisations de calculs symboliques :

- Tous les outils GeoGebra et toutes les étapes de construction n’y sont pas accessibles.
- Les outils accessibles ne peuvent agir que sur un ensemble limité d’objets géométriques, i.e. utilisant des points, lignes, cercles, ou coniques.
- Les demi-droites et les segments sont assimilés à leur droite support. Les arcs de cercle sont assimilés à leur cercle support.
- Les calculs trop compliqués de lieux ou enveloppes retournent « non défini » dans *Algèbre*.
- Les investigations de relations requérant des calculs trop compliqués vont afficher le message « (il est possible que ce soit vrai en règle générale) ». Il faut l’interpréter comme incapicité de GeoGebra à déterminer si oui ou non, la relation est valide en règle générale, mais les résultats numériques permettent de supposer qu«il en est ainsi. Ceci étant, cela n’empêche pas, aussi, que la relation puisse être fausse en règle générale dans ce cas.
- S’il n’y a ni lieu ni enveloppe, alors la courbe implicite est l’ensemble vide « 0 = -1 ».
  Exemple: Étant donné un point quelconque *P*,
  `EquationLieu[false,P]`
  retourne l’ensemble vide.
- Si le lieu, l’enveloppe sont le plan tout entier, alors la courbe implicite a pour équation « 0 = 0 » .
  Exemple: Étant donné un point quelconque *P*,
  `EquationLieu[true,P]`
  retourne le plan tout entier.
- Parfois, des branches de la courbe apparaîtront alors qu’elles ne font partie du lieu ou enveloppe original.
- La représentation graphique de la courbe implicite peut être inexacte dans certains cas.

4 Utilisations en classe : conjecture, preuve et généralisation.

Techniquement, l'outil symbolique le plus facile est l'outil $a \stackrel{?}{=} b$ *Relation* dans la liste ci-dessus. D'autre part, certains scénarios pédagogiques peuvent nécessiter des outils différents pour considérer plus d'un outil, mais dans un ordre différent de celui énuméré ci-dessus.

4.1 Théorème de l'angle inscrit dans un demi-cercle.

Dans de nombreuses classes de mathématiques traditionnelles, le théorème de l'angle inscrit dans un demi-cercle est énoncé sous une forme explicite : si *C* appartient à un demi-cercle, les segments *g* et *h* sont perpendiculaires. En fait, ce théorème peut être formulé en utilisant une question ouverte : Soit un triangle quelconque *ABC*, quel est le lieu géométrique de *C* tel que l'angle en *C* soit un angle droit ?
Dans cette approche, il peut être plus judicieux d'utiliser, en premier, la technique plus difficile de la commande `EquationLieu[g⊥h,C]`, que de terminer la construction et d'utiliser directement l'outil ou la commande `Relation` . Car en plus, la réponse de la commande `EquationLieu` peut suggérer une conjecture pour les élèves, à savoir que la courbe est en effet un cercle. Algèbre affiche l'équation du lieu, mais il peut cependant être difficile pour les jeunes apprenants de l'identifier.
Finalement, le théorème de l'angle inscrit dans un demi-cercle peut être généralisé vers le théorème de l'arc capable. Dans ce cas, la condition n'est plus `g⊥h`, mais que l'angle entre eux est égal à un angle donné. GeoGebra traite actuellement cette recherche à l'aide de la syntaxe :

`EquationLieu[SontIsométriques[α,β],C]`

si α est un angle construit fixé et $\beta=\widehat{ABC}$.

Pour résumer, dans cette approche

1. un lieu implicite est calculé par GeoGebra ;
2. une conjecture sur la nature de la courbe obtenue est émise par les élèves ;
3. la conjecture est testée à l'aide de l'outil ou de la commande *Relation* dans GeoGebra ;
4. la preuve peut être éventuellement établie avec papier et crayon par l'élève ;
5. le théorème peut être généralisé en construisant d'autres lieux implicites à l'aide de GeoGebra— ainsi que par d'autres expériences réalisées par l'élève .

4.2 Autres exemples.

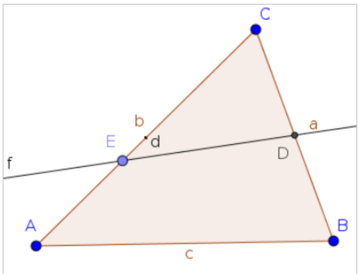
L'inégalité triangulaire peut se traduire par une égalité qui peut être transformée en une investigation des triangles dégénérés. Comme généralisation, la définition synthétique des sections coniques peut être mentionnée.

Une autre application est de décliner l'équation d'un lieu dans un triangle *ABC* dans la condition $a \stackrel{?}{=} b$, ici, il faut chercher *C* (étape 1). Il apparaît clairement que *C* doit appartenir à la médiatrice du segment [*AB*] (étape 2). Alors en plaçant explicitement *C* sur la médiatrice, GeoGebra confirme que *AC* = *BC* lors du démarrage de la machine symbolique de l'outil $a \stackrel{?}{=} b$ *Relation* (étape 3). Après avoir prouvé l'assertion par des moyens traditionnels (étape 4), ne généralisation peut être obtenue en validant par ex. `EquationLieu[a==2b,C]` : Cela peut être aussi une expérience intéressante pour les apprenants avancés (étape 5).

4.3 Un exemple détaillé : Le théorème de la droite des milieux.

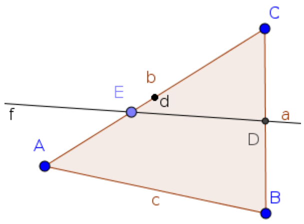
Voici des instructions pas à pas d'une démarche possible pour étudier le théorème de la droite des milieux en utilisant les fonctionnalités de raisonnement automatisé de GeoGebra.

Étape 1 :



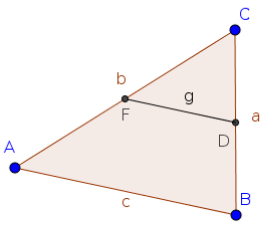
1. Avec l'outil *Polygone*, créer le triangle *ABC*. Ses côtés sont nommés *a*, *b* et *c* ;
2. Avec l'outil *Milieu ou centre*, créer le milieu *D* de *a* ;
3. Avec l'outil *Point*, créer un point *E* sur *b* ;
4. Avec l'outil *Droite*, créer la droite *f* définie par les points *D* et *E* ;
5. Dans « *Saisie* »,valider `EquationLieu[c∥f,E]`pour demander à GeoGebra les exigences sur le point *E* de sorte que *f* soit parallèle à *c*. Alors une courbe implicite *d* est calculée et tracée avec l'apparence d'un unique point.
   Note : Il peut être utile de changer l'« *Épaisseur du trait* » de la courbe implicite *d*, ainsi que d'augmenter le numéro du « *Calque* » pour s'assurer que d'autres objets ne la masquent pas. Ces deux paramètres peuvent être modifiés dans la fenêtre « *Propriétés...* » de l'objet.

Étape 2 :



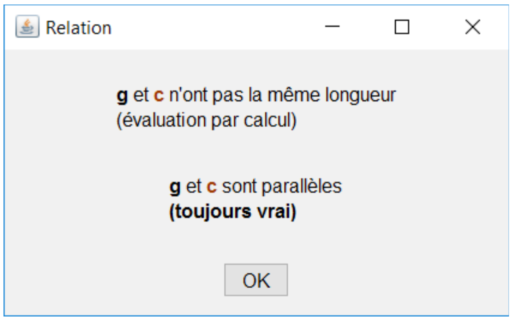
6. Avec l'outil *Déplacer*, glisser les objets libres et conjecturer que *E* doit être le milieu de *b*.
7. Pour confirmer cette conjecture, en utilisant l'outil *Milieu ou centre*, créer le milieu *F* du segment *b* (éventuellement, déplacer les étiquettes de *d* et *F* pour éviter les chevauchements).
   Avec l'outil *Déplacer*, glisser à nouveau les objets libres.
8. Cacher les objets *E*, *f* et *d* (par ex, en cliquant sur leur pastille de visibilité dans « *Algèbre* »)

Étape 3 :



9. Avec l'outil *Segment*, créer le segment *g* d'extrémités *D* et *F ;*
10. Avec l'outil *Relation* comparer *c* et *g*. Ils sont considérés comme étant parallèles *;*

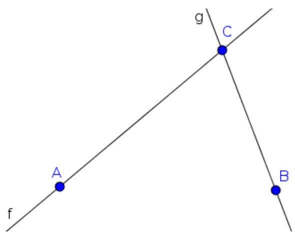
11. Cliquer sur le bouton « *Plus…* » de la fenêtre de message, afin de vérifier par calculs symboliques qu'ils sont effectivement parallèles



Les élèves peuvent poursuivre l'étape 4 s'ils ont besoin d'une façon élégante de prouver cette affirmation, ou s'arrêter ici s'il n'y a pas suffisamment de temps pour poursuivre le travail en classe.
En outre, à l'étape 5, d'autres questions peuvent être soulevées.On constate que *c* et *g* n'ont pas la même longueur, mais celle de *g* peut-elle être calculée en utilisant celle de *c* ? Peut-être que *c = 1,5 g* ou peut-être plus ?
La commande GeoGebra `Relation[c,1.5g]` nous répond que *c* et *1,5g* sont différents, mais peut-être y a t'il une autre constante que 1,5 conduisant à une réponse positive ?Même s'il n'y a pas suffisamment de temps pour poursuivre le travail en classe, certains élèves trouvent ces questions intéressantes et ils peuvent continuer à y penser seuls ou en groupes - mais en quelque sorte indépendamment, en utilisant l'ordinateur comme un système expert.

5 Limitations : Étude du cas Théorème de l'angle inscrit dans un demi-cercle.

L'utilisation intuitive des fonctionnalités de Raisonnement automatisé de GeoGebra peut entraîner des réponses inattendues dans certains cas. Cette sous-section explique quelques erreurs courantes pendant leur utilisation.
Le théorème de l'angle inscrit dans un demi-cercle servira de support à cette analyse.



1. Avec l'outil *Point*, créer les points *A*, *B* et *C* ;
2. Avec l'outil *Droite*, créer les droites *f* (resp. *g*) définies par les points *A* et *C*, (resp. *B* et *C*) ;
3. Dans « *Saisie* », valider `Relation[f,g]` : la réponse est « ***f*** et ***g*** sont sécant(e)s » ;
4. Dans « *Saisie* », valider `EquationLieu[f⊥g,C]`, pour demander à GeoGebra les prérequis afin que *f* soit perpendiculaire à *g*. Alors une courbe implicite *a* est calculée et tracée avec l'apparence d'un cercle ;
5. Avec l'outil *Déplacer*, essayer de positionner *C* le plus près possible de la courbe *a*.

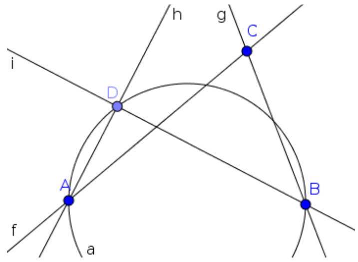
   Dans « *Saisie* », valider, à nouveau, la commande `Relation[f,g]` : la réponse est toujours « ***f*** et ***g*** sont sécant(e)s » alors que l'on pouvait espérer obtenir « ***f*** et ***g*** sont perpendiculaires ».

   Pourquoi cela ? Parce que le point *C* peut ne pas être exactement sur le cercle.

Nous devons préciser qu'il est effectivement sur le cercle.

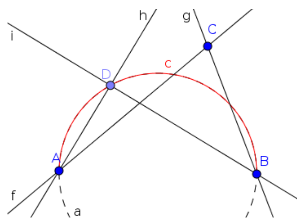
(a) Avec l'outil *Lier/Libérer Point*, essayer d'attacher le point *C* sur la courbe implicite. Cela ne va pas être permis dans GeoGebra, parce que par définition *a* dépend de *C*, et la dépendance circulaire n'aurait pas de sens

(b) Au lieu de cela, créer un nouveau point *D*, avec l'outil *Point* en cliquant sur *a*, cela est permis. Avec l'outil *Droite*, créer les droites *h* (resp. *i*) définies par les points *A* et *D*, (resp. *B* et *D*) ;



(c) Dans « *Saisie* », valider `Relation[h,i]` : la réponse est « ***h*** et ***i*** sont perpendiculaires (évaluation par calcul) ». Cliquer « *Plus*... », la réponse symbolique ne donne que : « (il est possible que ce soit vrai en règle générale) ». Pourquoi ne pas affirmer que c'est toujours vrai? Parce que GeoGebra interprète la courbe implicite *a* sous-jacente comme le résultat d'un organisation particulière de la construction . En d'autres termes, une courbe implicite est un objet numérique, elle n'a pas de représentation symbolique. C'est ainsi, il n'est pas possible d'effectuer des calculs symboliques basés sur une courbe implicite. Ici GeoGebra était juste optimiste sur la véracité de la conjecture, mais le logiciel était réellement incapable de la prouver.

(d) La manière appropriée de finaliser les étapes de cette approche est de créer le cercle de diamètre [AB] avec un outil « cercle », par exemple avec l'outil *Demi-cercle*, puis après avoir libéré, avec l'outil *Lier/Libérer Point*, *D* de *a*, et caché *a*, puis avec l'outil *Lier/Libérer Point*, lié *D* au demi-cercle



(Éventuellement, la courbe implicite *a* peut rester visible en l'affichant avec un style différent. Dans cet exemple, un autre style a également été utilisé pour le demi-cercle.) Enfin, `Relation[h,i]` donnera maintenant des résultats positifs à la fois numériquement et symboliquement.

6 Annexe.

6.1 Fonctionnalités de « bas niveau ».

Parmi les fonctionnalités de Raisonnement automatisé de GeoGebra, il en est certaines de « bas niveau » implémentées pour en apprendre plus et d'une manière plus précise sur les propriétés géométriques.

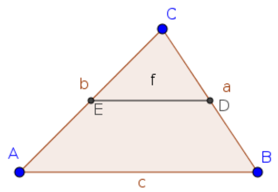
6.1.1 La commande `Prouver`.

La commande `Prouver` décide si une affirmation géométrique est vraie en règle générale.
Elle dispose de trois réponses possibles :

- *true* signifiant que l'affirmation est toujours vraie, ou vraie sous certaines conditions de non-dégénérescence ou conditions essentielles ou sur certains composants ;

- *false* signifiant que l’affirmation est fausse en règle générale. GeoGebra utilise la géométrie algébrique dans de nombreux cas pour trancher sur de telles questions . En géométrie algébrique « vrai en règle générale » (vrai dans la « plupart » des cas) et « faux en règle générale » (faux dans la « plupart » des cas) ne sont pas des propriétés contraires, en effet, une affirmation peut être à la fois non « vraie en règle générale » et non « fausse en règle générale ». GeoGebra interprète ce cas particulier comme faux (puisqu'il n'est pas vrai en règle générale).
- *non défini* signifiant que GeoGebra ne peut trancher pour différentes raisons :
    - L’affirmation n’a pu être traduite en un modèle étudiable plus avant. Cela signifie généralement que l’algébrisation de l'énoncé a échoué du fait
        - d’une impossibilité au niveau théorique (par ex. utilisation d’une fonction transcendante dans une étape de la construction, par exemple, sinus de x),
        - d’une implémentation manquante dans GeoGebra.
    - La traduction de l’affirmation en géométrie algébrique est trop difficile à résoudre, parce qu’il y a trop de variables ou que les équations sont trop difficile à traiter par l’algorithme de résolution. Il en résulte un délai dépassé ou un dépassement de capacité de la mémoire.
    - L’algorithme de résolution peut étudier la situation, mais le résultat est ambigu : soit l'affirmation est fausse, soit elle est vraie sous certaines conditions, mais l'algorithme n'a pas été en mesure de décider ce qu’il en est dans le cas présent.
    - Il y a eu une erreur interne dans GeoGebra pendant les calculs.

Exemple :



1. Avec l’outil *Polygone*, créer un triangle *ABC.* Ses côtés sont nommés *a*, *b* et *c* ;
2. Avec l’outil *Milieu ou centre*, créer les milieux *D* (resp. *E*) des côtés *a* (resp.*b*) ;
3. Avec l’outil *Segment*, créer le segment *f* d’extrémités *D* et E *;*
4. Dans « *Saisie* », valider `Prouver[f∥c]`. Alors dans « *Algèbre* », constater la création d’un booléen *d* de valeur *true* ;
   Noter que le signe de parallélisme peut être obtenu de différentes manières
    - à partir de la liste de symboles après avoir cliqué sur α à droite dans « *Saisie* »,
    - ou encore, en insérant ce symbole par Copier/Coller à partir d’une autre application.
    - Et aussi, l’expression `f∥c` peut être remplacée par `SontParallèles[f,c]`
5. Dans « *Saisie* », valider `Prouver[c==3f]`. Maintenant la réponse est *non défini,* parce que GeoGebra ne peut trancher si cette expression est fausse ou si elle est vraie seulement sous certaines conditions.
   Dans de tels cas la commande `PreuveDétaillée` (voir ci-dessous) peut apporter de l’aide.
   Noter que deux signes d’égalité doivent être utilisés pour écrire la condition ; deux autres possibilités d’écriture sont
    - `Prouver[c ≟ 3f]`, ou
    - `Prouver[SontÉgaux[c,3f]`.

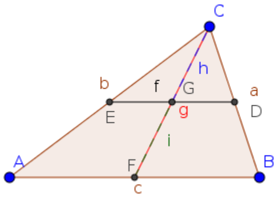
6.1.2 La commande `PreuveDétaillée`.

La commande `PreuveDétaillée` a un comportement similaire à celui de la commande `Prouver`, mais elle peut utiliser des algorithmes différents dans le processus de décision, et peut fournir plus d’informations dans les résultats. Elle dispose de trois réponses possibles :

- {*true*} signifiant que l’affirmation est toujours vraie ;
- {*true*, {**...**}} signifiant que l’affirmation est vraie sous certaines conditions de non-dégénérescence ou conditions essentielles ou sur certains composants: ces conditions sont listées dans les accolades intérieures. (Si la liste reste {**...**}, cela signifie qu'aucune traduction synthétique n'a pu être trouvée.) Si la conjonction des conditions rejetées est vraie, alors l’affirmation est vraie .
- {*false*} signifiant que l’affirmation est fausse en règle générale. Consulter les commentaires pour la commande `Prouver` pour plus de détails à ce sujet.

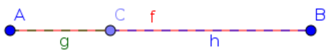
Exemple (suite).

1. Dans « *Saisie* », valider `PreuveDétaillée[c==3f]`. La réponse est {*false*}.
2. Dans « *Saisie* », valider `PreuveDétaillée[c==2f]`. La réponse est {*true*}.



3. Avec l’outil *Milieu ou centre*, créer le milieu *F* du côté *c ;*
4. Avec l’outil *Segment*, créer le segment *g* d’extrémités *C* et *F ;*
5. Avec l’outil *Intersection*, créer le point d’intersection *G* des segments *f* et g ;
6. Avec l’outil *Segment*, créer les segments *h* (resp. *i*) d’extrémités *C* et *G* (resp. *F* et *G*) ;
7. Dans « *Saisie* », valider `PreuveDétaillée[h==i]`, dans ce cas, la réponse est : {*true*, {"*SontAlignés*[*A*,*B*,*C*]"}} précisant que si *A*, *B* et *C* ne sont pas alignés, alors *h* = *i*.

Autre exemple :



1. Avec l’outil *Segment*, construire un segment [AB], nommé automatiquement *f ;*
2. Avec l’outil *Point*, créer le point *C* sur le segment *f* ;
3. Avec l’outil *Segment*, créer les segments g (resp. *h*) d’extrémités *A* et *C* (resp. *B* et *C*) ;
4. Dans « *Saisie* », valider `PreuveDétaillée[f==g+h]`. La réponse est {true, {"g + f = h", "h + f = g"}} précisant que si g + f ≠ h et h + f ≠ g alors f=g+h.

6.1.3 Une comparaison de `Prouver`, `PreuveDétaillée` et `Relation`.

Le tableau ci-dessous explique la signification des réponses fournies par ces commandes :

Réponses GeoGebra			Conclusion
`Prouver`	`PreuveDétaillée`	`Relation` (après calculs symboliques)	
true	{true}	(toujours vrai)	L’affirmation est vraie.
	{true,{conditions}}	Il est vrai en règle générale que … sous la condition.	L’affirmation est vraie si les conditions spécifiées sont réalisées. Ces conditions sont suffisantes mais peuvent être non

			nécessaires. Il peut exister d’autres conditions suffisantes rendant vraie l’affirmation.
	{true,{...}	Vrai sous certaines conditions	L’affirmation est vraie si certaines équations sont résolues. Ces équations n’ont pas une signification géométrique claire.
	{}	(vrai en règle générale)	L’affirmation est vraie si certaines conditions sont réalisées. GeoGebra ne peut établir ces conditions du fait de difficultés de calcul.
false	{false}	(mais ce n'est pas vrai en règle générale)	L’affirmation est fausse.
	{}	(il est possible que ce soit vrai en règle générale)	GeoGebra n’arrive pas à trancher si l’affirmation est vraie ou fausse. L’évaluation numérique assure la véracité, mais l’évaluation symbolique est en échec du fait de difficultés de calcul.
non défini	{true}	(toujours vrai)	L’affirmation est vraie.
	{true,{conditions}}	Vrai sous la conjonction de conditions spécifiées	L’affirmation est vraie si les conditions spécifiées sont réalisées. Ces conditions sont suffisantes mais peuvent être non nécessaires. Il peut exister d’autres conditions suffisantes rendant vraie l’affirmation.
	{true,{...}}	Vrai sous certaines conditions	L’affirmation est vraie si certaines équations sont résolues. Ces équations n’ont pas une signification géométrique claire pour GeoGebra.
	{false}	Faux en règle générale	L’affirmation est fausse.
	{}	(il est possible que ce soit vrai en règle générale)	GeoGebra n’arrive pas à trancher si l’affirmation est vraie ou fausse. L’évaluation numérique assure la véracité, mais l’évaluation symbolique est en échec du fait de difficultés de calcul ou n’est pas encore implémentée.

6.2 Débogage.

En démarrant GeoGebra en ligne de commande, il y a plus de possibilités d'étudier les résultats. Sur une installation Linux typique, l’utilisateur doit valider les commandes suivantes :

```
geogebra -logfile=/dev/stdout -logshowcaller=false \
    -logshowtime=false -logshowlevel=false
```

Une sortie typique ressemble à ce qui suit :

```
Using AUTO
Using BOTANAS_PROVER
A = (3.42, 1.86) /* free point */
// Free point A(v1,v2)
B = (10.48, 3.1) /* free point */
// Free point B(v3,v4)
f = Segment[A, B] /* Segment [A, B] */
```

```
C = Point[f] /* Point on f */
// Constrained point C(v5,v6)
Hypotheses:
1. -v5*v4+v6*v3+v5*v2-v3*v2-v6*v1+v4*v1
g = Segment[A, C] /* Segment [A, C] */
h = Segment[C, B] /* Segment [C, B] */
Processing numerical object
Hypotheses have been processed.
giac evalRaw input: evalfa(expand(ggbtmpvarf))
giac evalRaw output: ggbtmpvarf
input = expand(ggbtmpvarf)
result = ggbtmpvarf
eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh^2=v11^2,ggbtmpvarg^2=v12^2,ggbtmpvarf^2=v13^2],
[ggbtmpvarh,ggbtmpvarg,ggbtmpvarf])
giac evalRaw input: evalfa(eliminate([ggbtmpvarf-((ggbtmpvarg)+
(ggbtmpvarh))=0,ggbtmpvarh^2=v11^2,ggbtmpvarg^2=v12^2,ggbtmpvarf^2=v13^2],[ggbtmpvarh,ggbtmpvarg,ggbtmpvarf]))
Running a probabilistic check for the reconstructed Groebner basis. If successfull, error probability is less than 1e-07
and is estimated to be less than 10^-18. Use proba_epsilon:=0 to certify (this takes more time).
// Groebner basis computation time 0.000448 Memory -1e-06M
giac evalRaw output: {v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}
input = eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh^2=v11^2,ggbtmpvarg^2=v12^2,ggbtmpvarf^2=v13^2],
[ggbtmpvarh,ggbtmpvarg,ggbtmpvarf])
result = {v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}
giac evalRaw input: evalfa(eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh=v11,ggbtmpvarg=v12,ggbtmpvarf=
v13],[ggbtmpvarh,ggbtmpvarg,ggbtmpvarf]))
Running a probabilistic check for the reconstructed Groebner basis. If successfull, error probability is less than 1e-07
and is estimated to be less than 10^-18. Use proba_epsilon:=0 to certify (this takes more time).
// Groebner basis computation time 0.000592 Memory -1e-06M
giac evalRaw output: {v11+v12-v13}
input = eliminate([ggbtmpvarf-((ggbtmpvarg)+(ggbtmpvarh))=0,ggbtmpvarh=v11,ggbtmpvarg=v12,ggbtmpvarf=v13],
[ggbtmpvarh,ggbtmpvarg,ggbtmpvarf])
result = {v11+v12-v13}
giac evalRaw input: evalfa(simplify({v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}/{v11+v12-v13}))
giac evalRaw output: {v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-v13^3}
input = simplify({v11^4-2*v11^2*v12^2+v12^4-2*v11^2*v13^2-2*v12^2*v13^2+v13^4}/{v11+v12-v13})
result = {v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-v13^3}
giac evalRaw input: evalfa(factor(v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-
v13^3))
giac evalRaw output: (v11-v12-v13)*(v11-v12+v13)*(v11+v12+v13)
input = factor(v11^3-v11^2*v12+v11^2*v13-v11*v12^2-2*v11*v12*v13-v11*v13^2+v12^3+v12^2*v13-v12*v13^2-v13^3)
result = (v11-v12-v13)*(v11-v12+v13)*(v11+v12+v13)
Trying to detect polynomial -v13-v12+v11-v13-v12+v11 means h = f + g
Trying to detect polynomial v13-v12+v11v13-v12+v11 means f + h = g
Trying to detect polynomial v13+v12+v11v13+v12+v11 means f + g + h = 0, uninteresting
Thesis equations (non-denied ones):
2. v11^2-v6^2-v5^2+2*v6*v4-v4^2+2*v5*v3-v3^2
3. v12^2-v6^2-v5^2+2*v6*v2-v2^2+2*v5*v1-v1^2
4. v13^2-v4^2-v3^2+2*v4*v2-v2^2+2*v3*v1-v1^2
Thesis reductio ad absurdum (denied statement), product of factors:
(v13^4-2*v13^2*v12^2+v12^4-2*v13^2*v11^2-2*v12^2*v11^2+v11^4)*v14-1
that is,
5. -1+v14*v13^4-2*v14*v13^2*v12^2+v14*v12^4-2*v14*v13^2*v11^2-2*v14*v12^2*v11^2+v14*v11^4
substitutions: {v1=0, v2=0}
Eliminating system in 8 variables (5 dependent)
giac evalRaw input: evalfa([[ff:=\"\"],[aa:=eliminate2([v12^2-v6^2-v5^2,v11^2-v6^2-v5^2+2*v6*v4-v4^2+2*v5*v3-v3^2,-1+v14*
v13^4-2*v14*v13^2*v12^2+v14*v12^4-2*v14*v13^2*v11^2-2*v14*v12^2*v11^2+v14*v11^4,v13^2-v4^2-v3^2,-v5*v4+v6*v3],
revlist([v6,v11,v12,v13,v14]))],[bb:=size(aa)],[for ii from 0 to bb-1 do ff+=(\"[\"+(ii+1)+\"]: [1]:
unicode95uunicode91u1]=1\");cc:=factors(aa[ii]);dd:=size(cc);for jj from 0 to dd-1 by 2 do ff+=(\"
unicode95uunicode91u\"+(jj/2+2)+\"]=\"+cc[jj]); od; ff+=(\" [2]: \"+cc[1]);for kk from 1 to dd-1 by 2 do ff+=(\",\"+
cc[kk]);od;od],[if(ff==\"\") begin ff:=[0] end],ff][5])
Running a probabilistic check for the reconstructed Groebner basis. If successfull, error probability is less than 1e-07
and is estimated to be less than 10^-7. Use proba_epsilon:=0 to certify (this takes more time).
// Groebner basis computation time 0.000249 Memory -1e-06M
giac evalRaw output: "[1]: [1]: unicode95uunicode91u1]=1 unicode95uunicode91u2]=1 [2]: 1,1"
Considering NDG 1...
Found a better NDG score (0.0) than Infinity
Statement is GENERALLY TRUE
Benchmarking: 38 ms
STATEMENT IS TRUE (yes/no: TRUE)
OUTPUT for ProveDetails: null = {true, {"f + h = g", "h = f + g"}}
```

Il n’y a, intentionnellement, pas de méthode plus facile pour montrer aux utilisateurs ce type de sortie. Toutefois, les dernières lignes de l'information de débogage sont disponibles dans GeoGebra dans le menu « *Aide* », en choisissant « *À propos / Licence* », puis en cliquant sur « *Informations système* » - ceci copie les derniers messages de débogage dans le presse-papiers.

6.3 Traduction des commandes GeoGebra.

Les noms des commandes de Raisonnement automatisé de GeoGebra ont peut-être besoin d'être traduits dans d'autres langues. Par exemple, la traduction en allemand de `Prove` est `Prüfe`.
Pour découvrir les noms traduits, les étapes suivantes sont conseillées :

1. Créer un fichier GeoGebra contenant les commandes requises dans « *Algèbre* » ;
2. Changer la langue de GeoGebra via le menu « *Options, Langue* » ;
3. Les noms de commande vont être automatiquement traduits dans « *Algèbre* » ;
4. Survoler à la souris une commande dans « *Algèbre* » et lire son nom traduit.

Anglais.	Français.
Relation	Relation
Locus	Lieu
LocusEquation	EquationLieu
Envelope	Enveloppe
Prove	Prouver
ProveDetails	PreuveDétaillée

Traduction et Adaptation française par Noël LAMBERT, 17/03/25, Version bureau 5.0.346.0-3D
Révisé par Zoltán KOVÁCS, 17/07/13, Version bureau 5.0.374.0-3D