

CaRMetal : UNE GÉOMÉTRIE DYNAMIQUE ENRICHIE

Yves MARTIN

Université de la Réunion (LIM, EREDIM)

Résumé – Ce texte pose un regard didactique sur l'utilisation des différents concepts que met à notre disposition le logiciel CaRMetal. Nous y abordons le sens que prend l'anticipation systématique de tous les outils, l'opérationnalité de l'a-modalité généralisée, la pertinence augmentée de la manipulation directe et son extension aux structures finies, l'affranchissement du déterminisme avec le plongement des figures dans le temps de l'utilisateur, l'engagement direct augmenté qui en résulte ainsi qu'un déterminisme enrichi rendu possible par la rencontre de l'aimantation et de la récursivité. Puis, avec son intégration du JavaScript, nous abordons des usages scolaires spécifiques comme l'investigation algébrique avec la géométrie repérée dynamique. Nous terminerons en revenant sur le temps pour voir que l'animation respecte la temporalité dans laquelle les figures peuvent être plongées.

Mots-clés : géométrie dynamique, anticipation, modalité, a-modalité, déterminisme, continuité, logique, booléen, récursivité, aimantation, réalité augmentée, géométrie repérée, algorithmique, programmation, JavaScript, Laborde, Kortenkamp, Hakenholz, CaRMetal, EIAH.

Abstract. – *The following paper gives a didactic account of the handling of the various concepts offered by the CaRMetal program. It helps us understand the meaning of the systematic anticipation of all the tools, the operability of generalised a-modality, the pertinence coupled with direct manipulation and its extension to finite structures, the breaking away from determinism through figures immersed in the user's time, the increased direct engagement resulting from it together with a form of enriched determinism made possible by the merging of magnetisation and recursivity. Then, with the integration of Javascript we are informed about such specific utilisations in school as algebraic investigation with detected dynamic geometry. Finally, we will reverse to time and see that animation respects the temporality in which figures may be immersed.*

Keywords: *dynamic geometry, anticipation, modality, a-modality, determinism, continuity, logic, boolean, recursivity, magnetisation, increased reality, Cartesian geometry, algorithmics, programming, JavaScript, Laborde, Kortenkamp, Hakenholz, CaRMetal, EIAH.*

Sommaire

I. Introduction – L’héritage de Cabri-géomètre	
1. Manipulation directe – Engagement direct – Micromonde	167
2. Réification	169
3. Genèse de CaRMetal	170
II. Généralités sur l’interface	
1. L’anticipation des constructions	170
2. La question de la modalité	175
3. La logique dans les figures – Géométrie logique	181
4. L’environnement 3D de CaRMetal	185
5. Déterminisme et continuité	191
III. Nouveau paradigme géométrique	194
1. Micromondes et macro-constructions	195
2. La géométrie non arguésienne – L’exemple du plan de Moulton	196
3. Un autre exemple sur la pseudosphère de Beltrami	199
IV. Aimantation et réalité augmentée	
1. Premières utilisations didactiques : du continu au discret	202
2. Aimantation algébrique et déterminisme	210
V. Déterminisme enrichi par la récursivité	214
1. Récursivité et introduction du temps dans les figures	216
2. Récursivité et magnétisme – Première approche	218
3. Engagement direct augmenté – Déterminisme enrichi	221
4. Récursivité croisée en manipulation directe	224
VI. Un autre traitement du temps	225
VII. Les CarScripts	227
1. Introduction aux CarScripts	228
2. Géométrie repérée dynamique	229
3. Insertion de variables JavaScript dans les figures	234
4. Une figure CaRMetal comme entrée d’un script : scripts génériques	236
5. Utilisation en statistique	241
6. Statique/Absolu : un subtil mélange de statut	244
VIII. Aller (un peu) plus loin avec les scripts	
1. Utilisation pour réaliser des figures 3D	251
2. La récursivité dynamique (sur la courbe de Cesaro)	254
3. Animation, script et temporalité	255
IX. Aller (nettement) plus loin avec les scripts	
1. Retrouver les coordonnées 3D d’un point à l’écran	258
2. Premiers scripts objets – Patrons dynamiques de polyèdres	263
X. Conclusions et perspectives	267
Bibliographie	270
Webographie	272

La géométrie dynamique (GD) s'inscrit dans le cadre des Environnements Informatiques pour l'Apprentissage Humain (EIAH), qui, au sens le plus large, sont des artefacts construits dans l'interaction d'une intention didactique et d'un environnement informatique. Au sein des EIAH, la géométrie dynamique est un dispositif particulier, de type micromonde – opposé à celui de tuteur intelligent par exemple –, même si ces micromondes sont maintenant facilement incorporables à des dispositifs d'EIAH plus globaux (précepteurs artificiels, classes virtuelles, comme les structures Wims ou LaboMep). L'étude et l'analyse d'un dispositif d'EIAH peuvent provenir de différents regards : pédagogique, didactique, cognitive, informatique entre autres, selon des critères propres à chaque questionnement. On se propose ici de porter un regard essentiellement didactique sur l'évolution de l'implémentation de la géométrie dynamique et les conséquences d'ingénierie qui peuvent en découler, en prenant comme objet d'analyse l'un des logiciels libres et multiplateformes les plus récents, qui a cette particularité de proposer aussi un langage de script en conformité avec l'évolution des programmes de lycée français depuis l'introduction de l'algorithmique et la pratique de la programmation dans les classes scientifiques.

I. Introduction – L'héritage de Cabri-géomètre

Parmi les logiciels qui ont marqué le début de la géométrie dynamique, nous ferons référence à Cabri-géomètre (Laborde, 1985), pour le degré de finalisation qu'il a rapidement atteint et la richesse des analyses didactiques qui se sont construites autour de lui. L'autre logiciel, de même qualité et développé en même temps, Cinderella (Kortenkamp, 1999), évoque encore dans les mémoires une extraordinaire exploration des possibilités du continu avec un abandon assumé du déterminisme. Mais Cinderella n'était pas un micromonde au sens où nous allons le préciser, même s'il l'est devenu ensuite dans sa version 2, en même temps qu'il revenait au déterminisme. De son côté, Geometer's SketchPad, par son traitement des *sketchs*, n'était pas un micromonde. Quoi qu'il en soit, en raison des développements didactiques qu'il a entraînés, mais aussi par certaines implémentations qui n'ont jamais été reproduites depuis, Cabri reste une référence historique incontournable pour parler de l'évolution de la GD.

1. Manipulation directe – Engagement direct – Micromonde

Dans ses premières acceptations, le concept de manipulation directe (Shneiderman, 1983), défini pour les logiciels en général, correspond bien –

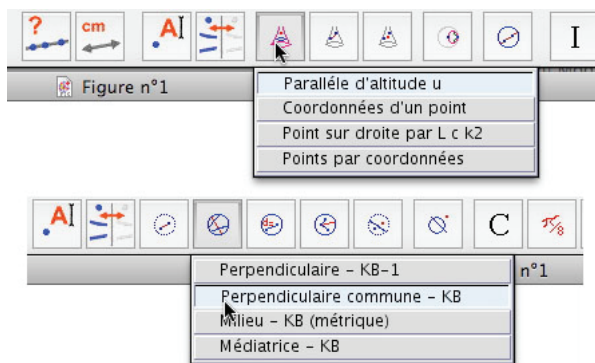
du point de vue de l'artefact à réaliser – à ce qu'on attend encore de la manipulation directe. Les quatre règles mises en évidence par Shneiderman étaient :

1. *Représentation permanente des objets d'intérêt.*
2. *Utilisation d'actions physiques au lieu de commandes textuelles : « voir et montrer au lieu de se rappeler et taper ».*
3. *Le résultat des actions physiques doit être immédiatement visible.*
4. *Ces actions doivent être rapides, incrémentales et réversibles.*

Mais, dès leur projet initial, les logiciels de géométrie dynamique, ayant une intention didactique manifeste – le concept d'EIAH n'existait pas, on parlait alors d'EIAO – ont ajouté, à ces règles générales, d'autres dites d'engagement direct (Nanard, 1990). D'une manière générale, ces règles consistent à implémenter, dans la réaction du logiciel à la manipulation, un comportement qui, de fait, reconnaît l'utilisateur comme un sujet cognitif vis-à-vis des concepts qu'il manipule avec ces outils. D'où la prise en compte du logiciel de géométrie dynamique comme micromonde, pas nécessairement au sens premier (Papert, 1980), mais dans un sens plus riche puisque le logiciel embarque déjà toute la géométrie euclidienne élémentaire.

Au sens de Papert, comme structure qui réifie les objets mathématiques et les opérations sur ces objets par la manipulation directe, mais aussi comme environnement d'apprentissage ouvert, les logiciels de géométrie dynamique sont tous, de fait, des micromondes. Aussi, dans ce cadre spécifique, on s'entend généralement pour donner une définition plus forte de ce concept. On dira d'un logiciel de géométrie dynamique qu'il est un micromonde s'il permet de créer de nouveaux outils (macro-constructions) et de les structurer pour permettre de travailler dans un environnement propre à un contexte géométrique particulier.

Selon les logiciels, la question de la structure est plus ou moins finalisée, ou l'est de façons différentes. Les choix de Cabri, très centrés sur cette définition forte des micromondes, avec les possibilités du logiciel de personnaliser ses macros par des icônes et surtout ses barres de menu, n'ont, semble-t-il, jamais été égalés depuis. On donne ci-dessous, à titre d'exemples, une barre de macros sur la pseudosphère (version Cabri II 2001) et une autre sur le disque de Klein-Beltrami (KB) pour travailler sur la pseudosphère par conjugaison.



2. Réification

D'une manière générale, par réification, on entend la transformation – ou la transposition – d'une abstraction en un objet concret. Par exemple, pour l'apprentissage du calcul au cycle 2, l'utilisation de collections organisées – doigts, constellations – pour un « calcul sur les objets » (Brissiaud, 2003), est une réification de la cardinalité dont l'efficacité a fait le succès de la méthode PicBille. En informatique, la réification consiste à transformer en objet informatique un concept (typiquement, les palettes d'outils de tous les logiciels contemporains sont des réifications). Mais pour parler véritablement de réification, des règles d'interaction instrumentale – au minimum la manipulation directe et l'absence de boîte à dialogue – doivent être respectées :

« De nombreuses applications bénéficieraient de l'usage de la réification.

Ainsi, les styles de Word n'ont pas réellement un statut d'objet : ils ne sont éditables que dans des boîtes de dialogue modales, et ne peuvent pas apparaître de façon permanente (à part dans le menu des styles) ; on peut cependant les stocker dans des fichiers. Dans Photoshop, les filtres sont des instruments, accessibles uniquement par l'intermédiaire de commandes et de boîtes de dialogue modales. Pourtant, les filtres sont centraux dans cette application de retouche photographique. En réifiant les filtres, il serait possible de créer des configurations prêtes à l'emploi, sans avoir besoin de spécifier les paramètres à chaque utilisation » (Beaudoin-Lafon, 1997, p. 5).

On voit que, dès 1997, la modalité est perçue comme un obstacle à la réification. C'est sur un constat de réification dégradée – et largement améliorable – que CaRMetal a vu le jour.

3. Genèse de CaRMetal

CaRMetal est une belle histoire du logiciel libre. Il provient de CaR, (*Compass and Ruler*, de René Grothmann), logiciel de GD avec un moteur mathématique efficace, repéré par Éric Hakenholz pour plusieurs spécificités dont l'extraordinaire anticipation des outils, les aspects conditionnels algébriques, ses macros et quelques fonctionnalités internes d'une grande pertinence. Sa prise en main étant peu pratique – beaucoup d'aspects modaux, icônes peu significatives – Éric Hakenholz a commencé par en améliorer l'interface, d'où le nom CaRMetal faisant référence à l'interface « métal brossé » des applications Mac OS X. Puis, au bout de deux ans, la maintenance de l'interface au sein d'un logiciel en évolution constante devenant de plus en plus lourde, il a fini par développer un logiciel à part, et s'est ensuite autorisé à apporter des améliorations au sein du moteur interne du logiciel, jusqu'à en récrire une grande partie pour la version 3.5 d'avril 2010, celle qui a servi pour cette analyse. C'est désormais un projet déposé sur SourceForge avec ce que cela implique d'ouverture.

Dans ce texte on mentionnera CaR lorsqu'il sera question des fonctionnalités du logiciel initial de René Grothmann et on parlera de CaRMetal quand il s'agira d'un ajout ultérieur. CaRMetal est écrit en Java, donc multiplateforme. Il est disponible pour Windows, Linux et Mac OS X.

II. Généralités sur l'interface

Dans cette partie, nous aborderons les thèmes de l'anticipation des constructions, de la modalité et du déterminisme des logiciels de géométrie dynamique. Ces questions sont présentes dans tous les logiciels. Puis nous traiterons de la gestion de la logique et de la 3D. Dans ces cinq thèmes, des progrès significatifs ont été réalisés.

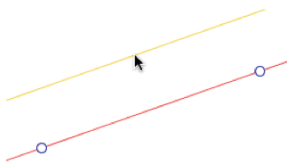
1. L'anticipation des constructions

Tous les logiciels de GD anticipent les *items de création* (segments, droites, cercles, polygones, coniques). Mais généralement, ils n'ont pas fait le choix d'aller plus loin, c'est-à-dire de proposer le même fonctionnement pour les outils de construction.

Dans une société où les outils mis à notre disposition sont en constante évolution, où l'apprentissage est permanent, la logique interne des logiciels est de contenir au sein même de leurs outils, une aide à leur propre apprentis-

sage. L'aide peut être de bas niveau – on pense à la bulle rétroactive sur une icône qui rappelle sa fonctionnalité – ou intégrée à l'interface en accompagnant l'apprentissage dans la manipulation même de l'outil (Photoshop, Illustrator pour les logiciels professionnels, mais encore la suite iLife pour les logiciels grand public).

C'est le choix qu'a fait René Grothmann pour CaR. Ce choix est fondamental pour le didacticien car l'anticipation des constructions provoque une tout autre instrumentation de la géométrie dynamique. Pour chaque outil, cette anticipation accompagne de fait l'utilisateur – en particulier jeune – dans la construction de ses représentations du concept engagé. Ainsi, dans CaRMetal, si on veut tracer une parallèle à une droite donnée, la parallèle est portée par la souris, en translation, dès qu'une droite est sélectionnée, jusqu'à ce que l'utilisateur propose un point.



L'anticipation de l'outil de construction « Parallèle »

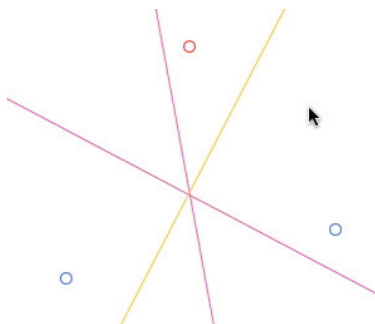
L'accompagnement que produit l'anticipation est cognitif : l'utilisateur, que ce soit un élève découvrant la géométrie élémentaire ou un étudiant explorant des micromondes non euclidiens, s'approprie les schèmes des actions accomplies par l'anticipation, les interprète et construit des représentations personnelles du concept qui sont naturellement bien plus pertinentes et rapidement stables que sans cet accompagnement. Si l'engagement direct des logiciels de première génération s'est construit sur la reconnaissance de l'utilisateur comme sujet cognitif, une anticipation des constructions va plus loin, elle le reconnaît en plus comme sujet *apprenant* et accompagne son apprentissage.

Instrumentalisation de cette anticipation

Cette première analyse correspond à celle de *l'instrumentation* de cette anticipation des constructions. On pense en particulier à une utilisation dans le cycle 3 de l'école primaire et au début du collège. À un autre niveau, par exemple au lycée ou dans l'enseignement supérieur, *l'anticipation peut aussi être instrumentalisée*, au moins par l'enseignant dans un premier temps – sous forme de TP – avant de l'être spontanément par ses élèves ou étudiants.

Cela conduit notamment à une approche différente de l'investigation puisqu'il n'est plus nécessaire de finaliser une construction pour tester une conjecture. Au-delà d'un réel gain de temps dans les phases d'exploration, c'est alors surtout une démarche cognitive *réflexive* sur la rétroaction observée du logiciel.

Un exemple élémentaire de cet enjeu est déjà présent dans la simple construction des médiatrices d'un triangle pour obtenir son cercle circonscrit : avec ce logiciel on rencontre très régulièrement une situation comme la suivante qui invite à la réflexion et à l'investigation.



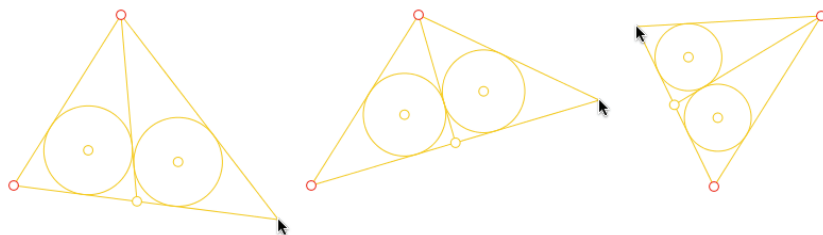
On s'apprête à construire la troisième médiatrice d'un triangle quand on passe à la souris sur une situation comme celle de la copie d'écran ci-dessus. Alors qu'on voulait aller d'un sommet du triangle à un autre pour finaliser la figure, on passe par un point qui, en l'ajustant un peu, rend aussi les médiatrices concourantes, puis un autre à côté et plusieurs autres ainsi avant d'arriver au point que l'on devait cliquer. On a déjà construit l'image mentale d'un arc de cercle, soit par jeu, soit par curiosité intellectuelle, avant même d'avoir terminé la dernière médiatrice. Le cercle ensuite construit, cette observation fortuite peut être réinterrogée par l'enseignant pour une consolidation, par l'échange verbal, des schèmes qui se sont construits à ce moment.

On comprend bien toute l'instrumentalisation que l'on peut faire d'un tel outil, et la dimension autoréflexive qu'elle permet pour peu qu'elle soit initiée et régulièrement réactivée en classe (même sans logiciel, dans l'environnement statique du tableau noir, comme démarche d'anticipation).

Anticipation de tous les outils

Dans ce logiciel, l'anticipation de la construction est le fonctionnement naturel de tous les outils. La règle d'utilisation générique qui en découle est

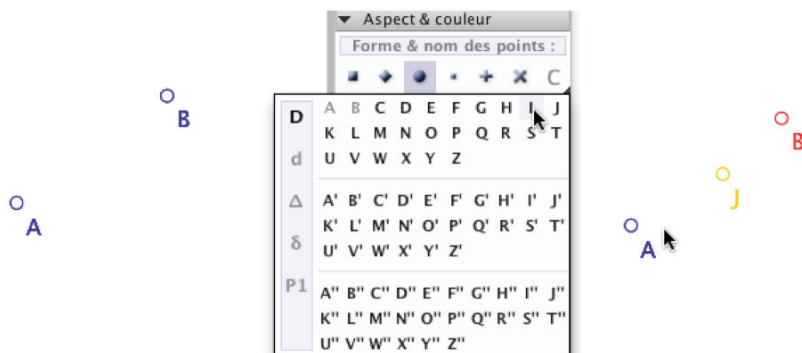
simple : si on hésite sur l'ordre des objets à montrer, on privilégiera toujours celui qui se termine par un point pour engager cette anticipation. Cela s'applique aussi aux macro constructions.



Application d'une macro qui traite le cas $n = 2$ du théorème TCIE¹

L'illustration précédente propose deux instances, à partir des deux mêmes premiers points, d'une macro construction au moment où elle attend un dernier point. La macro construit le triangle, et partage celui-ci, à partir du premier sommet, en deux triangles ayant des cercles inscrits de même rayon.

Même le nommage automatique des points est anticipé :



Dans la partie gauche, alors que deux points A et B sont donnés, on sélectionne dans la palette d'outils le nom I comme nom du futur point cons-

¹ Sangaku connu en Occident sous le nom de *Théorème des Cercles Inscrits Égaux*. On trouvera des figures dynamiques et une analyse didactique de ce Sangaku dans les articles suivant :

<http://www.reunion.iufm.fr/recherche/irem/spip.php?article148>

<http://www.reunion.iufm.fr/recherche/irem/spip.php?article155>

truit. Dans la partie droite, on observe l'effet de la construction quand on veut prendre le milieu I de B et A : B a été sélectionné, alors le logiciel anticipe que le prochain point cliqué se nommera I et donc appelle naturellement J le milieu préconstruit. Bien entendu dès qu'on clique sur B, c'est le nom I qui est donné au milieu.

Pourquoi les autres logiciels sont-ils passés à côté de cette fonctionnalité ?

Avec le recul de l'expérience et de l'histoire de la GD, il peut être intéressant d'essayer une première analyse de cela. Nous allons le faire sur notre référent historique, Cabri-géomètre, puisque ce sont les travaux et réflexions didactiques sur ce logiciel qui ont ouvert les portes à l'ergonomie que l'on connaît des logiciels de GD. Le fait que les autres aient pris les mêmes choix que Cabri tendrait à donner raison à Colette Laborde² quand elle remarquait que sur les 70 logiciels de géométrie dynamique qu'on peut répertorier, 60 sont pour l'essentiel des clones de Cabri.

Les plus anciens de ses utilisateurs se souviennent des toutes premières versions de Cabri. Pour tracer une parallèle il fallait montrer dans un ordre précis, par exemple, la droite et le point. L'item attendait l'un puis l'autre, et malgré cet engagement direct du retour textuel cognitif (« cette droite », « ce point ») l'utilisation ne paraissait pas naturelle. Plusieurs personnes ont suggéré que cela serait plus simple, pour les élèves de collège en particulier, si, quand les objets sont de types différents, on pouvait montrer les objets dans un ordre quelconque. Malgré la difficulté de la tâche, ce fut implémenté, et salué par chacun de nous comme un progrès de l'engagement direct sur les items de construction. En effet, l'intention de l'élève de construire une parallèle à une droite passant par un point était désormais encore mieux reconnue par le système puisque, l'outil sélectionné, il pouvait montrer en premier indifféremment le point ou la droite comme premier objet. Cabri-géomètre étant l'étalon de référence en géométrie dynamique, les logiciels suivants se sont généralement contentés de reproduire ce fonctionnement.

Dans CaRMetal il n'y a pas l'indépendance des objets de construction d'un item, mais il y a cette règle simple que « tout est construit autour de l'anticipation », donc on sait que le dernier objet, quand cela a du sens, est un point. Par ailleurs, ce manque d'indépendance est largement compensé par la réaction immédiate du surlignage de l'objet quand il peut être sélectionné (comme avec Geogebra).

² Intervention plénière lors du colloque de l'IREM de Lille en juin 2008.

Il ressort de cette première analyse que la perception d'une modalité interne aux items de construction est vite apparue à l'usage. L'analyse didactique, portée sur une plus grande reconnaissance de l'intention de l'utilisateur, a alors pris des chemins qui, s'ils furent significatifs à cette époque de la mise en œuvre de la GD, n'étaient en définitive pas si optimaux que cela. À l'usage, après des années de pratique, un recentrage sur l'apprenant et une interface accompagnante semblent plus efficaces.

2. La question de la modalité

Un outil est dit *modal* si son activation provoque une interruption de la tâche en cours pour être appliqué. C'est typiquement une boîte à dialogue par laquelle un élève modifie une couleur ou un aspect d'objet de la figure sur laquelle il travaille, ladite boîte cachant généralement la figure (c'est régulièrement plus de 25 % du temps devant une figure avec logiciel de GD « modal » en classe de Sixième). La modalité est un des aspects les plus flagrants du non respect de la quatrième règle de Shneiderman, en particulier sur la non rapidité de l'action, avec le blocage des boîtes à dialogue alors que l'on peut faire toujours autrement.

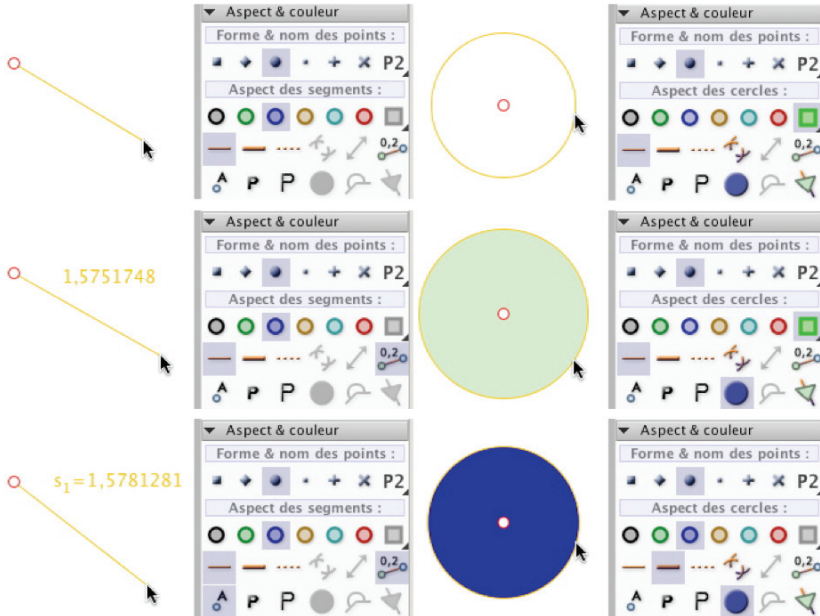
CaR, comme d'autres logiciels de GD, contenait de nombreuses situations modales qui ont été rapidement enlevées dès les premières versions de CaRMetal. C'était en fait le principal but initial de l'auteur : que l'utilisateur ait toujours devant lui sa figure quoiqu'il fasse.

Mais cette question de la modalité va plus loin que ce premier aspect. Si on étend cette notion d'interruption à l'indisponibilité temporaire de la manipulation directe, alors les comportements des outils de construction qui ne créent les objets qu'après le dernier clic ont, en dernière analyse, des comportements internes modaux. Cela n'était pas flagrant au début de la géométrie dynamique, cela le devient quand on a une autre implémentation que celle qui est devenue de fait un standard.

A contrario, une autre situation d'a-modalité était déjà présente dans les premières versions de Cabri II, celle des *aspects* des objets dans la palette *Attributs*. Cela montre bien que la réflexion sur l'a-modalité était bien présente³.

³ Elle l'a été très tôt. Par exemple le popup menu de *l'ambiguïté des objets* est un autre standard installé par Cabri dès 1989, qui répond parfaitement à la règle 4 de Shneiderman – disponibilité et rapidité – alors que la toute première version 1.0 de 1988 proposait une immense boîte à dialogue pour sélectionner « le premier » ou « le

Absente de CaR, cette a-modalité des aspects des objets a été rétablie dans CaRMetal et largement enrichie. Outre les aspects usuels standards, tous ceux spécifiques à chaque objet comme le remplissage des cercles en disque, des polygones, l'affichage des mesures des longueurs peuvent tous être modifiés pendant la création de l'objet et avant le dernier clic pour cet objet.



De haut en bas à gauche, dans une même séquence, on prend un segment et en cours de création on ajoute la mesure de sa longueur, puis son nom.

À droite, on prend un cercle, puis en cours de création on ajoute un remplissage du disque, et enfin on change son aspect et sa couleur avant la création finale.

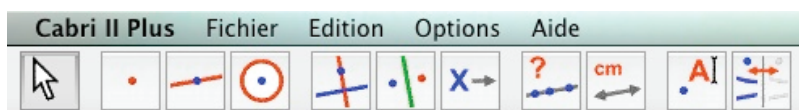
Repérer la modalité cachée

En poursuivant la réflexion, on peut étendre notre questionnement sur la modalité aux représentations de la connaissance ou aux modes d'organisation internes au logiciel, dont on pourrait dire qu'elles sont potentiellement porteuses de modalité. On aborde alors la question des modélisations implicites

dernier » objet. Cela illustre aussi l'effort, conceptuel plus que technique, qu'il y avait à faire pour considérer qu'une procédure de construction pouvait être modale.

– pragmatique au début des EIAH – soit des connaissances, soit des interactions homme machine.

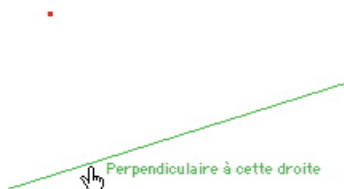
De ce point de vue, on peut compléter l'analyse de la non anticipation des outils dans Cabri II. La distinction, proposée dans les manuels d'utilisation du logiciel – reprise pendant des années dans les formations – entre les outils dits *de création* (suite de trois icônes) de ceux dits *de construction* (autre suite de trois icônes), bien entendu fondée et signifiante, participe aussi d'une représentation porteuse de modalité. Cette perception cognitivement différente des deux outils création/construction induit un traitement de l'engagement direct différent des deux situations au point qu'on ne perçoit pas la modalité induite.



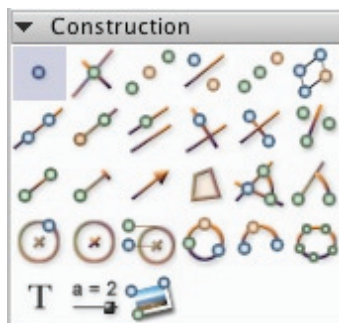
Après le pointeur, les trois types de création puis les trois de construction

Les raisons d'une perception cognitive différente de la création et de la construction peuvent être nombreuses et croisées. Il peut y avoir la dimension procédurale des constructions et l'unicité du résultat : étant donné une droite et un point, il n'y a qu'une parallèle/perpendiculaire à la droite passant par le point, il suffit donc de la créer. D'un point de vue constructiviste – le modèle sous-jacent des micromondes – il n'y a pas d'engagement direct à implémenter comme dans une phase de création, où tout est possible. On voit bien qu'il y a là, du côté de la conception, un effort à faire pour déceler un type de modalité dans cette vision de la construction d'un objet.

Une autre raison est peut-être liée à l'organisation générale du logiciel dans son traitement de l'engagement direct. Pour renforcer la perception de l'utilisateur dans sa dimension cognitive, le choix a été d'avoir un retour textuel de l'engagement direct pour les constructions.



Beaucoup d'énergie a été mise dans cette gestion – surtout dans son traitement multilingue complexe. Il a pu en résulter une centration sur les prémisses de chaque outil (comme le repérage du genre de l'objet dans la langue en cours, etc.) pour affiner ce qui a fait ensuite la spécificité du logiciel et son gage de qualité. Au point que, à part René Grothmann – qui, paradoxalement, avait par ailleurs un logiciel profondément modal – personne n'avait vraiment perçu la possibilité bien plus centrée sur l'apprentissage par accompagnement des schèmes d'action de l'implémentation *des outils*... tout simplement, car dans cette démarche, il n'y a pas de distinction entre création et construction. Une parallèle est d'abord une droite crée qui a de plus une propriété, celle d'être parallèle à une autre. D'où dans CaRMetal la palette dite de construction⁴ où par ligne, les objets sont – si possible – du même type : points, puis droites, segments et cercles.



Mais pourquoi cette quête d'a-modalité ? Pour en faire quoi ? Est-ce uniquement pour des considérations ergonomiques ? Voyons que non, que sous cette recherche de l'a-modalité, de nouvelles démarches ou de nouveaux outils peuvent émerger.

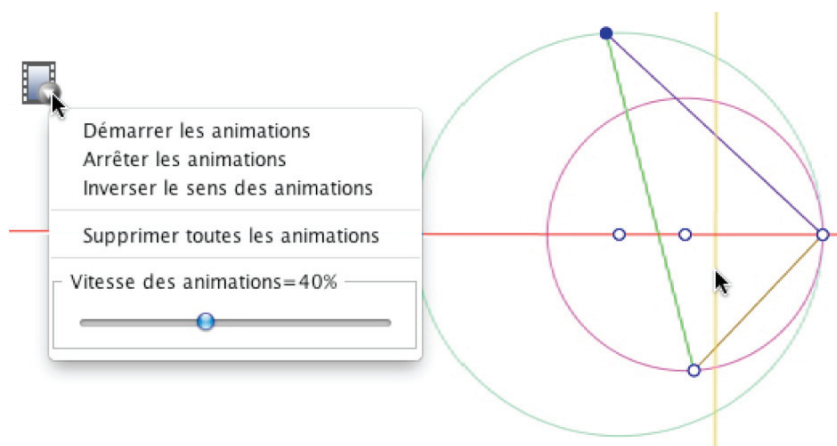
L'extension de l'a-modalité : une source de nouvelles démarches d'investigation...

Un exemple significatif est celui de l'animation, qui a longtemps été bloquée dans tous les logiciels, désormais a-modale dans la version 3.5. Cela signifie que l'on peut continuer à faire des constructions dans la figure pendant les animations. L'a-modalité de l'animation ouvre de nouveaux champs

⁴ L'analyse est trop rapide, il conviendrait de la nuancer. Par exemple en CM1, l'enseignante que nous suivons, avec cette palette, utilise oralement « créer un point », « construire une droite ». Les élèves, eux, « créent » des droites.

d'investigation. Pour rechercher des invariants, on peut animer des points sur des objets et, pendant l'animation, conjecturer puis, infirmer ou confirmer ces conjectures par des constructions qui peuvent être testées – toujours en cours d'animation – par les outils de test appropriés.

Par exemple dans la figure suivante, le point bleu tourne autour du cercle extérieur. La construction est un classique de la recherche d'invariant (l'autre centre d'homothétie des deux cercles que le point de contact). Traditionnellement on conjecture le point invariant par l'animation du point bleu, puis on cherche une construction, et on confirme la conjecture en animant à nouveau le point avant d'entreprendre une démonstration. Ici on peut désormais faire les constructions pendant l'animation. Sur cet exemple la visualisation de l'invariant reste élémentaire car la droite des centres est fixe, mais dans d'autres situations, la visualisation de l'invariant ne peut s'obtenir qu'à travers l'animation.



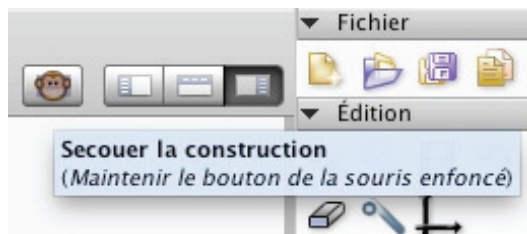
La construction de la perpendiculaire est faite pendant l'animation

... et de nouveaux outils

Un chantier important dans la réécriture du cœur du logiciel originel a été l'a-modalisation des exercices⁵ pendant leur réalisation par les élèves. Cela

⁵ CaR comprend une possibilité de faire des exercices. Mais la situation est totalement modale, l'exercice se fait dans une instance de la figure. On est dans la situation, dégagée très tôt par Nicolas Balacheff d'un simple prototypage uniquement pragmatique de certaines parties des EIAH sans modélisation de l'ensemble. Cet outil du logiciel

suppose des procédures de validation très différentes de celles employées dans le cas modal. La mise en place de ces procédures a rendu possible la création de l'outil *Monkey* qui, dans le cadre de l'auto-validation, permet aux élèves, d'un clic, de « secouer » leur figure pour valider la construction.



L'outil Monkey, directement issu de l'a-modalisation des exercices

Cet outil fait prendre des valeurs aléatoires aux objets de base – en restant dans la fenêtre de travail – tant que l'on maintient le bouton de la souris enfoncé. L'utilisateur teste lui-même la construction et sa robustesse dynamique, typiquement pour vérifier qu'il a bien fait une figure dynamique et non pas un dessin. Ainsi un carré construit de manière perceptive ou partiellement perceptive ne résistera pas au « secouage » du logiciel.

Mais, en terme d'instrumentalisation, cet outil peut avoir bien d'autres utilisations. Ainsi dans des activités de cycle 3 de l'école primaire ou de début de collège, des constructions même grossièrement fausses peuvent passer par des instances proches de ce que l'on a demandé aux élèves de réaliser, et alors activer une représentation de la construction à accomplir.

Cette instrumentalisation du Monkey, même si elle n'a pas à être systématisée, peut avoir un intérêt dans certaines situations, voire être un élément de réponse à la critique récurrente de la difficulté à rentrer dans la pratique des micromondes géométriques : une nouvelle partie de la tâche, quand elle est préparée, peut être prise en charge par l'EIAH.

Une instrumentalisation tout à fait différente peut être entreprise dans un contexte statistique, un clic maintenu sur le Monkey permettant toutes sortes de situations. Le travail d'uniformisation interne au logiciel fait que les nouveaux exercices de CaRMetal s'appliquent, de fait, à des scripts de programmation. Il en résulte que le module d'exercice est aussi (pour les élèves) un

n'est pas examiné ici, mais il est intéressant. Même des lieux peuvent être un objet de construction évalué par le logiciel. Pour plus de détail, voir l'onglet *Exercice* de cette page : <http://www.reunion.iufm.fr/recherche/irem/spip.php?article347>

vérificateur de programme ou un outil de *scripticiels* pour les enseignants. La poursuite de la réflexion sur l'a-modalité et des mises en œuvre associées sont une source de progrès réguliers de l'engagement direct en géométrie dynamique.

3. La logique dans les figures – Géométrie logique

L'insertion d'éléments de logique dans les figures est, d'une part, un besoin qui s'est fait sentir très tôt et, d'autre part, un domaine dans lequel les interfaces et les outils disponibles ont beaucoup progressé.

Un peu de préhistoire

L'expression *géométrie logique* remonte à 1993 quand, lors de la première Université d'été de géométrie dynamique (en fait « de Cabri-géomètre » dans sa première version) avec Charles Payan, de l'université Joseph Fourier, nous avons tous les deux développé indépendamment des techniques géométriques pour introduire des comportements logiques dans les figures. L'importance (au moins opérationnelle) de ce concept a même été à l'origine de la revue papier *abraCAdaBRI* qui s'est ensuite transformée en site dont on peut encore lire les pages d'Alice⁶.

Même si cela n'a pas été repris dans le site en ligne, nous avons développé à cette époque, dans la revue papier *abraCAdaBRI*, la notion de CLFC : combinaisons linéaires de fonctions caractéristiques⁷... entièrement en constructions géométriques, alors que la même chose se fait maintenant par un traitement algébrique des booléens. Les CLFC avaient été ensuite reprises et « popularisées » par Roger Cuppens dans son ouvrage *Jouer... avec Cabri-géomètre* (1996-1999).

⁶ <http://www-cabri.imag.fr/abracadabri/GeoLogique/AliceGene.html>

L'ombre du tiroir d'une commode avait été présentée en 1993, avec un logiciel qui n'acceptait alors que 500 objets maximum dans une figure.

⁷ <http://icosaweb.ac-reunion.fr/GeomJava/abraCAda/Docs/hacker/abra03.pdf>

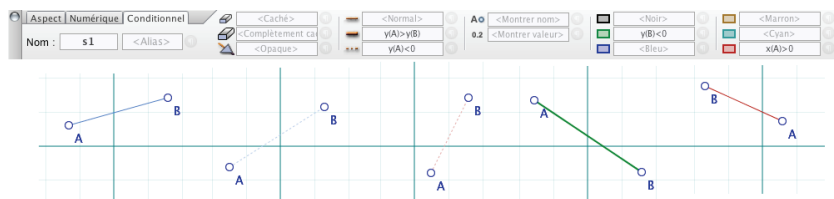
Les amateurs d'histoires anciennes peuvent s'amuser à feuilleter les pages d'Alice de l'ancienne revue papier (les pages de la rubrique Espace ou celles du hacker sont intéressantes aussi) conservées ici :

http://icosaweb.ac-reunion.fr/GeomJava/abraCAda/M_abra.htm

On y voit de la passion pour la géométrie dynamique – et donc pour Cabri à l'époque – mais aussi le chemin parcouru depuis par les logiciels.

L'onglet *Conditionnel* de l'inspecteur d'objets

La géométrie logique de cette époque, entièrement géométrique, ne pouvait contenir que des « if ... then » très limitatifs. Implémentée dans CaR, mais rendue bien plus accessible avec CaRMetal par un onglet spécifique de l'inspecteur d'objets – dans lequel on traite des objets déjà construits – la géométrie logique s'est largement simplifiée tout en s'enrichissant, car tous les aspects d'un objet peuvent être conditionnés par des expressions algébriques liées à la figure.



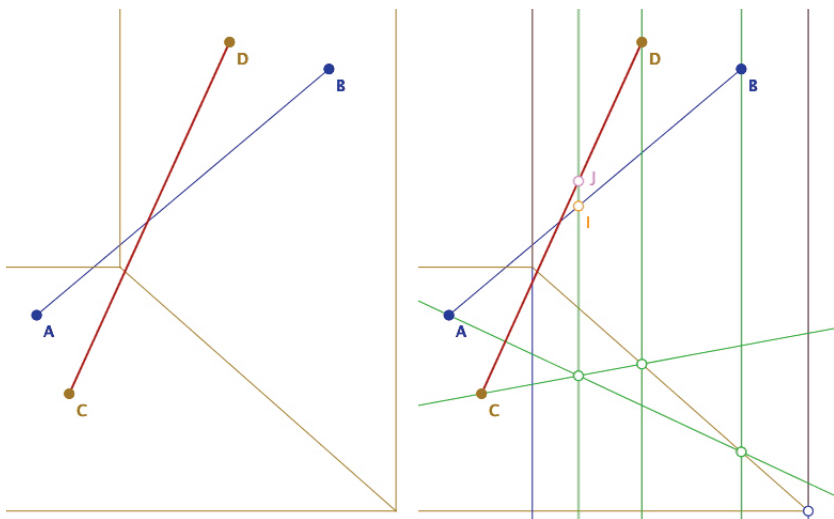
L'illustration précédente montre l'onglet *Conditionnel* d'un segment $[AB]$ qui prend des aspects différents selon les positions absolues ou relatives de A ou de B (gras si $y(A) > y(B)$, pointillé si $y(A) < 0$, rouge si $x(A) > 0$, vert si $y(B) < 0$). Rien qu'avec cet onglet, on peut traiter des situations réellement complexes comme on le verra quand nous aborderons la géométrie finie avec l'aimantation algébrique.

Voici un exemple tout à fait élémentaire, et assez prototypique de ce type de traitement pour enrichir les simulations que l'on peut proposer en classe. Deux segments $[AB]$ et $[CD]$ reposent sur le sol et sont posés contre un mur (ici ils sont élastiques car on peut déplacer les extrémités). On se propose de mettre, dynamiquement⁸, en gras celui qui est au-dessus de l'autre.

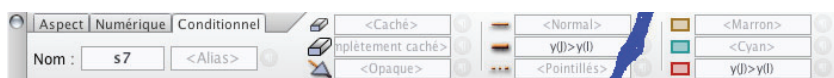
Sur l'illustration de droite, on voit que, quand les projections orthogonales sur le sol se coupent dans la partie où il y a une solution (du même côté que les pieds), ce point a alors deux antécédents⁹ I et J sur les segments. C'est la clé de la construction.

⁸ Il s'agit d'une transcription dynamique d'un exercice classique sur l'incidence et la projection orthogonale, puisé dans un ancien manuel Istra 1^{re} S, IREM de Strasbourg.

⁹ En classe de 1^{re} S, il y a déjà quelques années, j'avais dû apporter de la ficelle et utiliser des élèves pour les extrémités des segments, et d'autres élèves pour tenir les ombres au sol, pour mettre en scène cet exercice. Il est intéressant de remarquer que ce n'était alors pas tant pour « confirmer expérimentalement » (pas assez de précision dans la manipulation finale de la droite (IJ)), mais simplement mettre en situation la



Pour finaliser la figure avec l'onglet « conditionnel » de CaRMetal, il suffit alors de conditionner¹⁰ l'aspect *gras* et l'aspect *rouge* du segment sélectionné en fonction de l'ordonnée des points I et J :



Il faut bien entendu faire ceci sur les deux segments, en inversant le sens de l'inégalité, car le conditionnement est propre à chaque objet.

Les booléens de CaRMetal

Les expressions algébriques de CaRMetal peuvent être des booléens. Ou, plus exactement, les expressions qui valent 0 ou 1 sont considérées en interne comme des booléens. L'intérêt est de les utiliser à l'intérieur des coordonnées des objets et plus particulièrement des points. Par exemple pour un point U qui peut prendre quatre positions différentes I, J, K, O, positions régies par

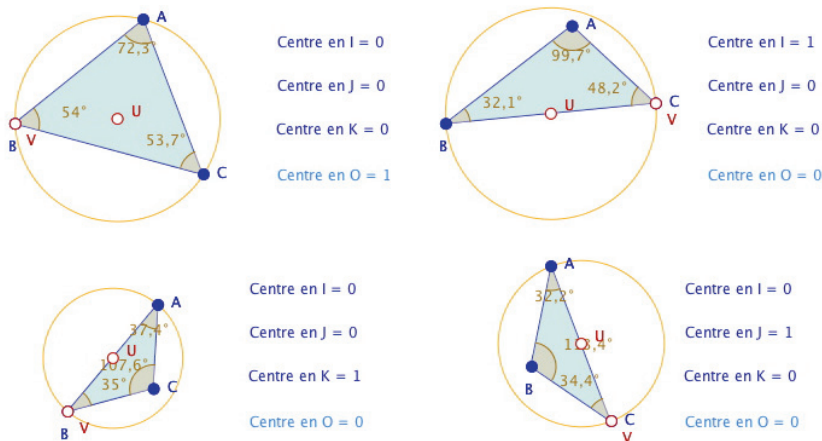
plausibilité du raisonnement, car les élèves avaient alors eu besoin de raisonner non pas sur des figures, mais *sur des objets*.

¹⁰ Les plus curieux pourront voir sur cette page : <http://www-cabri.imag.fr/abracadabri/GeoLogique/ExemplesAlice/AliceSegment.html> comment il fallait procéder, du côté de l'enseignant, quand tout devait être traité géométriquement.

des booléens associés EnI , EnJ , EnK , EnO , on peut donner pour U les coordonnées :

Aspect	Numérique	Conditionnel	X :	$EnI \cdot x(I) + EnJ \cdot x(J) + EnK \cdot x(K) + EnO \cdot x(O)$
Nom :	U	<Alias>	Y :	$EnI \cdot y(I) + EnJ \cdot y(J) + EnK \cdot y(K) + EnO \cdot y(O)$
			☒ Fixe	

On construit alors un *unique* point U qui correspond à quatre cas de figure. Il s'agit ici de construire *le* cercle – de centre U donc – de plus petit rayon contenant un triangle ABC donné. Le cercle construit est unique et rend compte des quatre cas qui sont déclinés dans une preuve écrite¹¹.



Un exemple comme celui-ci permet, sur un problème simple, à la fois de travailler la logique au lycée et de la mettre en œuvre concrètement dans un logiciel. C'est aussi l'occasion de montrer comment, dans un EIAH moderne, on peut réifier l'unicité d'une solution mathématique que le discours hypothético-déductif sépare et le compte rendu textuel séquence.

Dans l'approche « préhistorique », on voit que les CLFC étaient déjà un traitement en pur silex géométrique de ces mêmes questions, avec des contraintes particulières parfois complexes, à prendre en considération¹².

¹¹ La construction et son utilisation en classe sont détaillés dans cet article <http://www.reunion.iufm.fr/recherche/irem/spip.php?article65>

¹² Sur l'exercice de section d'un cube, connu sous le nom de « la copie de Bergson » <http://icosaweb.ac-reunion.fr/GeomJava/abraCAda/Docs/hacker/abra05.pdf>

Désormais incorporés à la géométrie dynamique, accessibles aux coordonnées de points, les booléens permettent d'aller encore plus loin en incorporant les combinaisons linéaires de booléens dans les CarScripts. On peut alors placer sur chaque point des dizaines de valeurs différentes sans effort particulier¹³.

Note technique sur l'usage des booléens : tous les langages de programmation ont une (forme simplifiée de) évaluation paresseuse des booléens, c'est-à-dire que si dans $a \ \&\& \ b$, a est faux, b n'est pas évalué, ou si dans $a \ || \ b$, a est vrai, b n'est pas évalué. C'est parfois significativement intéressant si, dans des situations particulières d'une construction¹⁴, certaines parties n'existent pas : le traitement paresseux bien utilisé peut alors éviter des constructions plus complexes. Si a et b sont des booléens, $a * b$ donne le même résultat que $a \ \&\& \ b$, à l'évaluation paresseuse près, justement.

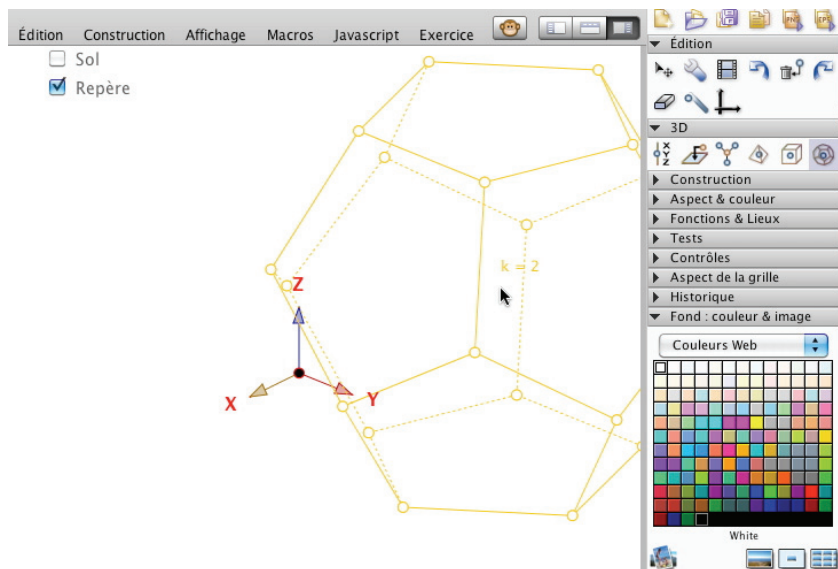
4. L'environnement 3D de CaRMetal

La question se pose de la pertinence, dans un système d'EIAH, d'ajout de simulation de situations hors du modèle qui fonde ce système. Nous avons insisté sur l'importance de la cohérence du système. Alors quelle cohérence interne peut-on maintenir avec un tel ajout ? Pour quels usages ? Dans le cas de CaRMetal nous sommes face à un logiciel qui n'est certainement pas un logiciel 3D, et qui pourtant a cherché à maintenir une cohérence en proposant un environnement 3D largement opérationnel pour ce qu'il peut faire. Voyons ce qu'il en est.

Quand on choisit de faire une figure 3D, un repère s'affiche ainsi qu'une nouvelle barre dans la palette d'outil. Ci-dessous, saisie de l'environnement en cours d'utilisation :

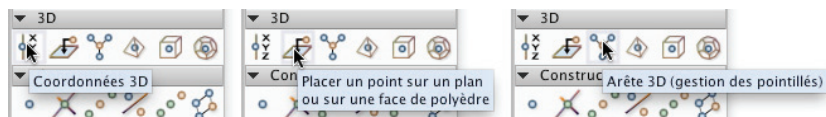
¹³ C'est ce qui a été fait – comme exercice d'école – dans le premier onglet de cet article : <http://www.reunion.iufm.fr/recherche/irem/spip.php?article326> où l'on voit comment placer sur 8 points, les 92 positions de chacune des reines d'un échiquier dans le problème des 8 reines, un curseur balayant les 92 solutions. Les autres onglets en donnent une utilisation en statistique qui donne à un exercice classique une dimension « manipulation directe » sur la quelle nous reviendrons.

¹⁴ Par exemple la prise en compte de l'éventuelle seconde intersection d'un cercle avec un segment qui, par construction, en a toujours au moins une connue.

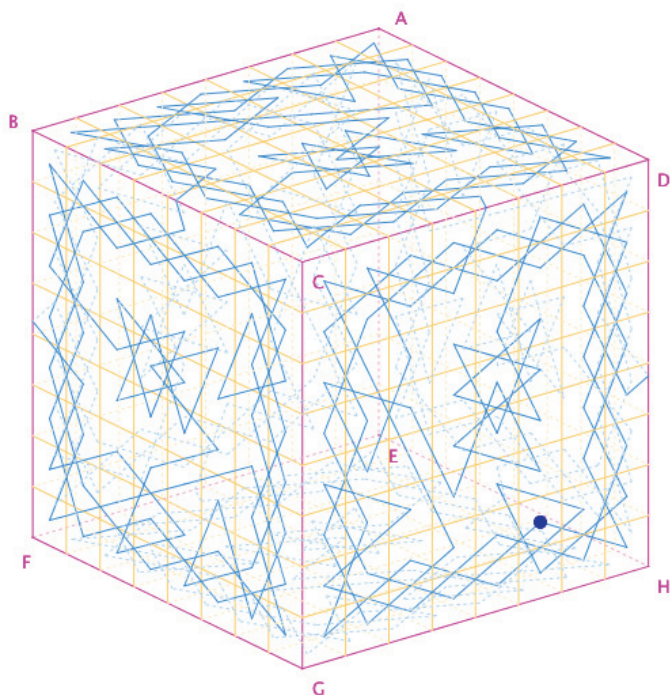


On observe à nouveau l'anticipation de la construction, y compris des arêtes cachées. Un « clic droit glisser » permet de déplacer le trièdre en manipulation directe, dans une projection axonométrique.

Le logiciel propose ces premiers outils de base :

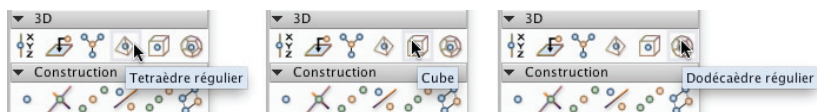


L'outil *Arête 3D* est souvent pris pour ce qu'il n'est pas – une gestion des parties cachées sur les faces – alors qu'il ne fonctionne *que* pour les arêtes, et pas sur les segments de ces faces. Cela montre que l'attente de l'utilisateur est souvent plus proche d'un environnement 3D. Dans la figure suivante, on distingue en pointillé les parcours sur les faces non visibles du cube. Elles sont gérées, par script, à partir des faces cachées.

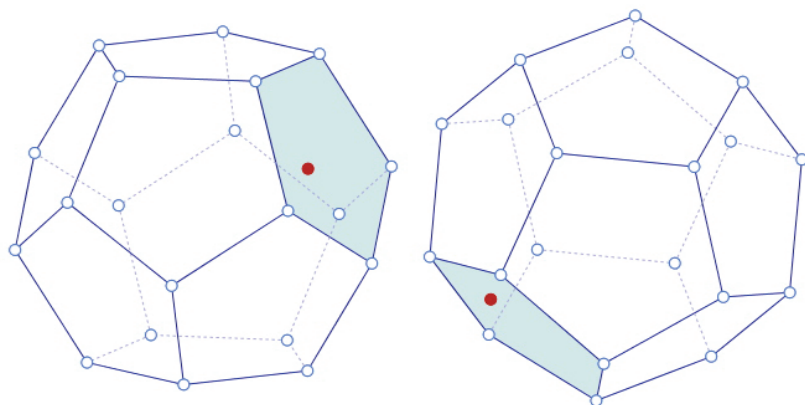


Parcours fermé dynamique du cavalier d'échec sur un cube

On trouve ensuite trois des cinq polyèdres réguliers, avec gestion des arêtes cachées :

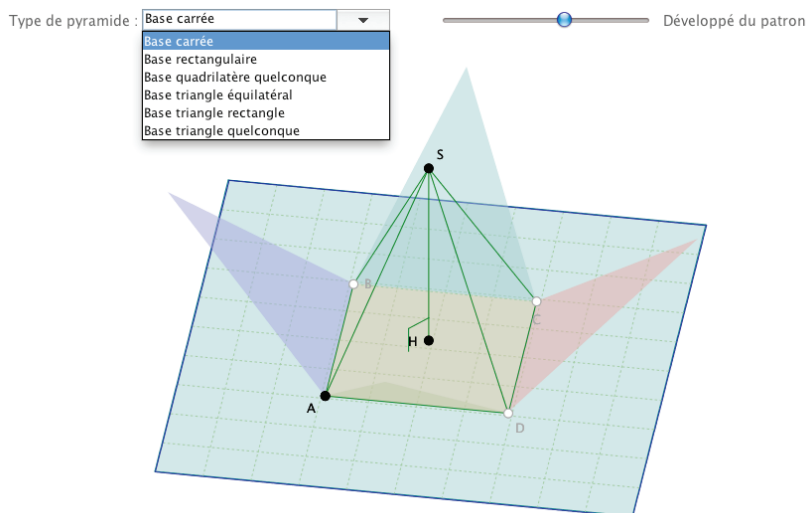


Pour conserver sa cohérence le logiciel propose de travailler essentiellement sur les polyèdres et l'incidence, même si d'autres objets peuvent être construits. Alors, l'outil *Point sur face* proposé rend bien compte de ce qu'il est possible de faire dans ce type d'environnement. Pour l'utiliser, il faut d'abord construire le polygone associé à la face. Il peut être plein ou non. On pose tout simplement un point avec cet outil, le point est alors *sur la face* et conserve sa position barycentrique sur la face dans la rotation, comme on le voit sur ce dodécaèdre dans deux positions du trièdre de référence :



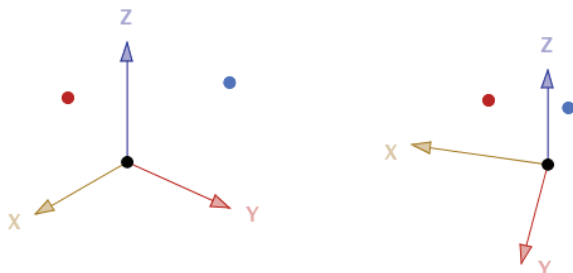
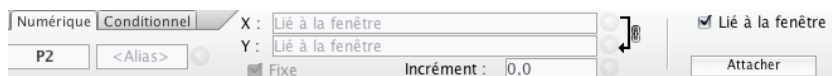
L'environnement proposé est éloigné d'un environnement réellement 3D comme peut l'être celui de Cabri 3D – qui reste actuellement le seul environnement 3D en manipulation directe. Il propose néanmoins des outils simples d'accès, adaptés à son utilisation, l'enseignement de la géométrie dans l'espace dans l'enseignement secondaire.

Travailler en 3D est toujours long quel que soit le logiciel. Aussi trouve-t-on des réalisations sur la toile, comme ce patron de pyramide multibase :

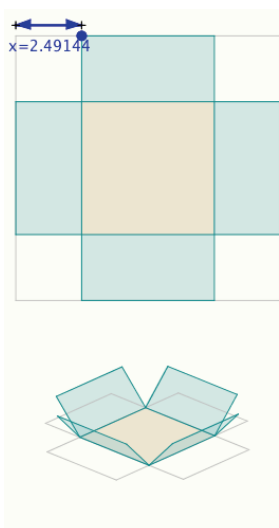


La gestion conjointe 2D et 3D

L'auteur de CaRMetal a développé des outils, d'utilisation élémentaire, pour permettre d'avoir dans la même figure une représentation 2D et une autre 3D liées entre elles. En pratique, dans une figure 3D, il suffit de créer un point « lié à la fenêtre » pour que ce point soit indépendant de la rotation du trièdre. Alors toute construction faite à partir de lui est aussi indépendante du trièdre.

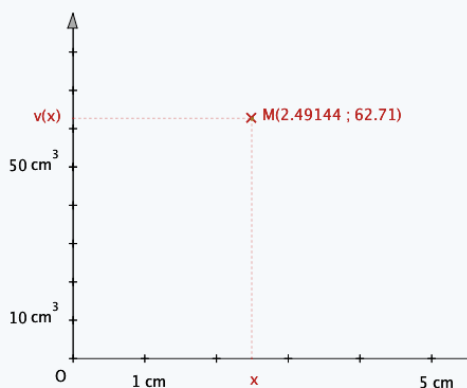


Le point bleu n'est pas lié à la fenêtre, le rouge est lié



On note v la fonction qui à x associe le volume de la boîte.

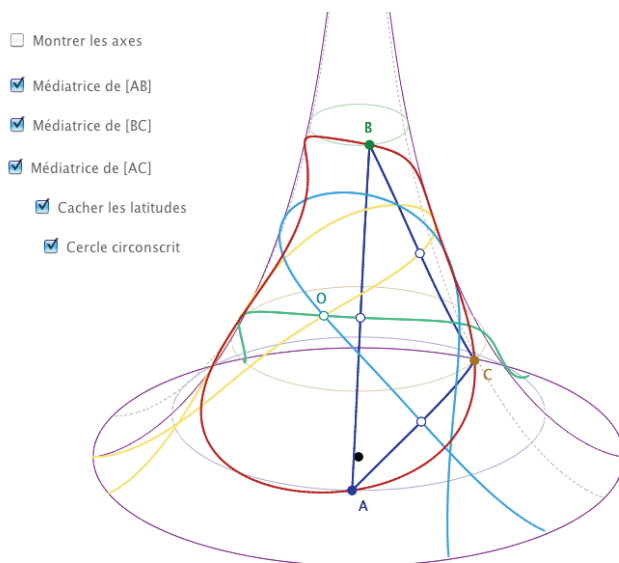
Déplacer le point bleu : cela fera apparaître dans le repère ci-dessous la représentation graphique de la fonction v .



Avec cet outil, on peut réaliser des constructions comme la précédente¹⁵, où non seulement la boîte est manipulable à la souris (trièdre) mais l'ouverture des côtés aussi par un curseur caché mais actif et accessible, juste sous la boîte.

Exemple du travail interne nécessaire à la cohérence des outils

Cette partie 3D de CaRMetal peut avoir des utilisations dans l'enseignement supérieur, par exemple pour illustrer, en manipulation directe, le concept de géodésique ou celui de géométrie non euclidienne dans un environnement où c'est la géométrie intrinsèque de la surface. C'est le cas des surfaces à courbure constante négative, dont la plus simple est la pseudosphère. Beltrami a montré en 1868 que la géométrie de cette surface est la géométrie hyperbolique de Lobachevsky. En voici une illustration avec CaRMetal :



*Médiatrices d'un triangle et cercle circonscrit sur la pseudosphère
(ici ce cercle existe)*

¹⁵ Les deux figures de ce paragraphe sont sur le site de CaRMetal.

C'est l'occasion de préciser un point de détail qui montre le travail interne nécessaire à la cohérence du plongement de la surface dans l'espace, pour son accessibilité à la manipulation directe. Les sommets du triangle sont définis par une altitude (ici cachée pour l'esthétique de la figure) qui donne la position d'un cercle parallèle à l'équateur de la pseudosphère, et une longitude prise sur ces cercles. Dans la représentation 3D, les cercles sont des ellipses. Il faut que, quand la pseudosphère tourne, par exemple selon (OZ), les points tournent en même temps pour être conforme au mouvement de la surface dans l'espace. Même si la situation est très simple ici, il y a un traitement à ajouter aux *points sur objet* d'une conique en fonction des points constituant de la conique pour obtenir un comportement, en 2D, conforme à la manipulation 3D. Par défaut, ce traitement n'existe pas dans un logiciel seulement 2D, il est tout simplement sans objet.

5. Déterminisme et continuité

En géométrie dynamique, on dit d'un logiciel qu'il est *déterministe* si la figure revient à sa position initiale quand ses objets constitutants, après manipulation, sont replacés dans leur position de départ. Dans un cadre scolaire, il est clair que le déterminisme – avec d'autres propriétés comme un fonctionnement intrinsèquement affine dans la déformation des figures – est dans le noyau du cahier des charges d'un logiciel de GD. Une autre propriété est dans ce noyau, la continuité, pendant la manipulation directe, des intersections des objets.

Mais la continuité et le déterminisme dans le même cahier des charges, est-ce compatible ? Ce problème est abordé, dans un cadre plus général, par la conjecture d'isotopie (Ringel, 1956), infirmée par Mnev en 1985 (donc avant la création des logiciels de géométrie dynamique). En résumé « la continuité et le déterminisme des constructions sont incompatibles : il faut privilégier l'un ou l'autre, on ne peut maintenir les deux simultanément jusqu'à leurs conséquences » (Genevès, 2004, p. 17).

Comme déjà mentionné en introduction, en même temps que Cabri se faisait en France par l'équipe de Jean Marie Laborde, un remarquable projet a été mené dans le cadre de la continuité, en Allemagne, par Richter-Geber et Kortenkamp, le logiciel Cinderella. Les auteurs ont, entre autres résultats, montré que la question de la continuité nécessitait un plongement dans un plan projectif complexe (Genevès, 2004, p. 23). Pour une utilisation scolaire, la version 2 de Cinderella est redevenue déterministe.

Maintenir la continuité

Une des tâches des pionniers de la géométrie dynamique a donc été de concilier les contraintes de déterminisme avec des choix de continuité. Non seulement on arrive, pour des figures usuelles (cercles, droites) à gérer correctement la continuité des intersections, mais les auteurs de Cabri ont inventé deux types d'intersections, l'intersection d'objets et l'intersection *à la volée*, où l'on désigne celle des deux intersections que l'on veut. Alors pour l'intersection d'une droite et d'un cercle, l'intersection à la volée résiste au passage à l'infini du rayon du cercle (« persistant naming problem »), alors que ce n'est pas le cas des points obtenus par intersection de deux objets.

En terme de continuité, c'est une avancée importante qui a été reprise par les logiciels de nouvelle génération comme Geogebra et CaR. L'utilisateur retiendra que pour mieux préserver la continuité dans les figures, quand il y a plusieurs intersections, il est préférable de prendre l'intersection à la volée.

Et essayer d'aller plus loin

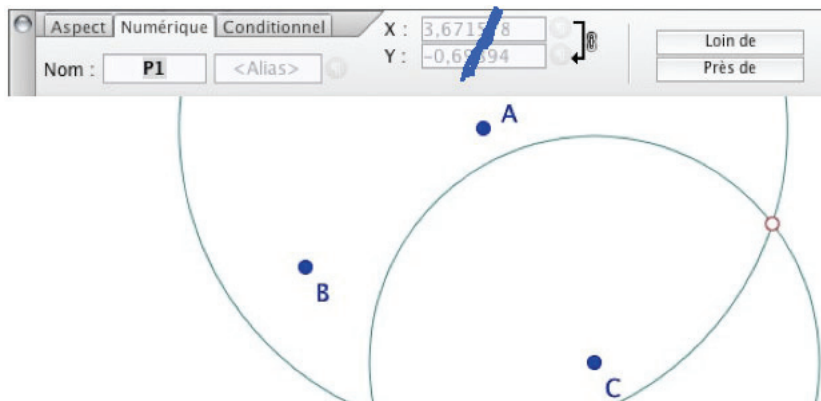
Mais ces deux logiciels ont fait un effort supplémentaire pour installer la continuité le plus loin possible¹⁶. On sait d'avance qu'il n'y a pas de proposition totalement satisfaisante, mais cela n'empêche pas d'améliorer quelques outils. Dans Geogebra l'utilisateur peut « activer » une *option de continuité* générale à une figure. Alors la continuité de l'intersection d'un cercle et d'une droite est satisfaite quand le cercle passe par un rayon infini, sans même qu'on ait fait l'intersection à la volée. CaRMetal, propose pour les points d'intersection, une personnalisation individuelle plus algorithmique.

Mais la saisie à la volée de l'intersection de deux cercles n'est continue dans aucun logiciel¹⁷. C'est le cas dans la construction très usitée du quatrième point du parallélogramme au compas. La démarche algorithmique de CaRMetal¹⁸ est alors particulièrement efficace :

¹⁶ Cabri II+ a travaillé sur la continuité de l'intersection des objets – autres que droites – avec les coniques. Dans ce domaine il est celui qui va le plus loin – et de loin – et arrive à des résultats largement suffisants.

¹⁷ Plus de détail dans l'onglet *Exercice* de cet article : <http://www.reunion.iufm.fr/recherche/irem/spip.php?article347>

¹⁸ Héritée de CaR. L'implémentation est de René Grothmann.

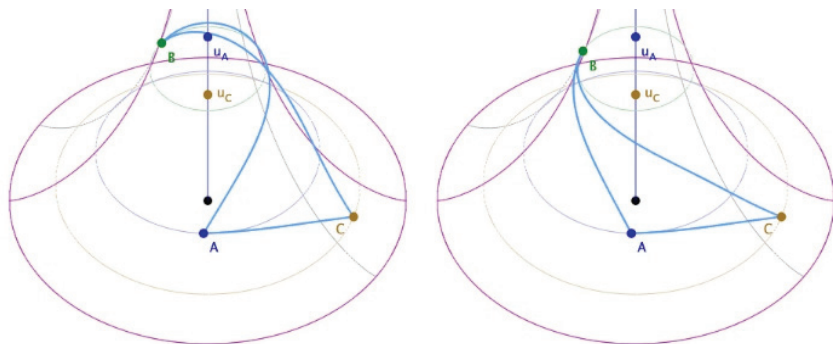


En cliquant sur l'intersection à la volée des deux cercles, on peut ensuite affiner l'algorithme interne de l'intersection en précisant le point comme étant « loin de » B. Alors la continuité est préservée. Ce n'est qu'une nouvelle application d'un algorithme pour préserver le « persistant naming ». Il est seulement ici plus transparent, et se révèle souvent pertinent.

On le voit, les choix des deux logiciels sont différents. Le problème difficile de l'intersection de deux arcs de cercles (intersection de deux droites hyperboliques) quand l'un des arcs change d'orientation (la droite hyperbolique passe par le centre du disque de Poincaré) est bien résolu par les deux. Il y a donc un vrai progrès dans la gestion interne des objets malgré une impossibilité générale théorique.

Le carcan du déterminisme

Au contraire de la question de la continuité des intersections, pour certaines situations, le déterminisme est un carcan dont on aimerait pouvoir sortir. L'exemple le plus simple est celui de l'enroulement de la droite sur le cercle. Les angles en géométrie dynamique sont traditionnellement mesurés entre 0 et 2π alors que l'on a souvent besoin d'enrouler sur plus d'un tour. Par exemple dans la construction suivante, les points A, B et C sont *a priori* sur la feuille principale de la pseudosphère (longitude dans $[-\pi, \pi[$). À gauche le point B s'apprête à franchir la frontière de la feuille ($+\pi$) mais *le déterminisme* fait que la figure se retrouve dans la position où B serait, à quelques pixels près, juste avant la même ligne frontière ($-\pi$).



Le déterminisme – ici des angles – ne permet pas de travailler en manipulation directe sur plus d'une feuille sur la pseudosphère

Nous verrons plus loin quelques spécificités de CaRMetal qui permettent – de deux manières différentes – de s'affranchir du déterminisme de façon choisie, contrôlée et mathématiquement significative. En particulier nous verrons que l'on peut résoudre, dans le contexte dynamique de la manipulation directe, cette question, dite de l'uniformisation du cercle.

III. Nouveau paradigme géométrique

Plus de vingt ans de recherches en didactique sur la géométrie dynamique ont réussi à faire que cet EIAH soit reconnu, par les institutions scolaires, comme un nouveau paradigme pour l'enseignement de la géométrie, désormais présent dans les curricula de l'enseignement secondaire, ainsi que dans les concours de recrutement des enseignants. C'est un résultat majeur de la didactique des mathématiques pour la géométrie.

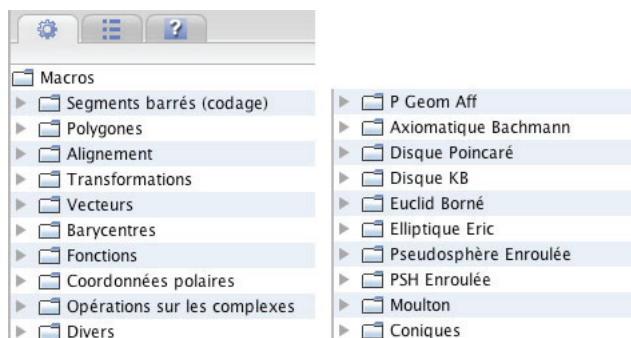
Cette partie se propose d'illustrer que ces possibilités – soit de présentation par l'enseignant, soit d'exploration dynamique par les élèves, se transposent dans l'enseignement supérieur. Avec sa capacité à construire des micromondes, la géométrie dynamique est d'une grande efficacité pour l'exploration de géométries spécifiques, peu étudiées, ne se prêtant pas à l'investigation classique, ou encore pour aller plus loin dans l'exploration de modèles mieux connus, mais pour lesquels on n'avait pas encore d'accès dynamiques en manipulation directe. Nous allons développer brièvement cela sur deux exemples.

1. Micromondes et macro-constructions

Nous avons déjà signalé la version forte de l'acceptation du concept de micromonde reconnue et développée par l'équipe de Cabri : la question des macro-constructions, disponibles maintenant sur tous les logiciels majeurs de géométrie dynamique, et de leur structuration. Nous avons vu que la structuration de celles de Cabri est la plus aboutie. Pour ce qui est de CaRMetal, en terme d'analyse de cohérence interne d'un EIAH, on notera une différence de traitement entre les outils standards et les macros, qu'a su dépasser Cabri II avec sa barre de menu *modifiable en manipulation directe* et enregistrable.

Néanmoins d'autres critères sont possibles. Dans la mise en place de la barre de menu de 2001 sur la pseudosphère, il avait fallu rivaliser d'ingéniosité dans l'organisation de toutes les constructions pour faire que l'ensemble de l'environnement soit entièrement inclus dans la donnée de la conique équateur afin de n'avoir à fournir, comme *unique donnée environnementale* sur la surface pseudosphérique, que cette information supplémentaire.

CaRMetal n'a pas de barre de menu iconique ni de palette personnalisable (pas encore). Toutefois ses macros sont très organisées et hiérarchisées. Elles sont du coup toutes accessibles à la fois, sans exclusive¹⁹. Ci-dessous on voit à gauche les macros standards livrées avec le logiciel (les dossiers contiennent parfois de nombreux sous dossiers) et à droite des dossiers de macros ajoutées, construites par l'utilisateur.



¹⁹ Avec les barres de menu de Cabri, il faut se constituer plusieurs barres, mais aussi conserver les macros indépendamment pour pouvoir assembler les barres quand on avait besoin de travailler par exemple par conjugaison dans deux modèles hyperboliques. Dans un contexte de production de figures, la démarche pouvait devenir chronophage même si le résultat est finalement très pertinent.

Ces macros ont surtout une propriété essentielle par rapport à ce qui pouvait se faire en 2001, celle d'avoir des *objets implicites*. Ainsi pour envoyer un point du disque de Klein Beltrami (KB) vers la pseudosphère, il y a plus aucun objet préalable à fournir pour décrire l'environnement, on ne clique que sur le point que l'on veut envoyer.

Le sentiment d'imprégnation dans le micromonde géométrique est autrement plus profond, on n'a plus à désigner les considérations techniques qui permettaient cette modélisation, nous sommes directement dans le modèle. Mais surtout, il n'a plus à concevoir une préparation des figures pour minimiser au maximum la description de l'environnement. Depuis Cabri a aussi intégré les macros implicites.

Ces précisions étant faite sur les micromondes, leur degré de réalisation, leurs évolutions, voici deux exemples abordés succinctement dans des contextes non euclidiens. Le premier a pour objectif – s'il devait être utilisé en formation (ce qui n'a pas encore été possible) de poser une réflexion à la fois sur les choix axiomatiques et sur les représentations que nous pouvons avoir nous-mêmes, par exemple de l'orthogonalité et de ses propriétés. Le second est une construction plus classique d'une figure sur la pseudosphère ayant des contraintes particulières.

2. La géométrie non arguésienne – L'exemple du plan de Moulton

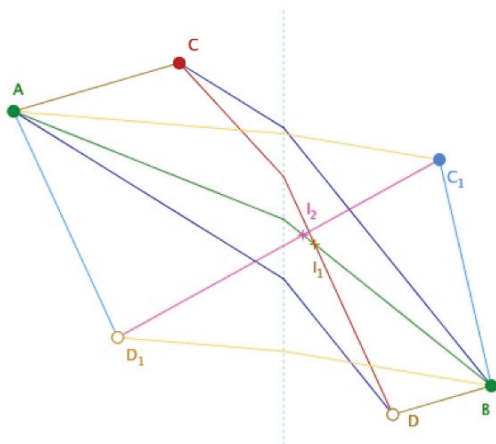
Dans les premières éditions de son ouvrage *Les Fondements de la géométrie*, Hilbert (1971, p. 126) a proposé un exemple de géométrie non arguésienne, à l'intérieur d'une ellipse, qui respectait tous ses axiomes sauf celui sur la relation entre la congruence des angles et des segments (en fait un cas d'isométrie des triangles). Il s'agissait d'illustrer que sans cet axiome on ne peut démontrer le théorème de Desargues dans le plan. Hilbert a montré que dans cette géométrie, le théorème de Desargues n'est pas vérifié.

Ensuite, en 1904, Moulton a proposé un modèle euclidien d'une géométrie non arguésienne beaucoup plus simple. On se place dans un repère ortho-normé du plan réel. Les points de sa géométrie sont les points du plan. La droite (AB) est, selon les positions de A et B :

- la droite euclidienne (AB) si celle-ci est parallèle à l'axe des ordonnées ou de pente positive ou nulle ;
- les droites affines par morceaux quand la pente (euclidienne) est négative avec une brisure sur l'axe des ordonnées, la pente étant le double dans le demi plan ($x > 0$) que celle dans le demi plan ($x < 0$).

L'intérêt de construire un micromonde²⁰ – avec des objets de cette géométrie manipulables comme ceux du logiciel, en particulier en pouvant traiter leurs intersections – est d'aller très vite dans l'exploration des propriétés de cette géométrie.

Le plan de Moulton est non arguésien, et en conséquence, même s'il existe des parallèles, avec l'axiome d'Euclide des parallèles vérifié, la géométrie n'a aucune propriété affine usuelle.



Les diagonales d'un parallélogramme ne se coupent pas en leurs milieux

Ci-dessus, à partir d'un segment $[AB]$ et d'un point C , on construit le parallélogramme $ACBD$ et l'intersection I_1 des diagonales. On recommence avec un point C_1 , le parallélogramme AC_1BD_1 et I_2 l'intersection des diagonales. Il n'y a pas concordance des intersections I_1 et I_2 . Les diagonales d'un parallélogramme ne se coupent pas en leurs milieux. Il n'y a pas de barycentre, ni de géométrie des coordonnées.

Le plan de Moulton étant localement euclidien, sauf aux points de l'axe des ordonnées, on peut aussi travailler sur l'orthogonalité. La perpendiculaire à une droite donnée en un point donné est nettement plus délicate à construire. Il y a huit cas possibles à inclure dans un seul objet. De plus, si généralement cette droite existe, il peut dans certains cas ne pas y en avoir et dans d'autres y en avoir deux.

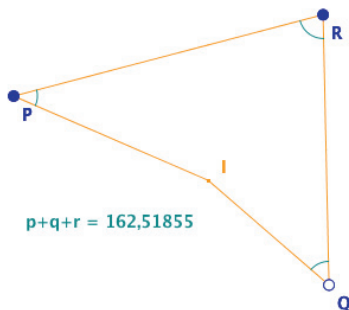
²⁰ L'auteur de ce texte a découvert CaRMetal autour de ce projet, non réalisable ni avec Cabri, ni avec Geogebra, qui de toute façon, à l'époque n'avait pas de macros.

En fournissant les macros du micromonde à des étudiants – et plus particulièrement à de futurs enseignants – il est facile de leur faire explorer un type de géométrie loin des standards non euclidiens, où la somme des angles peut être parfois supérieure à 2π , et parfois inférieure. On peut faire calculer les extrema de la somme des angles d'un triangle, faire réaliser ces extrema.

Manipulations

On peut déplacer les points P, Q et R. Le point P est légèrement attiré par une droite dont la pente réalise les extrema pour la somme des angles.

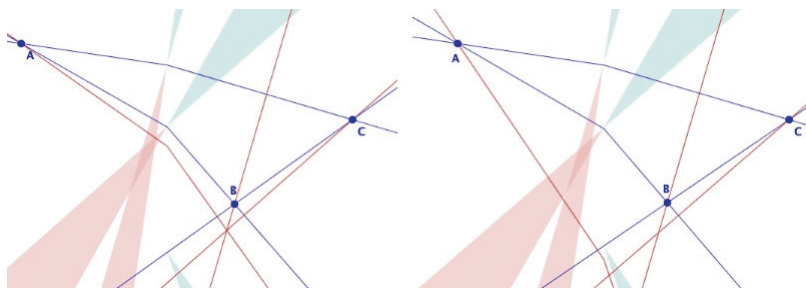
On peut donc facilement placer le triangle dans une configuration de somme d'angles extrême.
On peut aussi chercher à montrer le résultat annoncé.



Somme minimale : $2\pi - \tan^{-1}\left(\frac{\sqrt{2}}{4}\right)$ soit 160.52878°

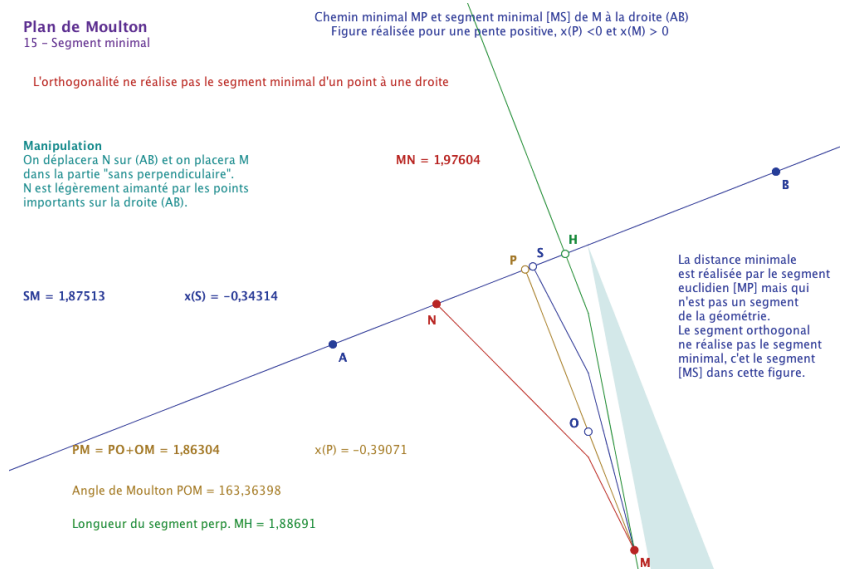
Somme maximale : $2\pi + \tan^{-1}\left(\frac{\sqrt{2}}{4}\right)$ soit 199.47122°

On peut aussi proposer d'explorer cette géométrie sur des configurations usuelles et inviter à chercher des conditions suffisantes pour retrouver des propriétés classiques, ici la question de l'orthocentre d'un triangle :



Pour chaque droite, en vert la partie du plan d'où on ne peut mener aucune perpendiculaire à la droite, en rose celle dans laquelle par chaque point il a deux perpendiculaires. À gauche : cas où il y a orthocentre (hauteur à pente négative issue d'un sommet à deux côtés de pente négative... à condition (à droite) que les intersections des hauteurs soient bien dans le même demi plan par rapport à l'axe des ordonnées. Entre les deux figures, B n'est déplacé que de quelques pixels pour faire basculer la hauteur issue de A, d'un côté de la zone verte pour (BC), à l'autre côté.

L'enjeu d'une telle pratique pour des futurs enseignants peut être la réactivation de la curiosité, et de la précision, sur les propriétés de la géométrie élémentaire comme elle est enseignée en collège et la prise de distance vis-à-vis des réflexes mathématiques quand on n'a jamais rencontré de contre-exemple géométrique, manipulable aussi simplement. Par exemple dans cette géométrie, la distance d'un point à une droite n'est pas réalisée par le segment orthogonal :



3. Un autre exemple sur la pseudosphère de Beltrami

Quand nous avons donné l'exemple du micromonde construit autour de la pseudosphère, celui-ci était *intrinsèque*, c'est-à-dire entièrement construit sur la pseudosphère, avec les formules pour les géodésiques, les perpendiculaires des médiatrices, et même les cercles. Dans l'exemple qui suit nous allons utiliser la projection de Beltrami de la pseudosphère dans ce qu'il avait appelé le cercle limite, qui sera très rapidement repéré par Klein comme un modèle complet du plan hyperbolique : KB, le disque de Klein Beltrami. On se propose cette fois de « donner vie » à une phrase un peu malheureuse de Coxeter, c'est-à-dire construire une situation correspondant réellement à son interprétation des droites de la pseudosphère. En effet, au détour d'une phrase, Coxeter laisse à penser qu'elles s'enroulent à l'infini sur la surface

alors qu'on sait qu'elles sont bornées en altitude, qu'elles partent de l'équateur pour y revenir : « Every geodesic that is not merely a meridian winds itself round the 'horn' as it proceeds in one direction whereas in the opposite direction it is abruptly cut off by the cuspidal edge » (Coxeter 1989, p. 378).

On se propose donc de construire une courbe ayant un sens géométrique – si possible élémentaire – qui s'enroule à l'infini sur la surface depuis l'équateur. La pseudosphère est aussi appelée, selon une terminologie initiée par Klein, surface pseudosphérique *parabolique*, car la partie du plan hyperbolique qu'elle représente ne peut approcher qu'un seul point à l'infini du plan hyperbolique (comme la parabole dans le plan projectif).

Ce point s'atteint soit selon tout méridien quand l'altitude augmente (ce qui peut avoir un rapport avec la représentation de Coxeter), soit encore en s'enroulant à l'infini sur un cercle parallèle à l'équateur. En effet, les cercles parallèles à l'équateur de la pseudosphère (en terme euclidien, avec le plongement de la surface dans l'espace de dimension 3) sont des *horicycles*²¹ pour la géométrie de la surface.

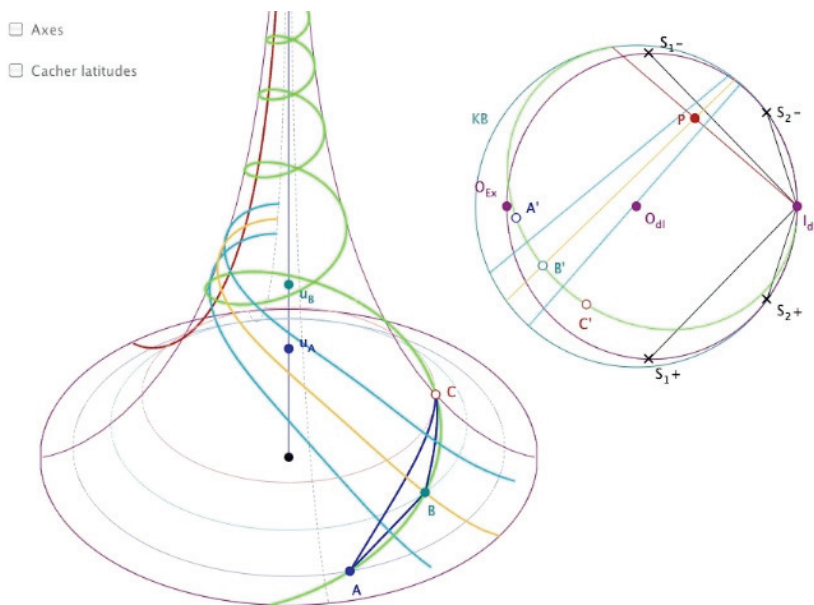
Tous les méridiens de la pseudosphère sont des droites en faisceau « sans support » (*i. e.* des droites parallèles au sens de la note précédente) de centre l'unique point idéal de la pseudosphère. Il en résulte que si on construit un triangle ayant trois points sur une équidistante dont un méridien donné est l'axe, cette équidistante aura le point idéal de la pseudosphère comme point limite à l'infini et donc s'enroulera à l'infini sur la surface.

Voyons maintenant comment construire cette équidistante concrètement. Nous allons cette fois utiliser la conjugaison avec le disque de Klein Beltrami (KB). Dans KB, l'image de la pseudosphère (PS) est l'intérieur strict de l'ellipse de centre O_{dl} et d'axe $[O_{Ex}I_{dl}]$. Le point I_{dl} est le seul point à l'infini accessible à la pseudosphère. Sur PS on se donne deux points A et B (mani-

²¹ Il y a trois types de faisceaux de droites hyperboliques : les faisceaux à centre (droites concourantes en ce point), les faisceaux à axes (droites ayant cet axe comme perpendiculaire commune) et les faisceaux « sans support » (ou « à point idéal »), c'est à dire les droites parallèles, se coupant en un point idéal du plan hyperbolique. Hilbert, comme Bolyai parlait de « bouts » pour le point idéal. Alors si on appelle *cycle* l'image d'un point par toutes les symétries d'un même faisceau, il y a trois types de cycles : les *cercles* – dont le centre est le centre du faisceau – les *équidistantes* – dont l'axe est l'axe du faisceau, et les *horicycles* – pour lequel le point idéal du faisceau est parfois aussi appelé « centre ». Les équidistantes ne sont pas des droites en géométrie hyperbolique. Les droites remarquables d'un triangles (hauteurs, médianes, médiatrices) sont en faisceau.

pulables en longitude, et en altitude par u_A et u_B). Ils sont envoyés sur KB. On construit la médiatrice (bleue) de leurs images, puis le méridien rouge orthogonal à cette médiatrice passant par I_{dl} .

D'un point P de ce méridien est issue une perpendiculaire (orange). C'est le symétrique de A' par rapport à cette droite. Le triangle A'B'C' ayant deux médiatrices qui ont – par construction – une perpendiculaire commune (le méridien rouge), les trois médiatrices du triangle – dont on sait qu'elles sont toujours en faisceau – sont nécessairement en faisceau à axe.



Le rêve de Coxeter

Il en résulte que A', B' et C' sont trois points d'une équidistante d'axe un méridien, et dont un des deux points limites est l'unique point idéal I_{dl} pour la pseudosphère. Renvoyons cette construction sur la surface²². Le point C est donc un point construit à partir de A et B pour que le triangle ABC soit sur une équidistante d'axe le méridien rouge (de la PS) et s'enroulant sur la sur-

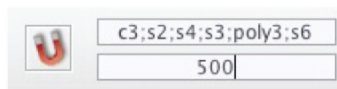
²² Dans l'illustration précédente, les médiatrices ne sont tracées que sur la feuille principale pour plus de lisibilité de l'orthogonalité avec le méridien rouge.

face à l'infini. Dans cette construction, cette figure est directement manipulable par les points u_A , u_B , A et B.

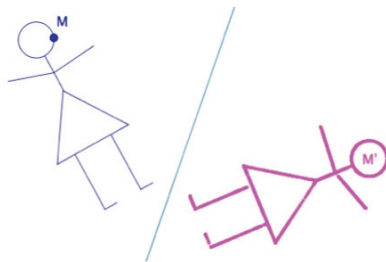
Ces deux exemples veulent seulement illustrer que, moyennant un travail préalable de construction des outils²³ des micromondes que l'on veut explorer, la géométrie dynamique peut devenir – avec d'autres EIAH – un nouveau paradigme pour l'enseignement de la géométrie, même dans le supérieur.

IV. Aimantation et réalité augmentée

Les points libres – et uniquement eux – peuvent être aimantés par d'autres objets (points, segments, droites, cercles, polygones, etc.) avec un rayon d'aimantation commun ou propre à chaque objet. On clique simplement sur les objets qui vont attirer le point sélectionné, puis on donne un rayon.



Avec un rayon important, en pratique tout se passe comme si le point était sur la figure constituée de la réunion des objets attirants. Ci-dessous, l'image par une symétrie orthogonale d'un point M aimanté.



Dans cette utilisation, l'aimantation est une *extension topologique* de la notion de point sur objet en *point sur figure*.

1. Premières utilisations didactiques : du continu au discret

Déjà dans ce simple contexte, pourtant élémentaire, une utilisation de l'aimantation permet de mettre en place des ingénieries nouvelles pour la

²³ Très beau challenge pour l'enseignant concerné par la géométrie dynamique.

géométrie des configurations. Afin de dégager plus clairement l'importance de cette utilisation, revenons un instant sur l'un des sens que l'on donne à la géométrie dynamique.

On a régulièrement dit, depuis longtemps, au début dans un discours critique avant d'en reconnaître l'intérêt, que la géométrie dynamique est une géométrie qui valide les propriétés par résistance au mouvement – *i. e.* à la manipulation directe. Une propriété (concours des médianes, des hauteurs d'un triangle) est découverte – et conjecturée – en manipulant une figure, en tant qu'invariant, un peu comme des lois physiques. La géométrie dynamique place l'élève dans un contexte scientifique comme observateur et expérimentateur sur le monde géométrique qu'il a à étudier. La critique portait non seulement sur l'utilisation du mouvement en mathématique mais surtout sur la pertinence de l'observation dans un monde que l'on veut *a priori* présenter et enseigner comme hypothético-déductif²⁴.

Or, bien sûr il n'y a pas de mouvement dans une figure, même quand l'élève déplace un sommet du triangle à l'écran. Il n'y a que des instances des points constituants – et donc des instances de la figure. Il n'y a que des réalisations successives de dessins d'une seule figure géométrique. Il n'y a pas de mouvement, car il n'y a pas de temps dans les figures. Comme construction d'objets ayant des relations mathématiques entre eux, elles sont atemporelles. Le temps est du côté de l'utilisateur quand il déclare qu'il « déplace les sommets depuis 20 s et les droites se coupent tout le temps²⁵ » c'est-à-dire dans toutes les instances observées : c'est l'utilisateur humain qui est plongé dans le temps pas la figure géométrique. Et c'est cette rencontre entre la temporalité de l'utilisateur et l'atemporalité de la figure mathématique dans un contexte d'instanciation facilitée qui fait de la géométrie dynamique un nouveau paradigme pour aborder la géométrie.

²⁴ Le premier point est simplement réfutable en précisant que les mathématiciens, de tout temps, ont toujours expérimenté même sur les nombres ou les structures. La critique s'est alors portée sur la question des *premières représentations* que l'on donne de la démarche hypothético-déductive puis peu à peu s'est éteinte devant la richesse du concept de géométrie dynamique et ses propres arguments de *renforcement de la nécessité* de la démarche finale de preuve.

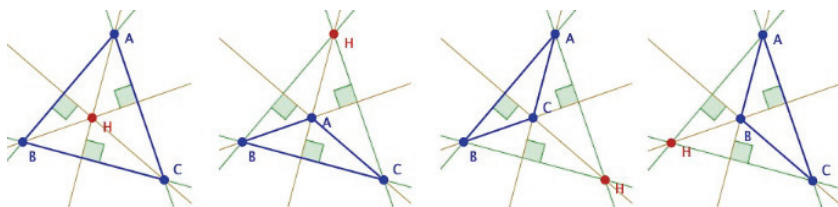
²⁵ L'élève rajoute même parfois « au même point » pour rendre compte de la concurrence des droites alors que le point de concours se déplace à l'écran avec le mouvement de la souris. Ce raccourci linguistique, dénote (ou participe à la construction au moment où il est prononcé) un début de conceptualisation qu'il y a là un invariant et donc une propriété particulière.

Cette description du paradigme de la géométrie dynamique – comme relation entre le mouvement des objets à l'écran par la manipulation de l'utilisateur et l'atemporalité des figures – renforce de fait une représentation de la géométrie dynamique déjà implicitement plongée dans un contexte continu : l'invocation du « mouvement » des figures, du temps de l'utilisateur s'ajoute à la problématique de la déformation barycentrique, du suivi continu des intersections. Même si personne n'oublie que le nombre de pixels à l'écran est fini, *le paradigme de la géométrie dynamique est habillé de continuité*.

Manipulation directe discrète sur configuration finie

Nous allons voir qu'une approche particulière, mais importante, de l'aimantation va être d'étendre cet habillage implicite au discret. Par l'aimantation, les élèves vont pouvoir expérimenter – puis raisonner – sur des configurations finies, a priori fixes, avec la même richesse de manipulation que le permet la géométrie dynamique. C'est pour cela que nous disons que *l'aimantation étend la pertinence de la géométrie dynamique aux configurations finies*. Voyons cela sur un exemple.

On s'intéresse à la propriété du quadrilatère orthocentrique : dès que H est orthocentre de ABC , alors tout point A , B C (ou H) est orthocentre du triangle formé par les trois autres points. Ce théorème est généralement travaillé non pas pour son résultat, mais pour mettre en œuvre le changement de point de vue sur un même dessin vis-à-vis des côtés et des hauteurs.



Dans la figure précédente, un point H est aimanté par les trois sommets d'un triangle et son orthocentre, il ne peut prendre d'autres positions. Ensuite un jeu sur les noms des points permet de rendre le visuel ci-dessus. Dans cette manipulation, on a choisi de faire que H soit toujours l'orthocentre de ABC , chaque point semblant se déplacer en manipulant « à la main » (en manipulation directe) le point H sur l'un des quatre points. On rend ainsi visible la propriété qu'on veut montrer, par exemple que (HC) est toujours perpendiculaire à (AB) et donc que chaque point est orthocentre des trois autres. On transforme ainsi *une configuration finie statique en une configura-*

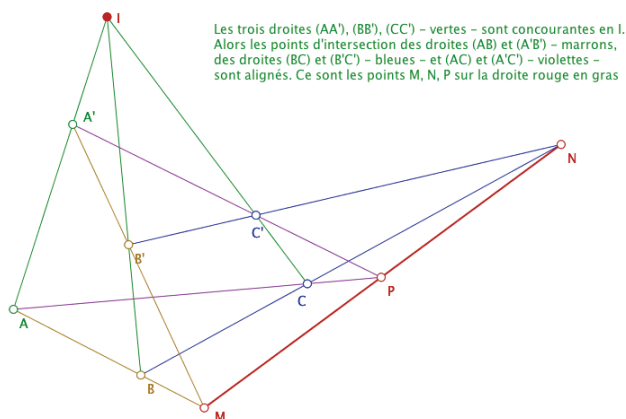
ration finie dynamique avec une manipulation directe discrète limitée à ce que l'on veut étudier.

Un autre exemple, identique mais dans un contexte plus complexe, aura cette fois, explicitement, pour but de montrer l'intérêt d'apprendre (ou de connaître) une configuration plutôt qu'un théorème :

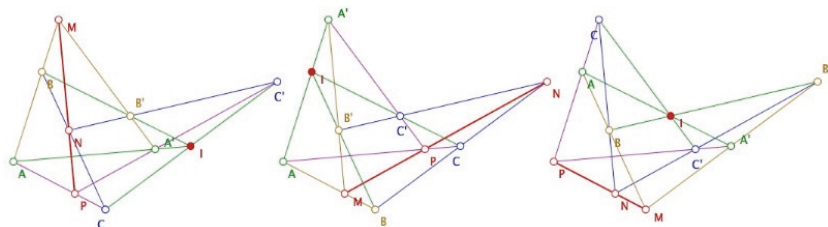
Aimantation multiple

3. Aimantation sur des points (CAPES)

Configuration de Desargues



Cette configuration est symétrique, elle comporte 10 points et 10 droites, chaque droite contient trois points et par chaque point passent trois droites. Il en résulte que chaque point de la configuration peut être le point I de l'hypothèse initiale et chaque droite, la droite des points alignés M, N, P. Pour montrer qu'il est préférable de connaître la configuration plus que l'énoncé formel du théorème, on peut proposer une figure, où, comme la précédente, I est aimanté par les points de la configuration et quand on déplace I, tous les points se déplacent sur la configuration ainsi que la droite des points alignés M, N, P :



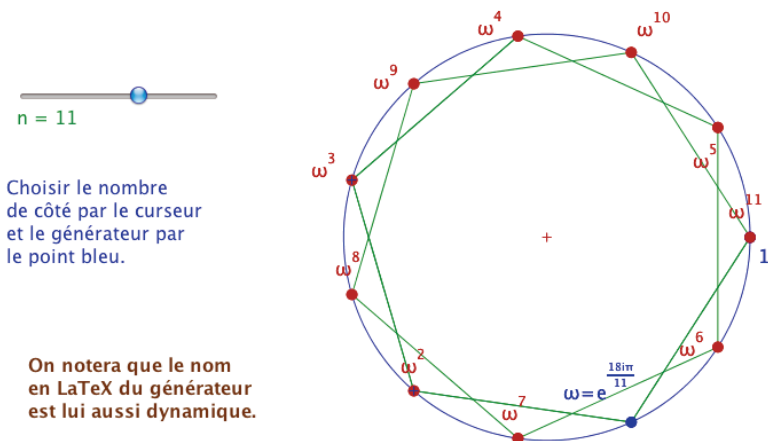
Dans les deux cas l'effet, en situation, est saisissant en terme cognitif, comme si un saut avait été franchi dans le regard des configurations finies. C'est la raison pour laquelle nous y percevons une extension du paradigme de la géométrie dynamique aux configurations finies, par aimantation.

L'utilisation de l'aimantation dans un contexte fini peut avoir une utilisation plus technique. Par exemple pour rapidement émettre des conjectures sur les sous groupes des groupes des racines n -ièmes de l'unité, on peut travailler dans ce contexte :

Aimantation multiple

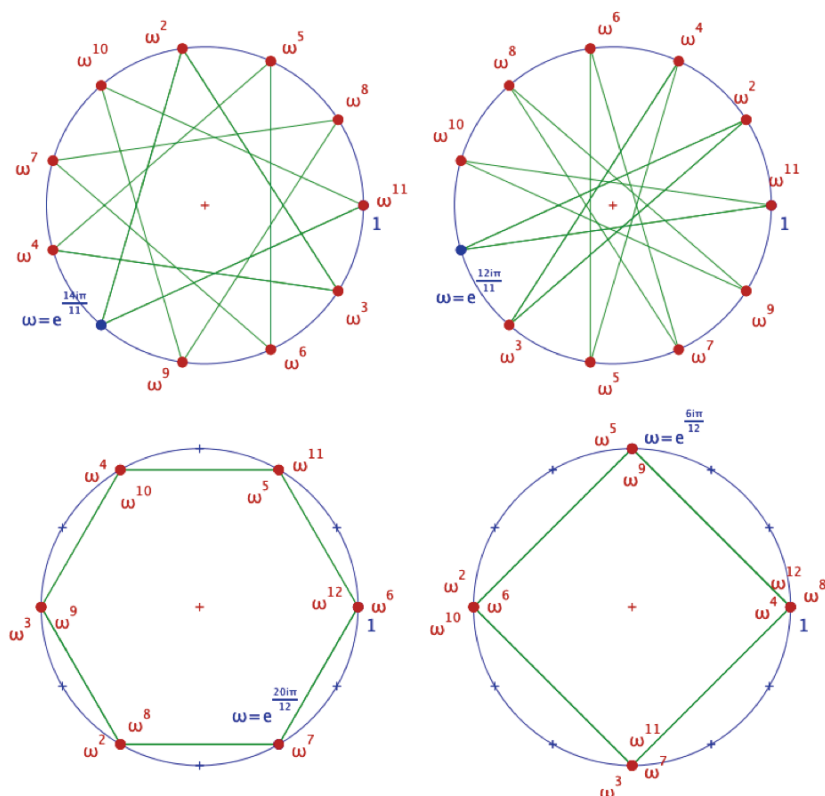
5. Aimantation sur des points (CAPES)

Utilisation en Algèbre
Polygones réguliers étoilés
et racines de l'unité



Là encore le point ω est aimanté sur les racines de l'unité, en fonction de la valeur de n produite par le curseur. La présence du curseur et le nom des points donnent une certaine technicité à la figure, qui n'a pas l'épure à la fois esthétique et conceptuelle des deux précédentes.

Il est intéressant d'observer que, même si elle traite elle aussi de configurations finies, le rapport au discret ne s'installe pas de la même façon. Les étudiants restent plus « plongés dans le plan complexe » que sur une configuration finie. Une raison possible est que cette figure – contrairement aux deux configurations précédentes – est là pour apporter plus qu'elle-même, elle est constamment interprétée en terme de structure algébrique, de générateurs de sous-groupe.

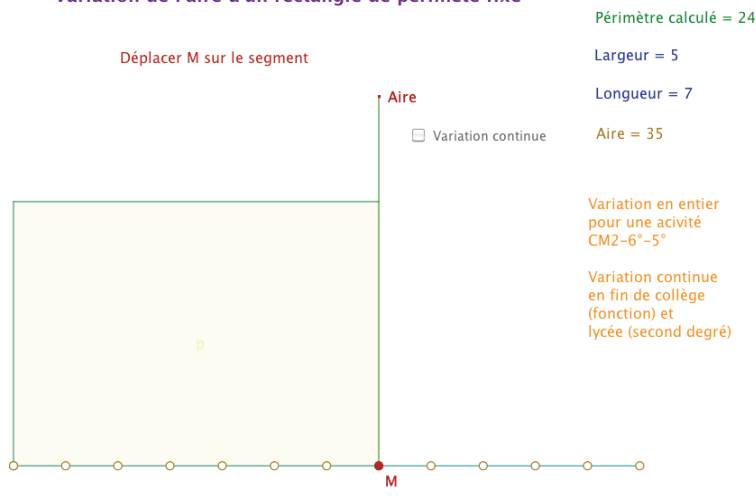


En haut à gauche un polygone étoilé pour $n = 11$, puis trois cas pour $n = 12$

Mélanger le continu et le discret

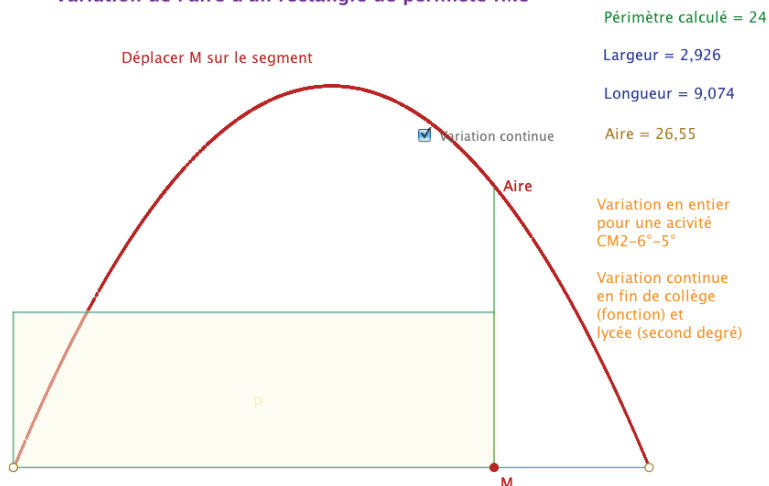
Un même problème peut avoir des approches à différents niveaux de la scolarité, dans un contexte continu ou dans un contexte discret. On s'intéresse ici à un problème qui est soulevé en classe de CM2 quand on travaille sur la non relation variationnelle entre l'aire et le périmètre. Il s'agit de mettre en défaut la représentation – très installée – que plus le périmètre est grand (plus on a de ficelle pour entourer) plus l'aire (entourée) est grande. On travaille par exemple avec des rectangles de périmètre constant, on demande d'en construire – sur du papier quadrillé – avec des mesures entières pour constater que les aires sont différentes et même, éventuellement se questionner sur l'aire maximale rencontrée.

Variation de l'aire d'un rectangle de périmètre fixe



Voici une figure qui permet de travailler en entier – éventuellement en CM2 on peut enlever le segment avec l'aire et ne s'intéresser qu'aux valeurs numériques et à l'illustration – mais aussi de manière continue pour d'autres classes. La figure a été faite à des fins de formation pour illustrer la continuité des thèmes dans l'enseignement en formation continue.

Variation de l'aire d'un rectangle de périmètre fixe



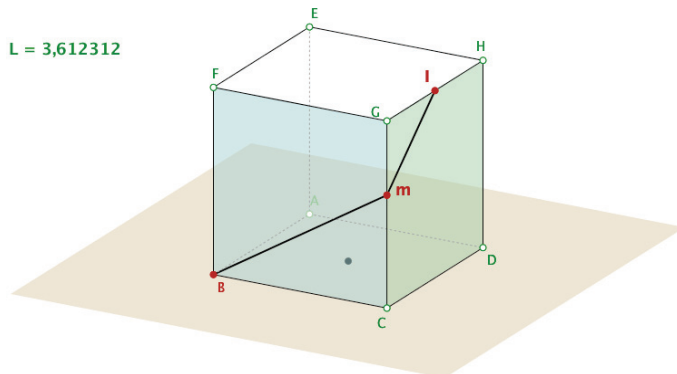
Aimantation et réalité mathématique augmentée

Dans le contexte d'une illustration ou d'une simulation, la manipulation directe présente le défaut – didactiquement important – de généralement ne pas pouvoir passer par certains points significatifs de la figure puisque ceux-ci ne rentrent pas dans le réseau des pixels *exactement* accessibles à la souris.

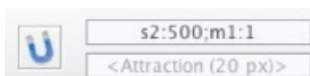
Ainsi si un segment est simplement partagé en trois, par le théorème de Thalès, sauf configuration très particulière, il n'y a aucune chance qu'un point sur objet d'un segment manipulé à la souris puisse passer par un point au tiers d'un segment. L'aimantation va permettre d'enrichir la manipulation directe et augmenter la pertinence mathématique des figures d'où cette expression de *réalité mathématique augmentée*.

Dans l'exemple suivant, on veut aller de B à I en passant par l'arête [CG] par un chemin minimum. Il est facile de voir que le chemin est minimum quand le point m est au tiers du segment [CG], point inaccessible à la manipulation directe.

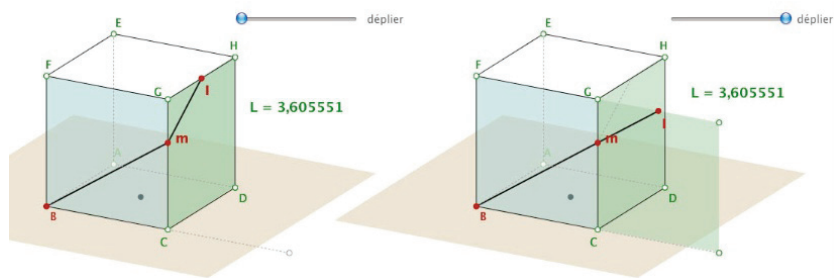
Aimantation et réalité mathématique augmentée
6. Par aimantation multiple pondérée



Comme les points sur objet ne peuvent être aimantés, on aimante m à la fois fortement sur l'arête, et à un pixel sur le point solution. Dans ce cas le point m paraît passer naturellement sur le point solution puisque physiquement il y passe à moins de 1 pixel, l'aimantation ne se voit pas, on conserve l'impression de continuité tout en enrichissant la manipulation directe.



Dans le cas de cette figure de démonstration²⁶, on a choisi une illustration un peu ostentatoire (un curseur apparaît quand on passe sur le point solution) pour montrer en même temps qu'avec la gestion conditionnelle, on peut faire apparaître un curseur jusque-là caché, qui conduit à une autre possibilité d'action, elle-même conduisant à la perception claire de la solution :



Nous utiliserons ultérieurement cette notion de réalité mathématique augmentée dans un contexte plus riche.

2. Aimantation algébrique et déterminisme

L'aimantation peut non seulement être pondérée individuellement, comme on l'a vu dans le paragraphe précédent, mais elle peut aussi être pondérée de manière algébrique en fonction de la figure. C'est déjà ce qui a été dans la figure sur l'aire et le périmètre, pour le choix du type de la variable discret/continu. Le point M de cette figure est aimanté de la façon suivante (où a est le nom de la variable associée à la case à cocher) :

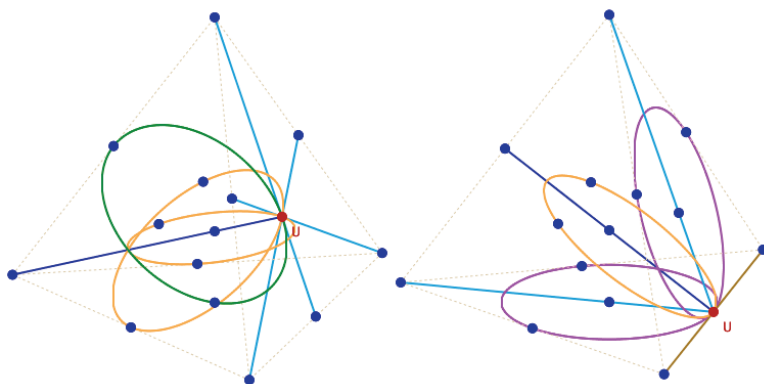
$I0:500*(a==0);I1:500*(a==0);I2:500*(a==0);I3:500*(a==0);I4:500*(a==0);I5:500*(a==0);I6:500*(a==0);I7:500*(a==0);I8:500*(a==0);I9:500*(a==0);I10:500*(a==0);I11:500*(a==0);I12:500*(a==0);s4:500*(a==1)$

Ce qui signifie simplement M est aimanté par les 13 points à abscisses entières I_0 à I_{12} si la case *Continu* n'est pas cochée et par le segment si elle est cochée. Cela signifie aussi que M n'est pas un point sur objet et donc on ne peut pas faire un lieu de l'aire quand M varie, d'où la *trace* sur l'illustration.

On voit tout de suite que cette situation est autrement plus riche de possibilités que la simple pondération numérique. Voici un exemple moins trivial de cette situation en géométrie finie de dimension 3. On s'intéresse aux droites du plus petit espace projectif sur le corps à deux éléments, $PG(3, 2)$. Il

²⁶ Réalisée par Monique Gironce de l'IREM de Toulouse.

a 15 points et 35 droites de trois points chacune. Une représentation très esthétique de cet espace est le tétraèdre régulier avec comme points de l'espace, les sommets, milieu des arêtes et des faces, et l'orthocentre du tétraèdre. Voici les 15 points et 14 des 35 droites :



On s'intéresse plus précisément aux partitions de l'espace en cinq droites disjointes. Il y a alors 56 partitions possibles que l'on répartit traditionnellement en cinq types de configurations (Polster, 1998) :

Les 56 partitions de l'espace projectif $PG(3,2)$ en droites (spreads)

Afficher

Le tétraèdre de base

Précisions

Manipulations de la figure

☐ en trait plein ☐ Montrer les noms

Configuration de type 5 (24 partitions)

Un sommet est sur une hauteur (bleu clair, manipulée par le point U).
 Deux autres sommets sont sur une arête (manipulée par V : 3 possibilités pour une hauteur donnée).
 Le dernier sommet est sur une arête de face : deux positions possibles manipulables par W.
 Note : ces constructions fonctionnent car on montre que 3 droites deux à deux disjointes engendrent une unique partition.
 On peut facilement observer que deux droites disjointes engendrent toujours deux partitions distinctes.

Deux points U et V et un autre point W dans la configuration 5 permettent d'explorer toutes les partitions de l'espace en 5 droites.
 Le point V a un déplacement contraint à la fois par la configuration et la position du point U.
 Ses mouvements sont décrits en commentaire de chaque configuration.
 De même le déplacement de W est fonction des positions de U et V.

Pour U donné on peut déplacer V.
 Si on déplace U, cela peut modifier la position du point V en fonction de la configuration.

☐ Cacher "Précisions"

Quand on change de configuration il faut déplacer un peu le tétraèdre pour que la nouvelle situation s'actualise

Utiliser le clic-droit glisser pour manipuler le tétraèdre et la molette de la souris pour régler sa taille.

Sans entrer dans le détail des constructions de cette figure, voyons les possibilités d'aimantation algébrique. Les types des configurations sont indiqués par la variable *sp*. L'aimantation du point U est simplement régie par les configurations, avec une aimantation de 300 sur certains points selon les configurations :

$P:300*(sp==5||sp==2);Q:300*(sp==5||sp==2);R:300*(sp==5||sp==2);S:300*(sp==5||sp==2);L:300*(sp==1||sp==3);M:300*(sp==1||sp==3);N:300*(sp==1||sp==3);I:300*(sp==3||sp==4);J:300*(sp==3||sp==4);K:300*(sp==3||sp==4)$

Pour la configuration de type 5, la plus complexe, l'aimantation du troisième point, W, est :

$I:WatI;J:WatJ;K:WatK;L:WatL;M:WatM;N:WatN;H:300*(sp==1||sp==2||sp==3||sp==4)$

L'attraction en H signifie que W n'étant utilisé que dans le cas de la configuration de type 5, on a choisi de le ranger sur H dans les autres cas.

Les *WatXX* sont des expressions qui décrivent à quelles conditions W peut être en XX. Ainsi *WatI* a pour expression :

$300*(sp==5)*((y(U)==y(P))* (y(V)==y(L)||y(V)==y(N))+(y(U)==y(R))* (y(V)==y(J)||y(V)==y(K)))$

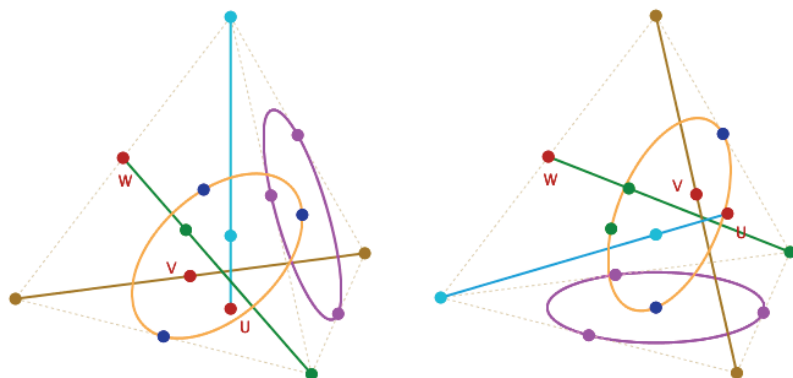
Ce qui est la traduction algébrique de la situation suivante :

- être dans la configuration de type 5 et
- ((U est en P) et (V en L ou N)) ou ((U est en R) et (V en J ou K)).

Rupture du déterminisme

Voyons maintenant ce qu'il en est de la manipulation. Depuis la situation de l'illustration précédente, on glisse U au centre de la face « du bas » (à gauche), puis on remet U à sa position initiale (à droite).

On constate (figure à la page suivante) que V revient à sa position initiale mais pas W. Ainsi l'aimantation algébrique s'affranchit du déterminisme. La raison en est très simple : quand U revient à sa position initiale, comme la position de W déjà prise est aussi l'une des deux solutions possibles dans la position initiale, quand U revient à cette position, W étant déjà dans une situation solution, il y reste.



Avec cet exemple, nous avons voulu montrer les possibilités fort riches de l'aimantation algébrique, mais il n'est pas nécessaire de se placer dans une situation aussi complexe pour briser le déterminisme du logiciel : on peut l'obtenir avec seulement quatre points. La raison est différente de la précédente, et très simple pour une situation bien moins contraignante : le logiciel privilégie la distance pour choisir quel point va attirer en premier.

Le point de vue de l'auteur

Interrogé sur cette situation et sur ce choix, l'auteur du logiciel répond : « il me faudrait 10 min pour introduire un balayage par ordre de saisie dans la liste des objets qui attirent un point, et ceci rétablirait le déterminisme sur l'aimantation algébrique, mais j'ai pensé que les personnes qui utilisent cette aimantation-là ont probablement besoin d'un peu de liberté géométrique et qu'il valait mieux leur laisser cette liberté ».

En fait l'aimantation algébrique reste encore peu utilisée même si elle est riche de potentialité comme on l'a vu ici. On notera que si l'aimantation pondérée est scriptable, l'aimantation algébrique ne l'est pas encore (dans la version 3.5) : la figure précédente²⁷, sur les 56 partitions des droites a été faite « manuellement ».

²⁷ Toutes les figures mentionnées dans cet article sont téléchargeables dans des classeurs CaRMetal. Liens en webographie.

V. Déterminisme enrichi par la récursivité

Dans cette partie nous abordons une autre spécificité de CaR, la récursivité naturelle des points et des expressions. Cette récursivité va nous permettre de nous affranchir du déterminisme et nous verrons comme, avec le magnétisme de CaRMetal, nous pouvons faire progresser l'engagement direct dans un comportement plus global, au niveau de la figure, et aller ainsi jusqu'à un *déterminisme enrichi*. Nous allons voir cela en plusieurs étapes.

Une façon à la fois rapide et élémentaire d'illustrer la récursivité sur les points est de réaliser *un dessin* comme limite d'un attracteur. C'est le cas du triangle de Sierpinsky, attracteur des trois homothéties de centre chacun des sommets et de rapport $\frac{1}{2}$. Autrement dit, étant donné un point P, on va construire, récursivement, ce point P par la relation ci-dessous, où *alea3* est un nombre aléatoire qui prend trois valeurs :

Référence circulaire

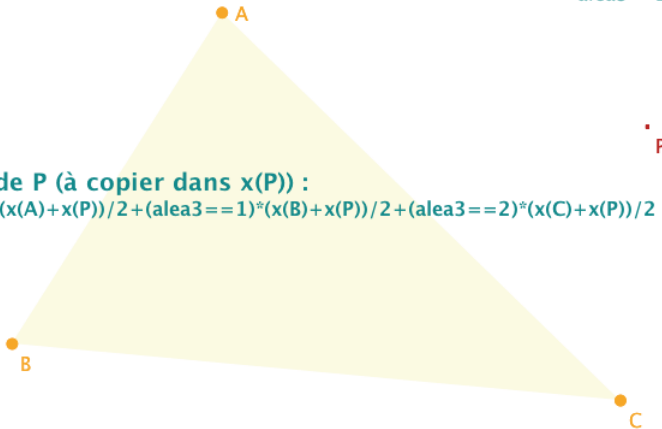
1. Points autoréférents
Récursivité élémentaire

Sierpinsky Coordonnées ▼

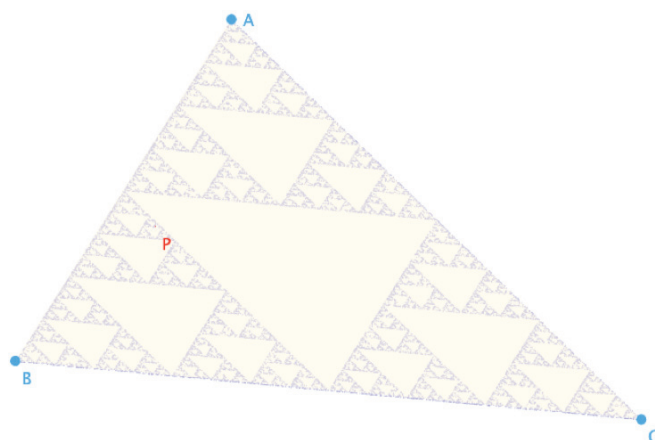
alea3 = 2

abscisse de P (à copier dans x(P)) :

$$(alea3=0)*(x(A)+x(P))/2 + (alea3=1)*(x(B)+x(P))/2 + (alea3=2)*(x(C)+x(P))/2$$

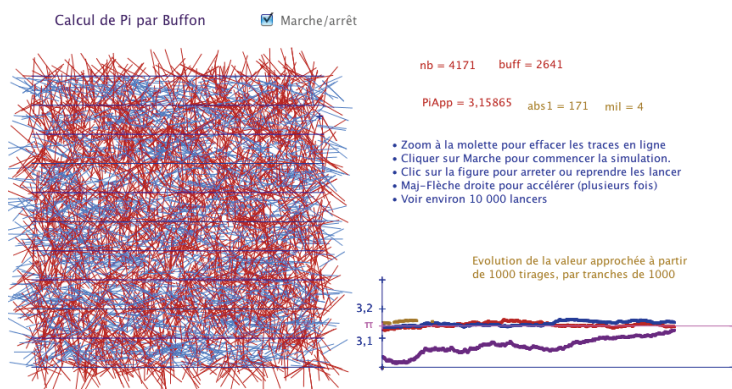


On voit que $x(P)$ est défini par $x(P)$. La récursivité ne boucle pas car le point est créé avant de lui donner ses coordonnées, *cette création est une initialisation*. On notera l'utilisation des booléens pour éviter les tests conditionnels imbriqués. On obtient la figure suivante avec *la trace* du seul point P. Comme c'est une trace, c'est un dessin, ce n'est pas dynamique.



Nous n'insisterons pas plus sur cette première utilisation. Elle est très efficace, en particulier on obtient rapidement l'illustration précédente, mais ce n'est pas dynamique.

La récursivité s'applique aussi aux expressions. Contrairement aux points, une expression n'est pas initialisée quand elle est posée sur la feuille de travail, il faut donc un processus pour l'initialiser soi-même. Dans la figure suivante (aiguilles de Buffon²⁸), toutes les variables sont récursives, l'expression récursive par excellence étant $n = n + 1$.



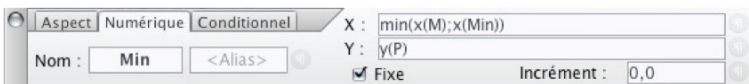
²⁸ Détails sur <http://www.reunion.iufm.fr/recherche/irem/spip.php?article98>

1. Récursivité et introduction du temps dans les figures

En introduisant la manipulation directe discrète, nous avons insisté sur l'atemporalité d'une figure dans un logiciel déterministe, et nous avons vu en quoi elle participait à la construction du paradigme de la géométrie dynamique. Nous avons ensuite brièvement signalé comment l'aimantation algébrique pouvait briser le déterminisme, mais généralement de manière furtive, car dans un environnement discret. On se propose cette fois de *s'affranchir* du déterminisme, c'est-à-dire de le rompre, tout en contrôlant le résultat pour un comportement mathématique recherché précis.

L'introduction du temps dans un contexte mathématique va relever du plongement d'une situation locale dans un contexte global. L'archétype de cette situation pourrait être l'incorporation d'une mémoire temporelle extérieure, comme celle du thermomètre de jardin avec des curseurs que le jardinier manipule à la main pour mettre à jour quotidiennement les extrema sur une semaine par exemple. Comme une figure géométrique, le thermomètre est toujours au présent. Mais il est plongé dans une globalité : le jardinier agit sur un curseur du thermomètre en y introduisant sa propre temporalité. Nous allons introduire nous aussi la temporalité de l'utilisateur dans une figure. Contrairement au thermomètre physique de notre jardinier²⁹, celui de CaRMetal, parce qu'il peut être récursif, va prendre en compte lui-même cet apport extérieur. Ainsi *l'histoire de la manipulation de la figure* par l'utilisateur va faire partie intégrante de la figure, sinon l'histoire, sa trace, et la partie mathématiquement significative de cette trace.

Poursuivons avec une modélisation du thermomètre : sur un axe horizontal, on place un point M. On se propose de construire un point *Min* qui garde la trace du minimum de l'abscisse du point M. Par essence même de la géométrie dynamique, cette trace est impossible à réaliser dans un contexte déterministe. Or dans un contexte récursif, c'est élémentaire à construire, il suffit de donner à ce point *Min* l'abscisse $\min(x(M), x(M))$ où *min* est la fonction minimum de CaRMetal.



En pratique on construit un point libre sur la figure, en lui donnant le nom *Min*. Cette construction *l'initialise*, et c'est essentiel. Ensuite on rentre ses

²⁹ Disons qu'il avait un vieux thermomètre, mais nous conservons cette analogie.

coordonnées, avec abscisse récursive. On peut alors explorer cette incorporation du jardinier dans le thermomètre :



À gauche la situation initiale, puis on déplace à la souris le point M vers la gauche, Min suit, puis on déplace M vers la droite, Min reste à la position minimale prise par M. Dans les deux dernières illustrations on est ensuite allé à gauche du point P, poignée de manipulation de la droite.

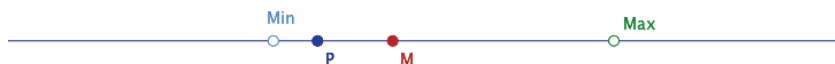
Le point *Min* ne revient pas à sa position initiale quand M, lui, y revient, et plus encore il fait bien ce pour quoi il a été construit : il garde la trace de la manipulation de la figure par l'utilisateur du côté du minimum de l'abscisse. Non seulement on a brisé le déterminisme du logiciel par incorporation du temps de l'utilisateur dans la figure, mais en faisant cela, on donne une valeur ajoutée mathématique à la figure. On résumera cette situation en disant que l'on s'*affranchit* du déterminisme.

On ferait de même avec un point *Max*. Et on voit bien qu'à force de déplacer *Min* à gauche et *Max* à droite, l'information va finir par ne plus être pertinente.



Comme le jardinier réinitialise ses curseurs *Min* et *Max* chaque semaine ou chaque mois, il faudrait, au sein de la figure, mettre en place un procédé qui permettrait, d'une façon ou d'une autre, de réinitialiser les positions des points *Min* et *Max*. CaRMetal dispose d'une fonction interne, *d*, qui va permettre de faire cela, de la façon suivante :

La fonction *d* (comme différence finie ou distance)
 $d(x)$ calcule les changements de l'objet x ; si x varie, $d(x) \neq 0$.
 Pour un point M , $d(M)$ est la distance à la position précédente.



Réinitialisation de $x(\text{Min})$ quand P se déplace :
 $\text{if}(d(P)>0;x(M)-1;\min(x(\text{Min});x(M)))$

Réinitialisation de $x(\text{Max})$ quand P se déplace :
 $\text{if}(d(P)>0;x(M)+1;\max(x(\text{Max});x(M)))$

Alors dès que l'on déplace le point P, les points *Min* et *Max* sont à une unité de M :



La récursivité permet de s'affranchir du déterminisme du logiciel, en incorporant au sein de la figure, non seulement le temps de l'utilisateur, mais aussi des règles de réinitialisation de ses paramètres. Nous avons pourtant dit que le paradigme de la géométrie dynamique se construisait dans la rencontre – et la distance maintenue – entre la temporalité de l'utilisateur et l'atemporalité de la figure. Et maintenant nous introduisons la première dans la seconde. Que devient alors cette distanciation ?

En pratique, insérée dans une figure, une situation a-déterministe n'est pas perçue comme telle par l'utilisateur car elle a d'abord un sens mathématique, et c'est à travers lui qu'est interprétée la manipulation. Pourtant, alors que la didactique plonge l'apprenant dans un milieu pour mieux le modéliser (Brousseau, 1986), ici c'est le comportement de l'EIAH, par la récursivité de ses objets, qui est plongé dans le milieu de l'utilisateur.

La modélisation didactique produit ainsi une récursivité croisée que nous allons illustrer sur un exemple plus loin mais dans un contexte non scolaire (pavages hyperboliques), qu'il conviendrait d'expérimenter dans des situations et des niveaux scolaires variés pour en faire une analyse approfondie. Dans un premier temps, nous dirons simplement que la figure *intègre la globalité du milieu* avec le plongement dans le temps³⁰ de la manipulation, éventuellement le temps de l'apprentissage.

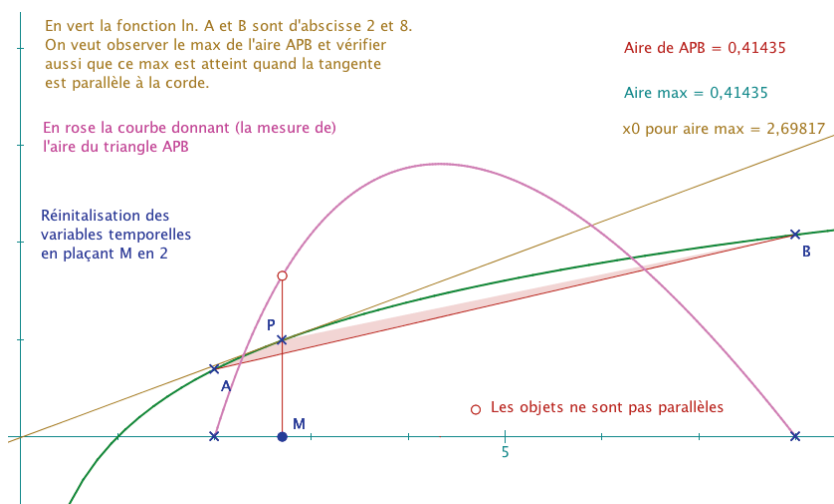
2. Récursivité et magnétisme – Première approche

Ce qui vient d'être fait avec les points peut l'être aussi avec les expressions, avec cette précaution que la création d'une expression, même mise à 0 par défaut par le logiciel, *ne l'initialise pas*. Il faut donc trouver un critère

³⁰ On reste dans le domaine temporel, on ne tient pas compte ici des choix multiples des outils de contrôle comme les curseurs, les menus, les cases à cocher qu'on peut mettre dans une figure qui ne sont que des choix conditionnels ou des paramètres de la figure. Tout ceci reste bien entendu déterministe tout comme les animations qui correspondent à des horloges que l'on déploie dans la figure sans s'affranchir du déterminisme non plus (pour le moment, nous y reviendrons).

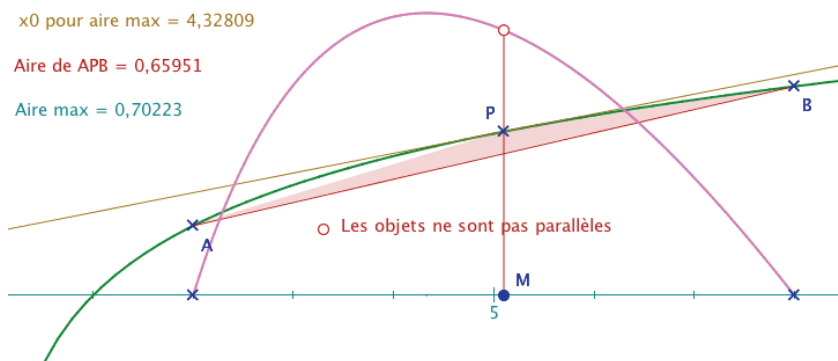
d'initialisation au sein de la figure. En pratique, souvent un simple critère de réinitialisation suffit (sauf bien sûr dans des contextes de compteur automatique comme dans l'exemple de l'aiguille de Buffon).

Avant d'aborder une nouvelle possibilité de l'engagement direct que peut procurer la rencontre entre le magnétisme et la récursivité, commençons par regarder une application scolaire qui illustre à la fois la valeur ajoutée de l'affranchissement du déterminisme et du magnétisme mais aussi, *leur extrême discrétion*³¹ au sein d'une figure, au point que l'utilisateur ne le perçoit pas. Le curseur précédent – adapté aux expressions – va servir à conserver le maximum d'une fonction dans la situation suivante :

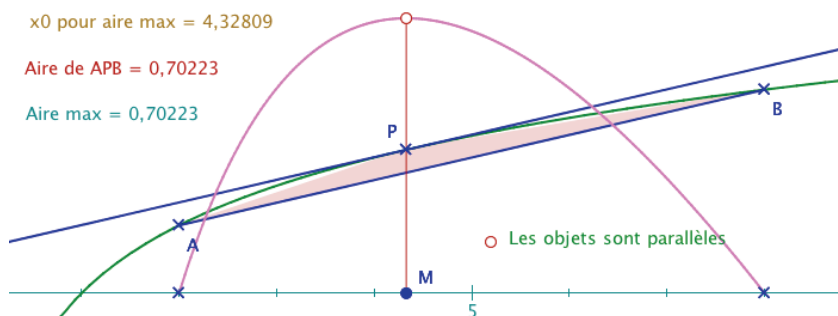


On note que les variables *Aire max* et x_0 pour aire max ont bien chacune une valeur à l'ouverture de la figure qui est la valeur courante de la figure. On observe le critère de réinitialisation (avec un texte adapté à la formation des enseignants et non pas à un public scolaire). Quand M a franchi l'abscisse du maximum, la figure a cette allure :

³¹ Comme une a-modalité de l'affranchissement du déterminisme, aussi naturellement disponible qu'imperceptible.



Les deux expressions temporelles conservent bien les extrema qu'elles ont rencontrées. Puis nous faisons passer M par « le point » qui réalise le maximum. Ici ce point a une abscisse transcendante, il est donc inaccessible – et irréprésentable en machine.



Exemple de réalité mathématique augmentée par l'aimantation

Le point M passe par le point d'abscisse $3/\ln(2)$, a priori inaccessible à la manipulation directe et ceci dans une réelle continuité du mouvement de M à la souris.

Mais M est aimanté par la valeur associée par l'ordinateur et donc les algorithmes internes de test fonctionnent correctement³².

³² Pour obtenir ce comportement systématiquement sans difficulté, les logiciels travaillent à une précision sur les nombres (10^{-13} par exemple pour un comportement multiplateforme) et à un ordre de grandeur assez différent pour les tests 10^{-10} pour CaRMetal). C'est l'occasion de noter que si les entiers sont passés en 64 bits – ce qui est pratique pour bien des choses – les bibliothèques standard des réels ont perdu en

3. Engagement direct augmenté – Déterminisme enrichi

Tout comme le magnétisme a introduit une réalité mathématique augmentée en donnant accès à la manipulation directe, comme ci-dessus, et à des situations mathématiquement significatives d'ordinaire non accessibles, un mélange entre l'introduction de la temporalité de l'utilisateur et le magnétisme va permettre de faire la même chose sur l'engagement direct au sein d'une figure. On se place dans une situation où l'engagement direct peut, justement, être frustrant³³, c'est-à-dire une situation où l'on aurait envie que l'engagement direct décolle de l'interaction entre un objet géométrique et l'utilisateur et s'élargisse, dans un rapport nouveau, à toute la figure dans sa globalité. Voyons un exemple concret.

On s'intéresse désormais aux possibilités d'exploration des pavages hyperboliques par des carrés : $P(4, k)$, pour $k = 5, 6$ et 8 , c'est-à-dire avec $5, 6$ ou 8 carrés autour de chaque sommet. S'il est – éventuellement³⁴ – difficile de construire les cercles supports de ces pavages, il est très simple de simuler une approche de ces pavages en manipulation directe. L'engagement direct permet alors, par exemple quand on est à moins de $0,1^\circ$ de la solution, de signaler que l'on est proche de ce pavage, de le décrire par exemple en fonction de k . Mais bien entendu, on ne peut manipuler la figure dans cet environnement et poursuivre un travail dans cette situation.

La manipulation directe permet seulement d'approcher la configuration du pavage. L'ajout d'un magnétisme permettrait de placer l'utilisateur dans une configuration précise, par exemple $P(4, 5)$, il pourrait travailler dans ce contexte-là, mais ne pourrait pas en sortir³⁵.

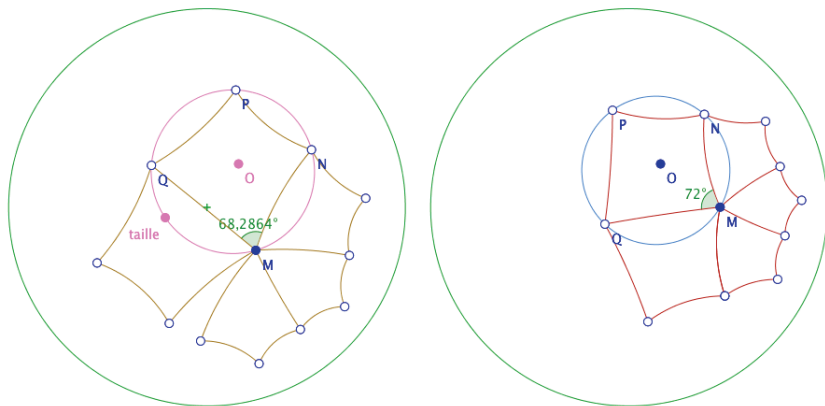
double précision : les premières versions de Cabri sur Mac étaient à 10^{-18} pour les coordonnées et 10^{-14} pour les tests.

³³ Cet exemple relève effectivement d'une « frustration » de n'avoir pu construire des ingénieries didactiques sur l'exploration des pavages hyperbolique lors de mon premier travail sur ce thème (thèse sur les GNE en 2003) avec Cabri-géomètre : à cette époque l'incorporation de la temporalité comme on le fait ici n'était pas envisageable, c'était juste une *impossibilité* non pas technique, mais bien *théorique*, ce qui illustre l'évolution non seulement des EIAH mais aussi de nos représentations de ce qui est « théoriquement possible ».

³⁴ Cela dépend si on veut faire une construction entièrement géométrique, c'est effectivement difficile pour le maintien de la continuité, ou si on s'autorise un recours à la trigonométrie hyperbolique, et alors c'est bien plus simple. En 2003 nous avons fait le premier choix, conformément à l'orientation générale du travail, ici le second.

³⁵ On pourrait penser qu'il suffit de diminuer le coefficient d'aimantation, mais si on fait cela, pendant la manipulation du point M, on passe d'un pavage $P(4, k)$ à un autre

C'est le cas de l'illustration suivante. Nous sommes dans le modèle du disque de Poincaré; on commence un début de pavages de carrés, autour du point M sur un cercle générique de centre O passant par le point *taille*.



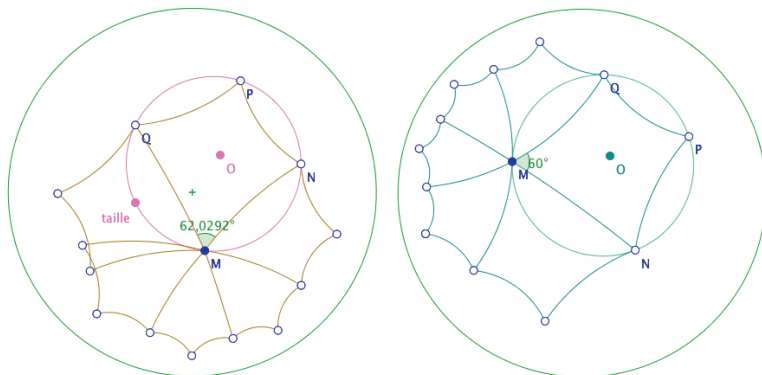
En déplaçant le point taille (à gauche), on se rapproche du cercle réalisant $P(4, 5)$. Quand on est proche (détailé ci-dessous), le cercle du pavage attire le point M qui passe alors sur celui-ci, la configuration prenant alors une allure de pavage $P(4, 5)$, le point taille disparaissant. On voit sur l'illustration de droite qu'on peut déplacer M dans cette configuration, et travailler spécifiquement dans un environnement $P(4, 5)$.

Ceci aurait assez peu d'intérêt si on ne pouvait sortir de cette configuration pour aller vers un autre pavage. Pour cela on introduit une temporalité sur l'aimantation, que l'utilisateur peut briser très simplement en déplaçant le centre O du cercle hyperbolique du pavage. La mise en œuvre est en fait très simple. L'attraction de M par le cercle de pavage (ici $P45$ de rayon connu $r45$) se fait en deux temps : à partir du booléen $att45 = \text{abs}(OA - r45) < 0,01$ (A est le véritable nom interne du point d'alias *taille*), on calcule cet autre booléen temporel : $Sur45 = \text{if}(att45 == 0; 0; \max(Sur45, att45))$ qui lui, est utilisé pour attirer M sur le cercle $P45$.

Ainsi dès que $att45$ vaut 1, $Sur45$ vaut 1 aussi, et ceci, temporellement, tant que $att45$ ne revient pas à 0, ce qui ne se produit que si l'utilisateur

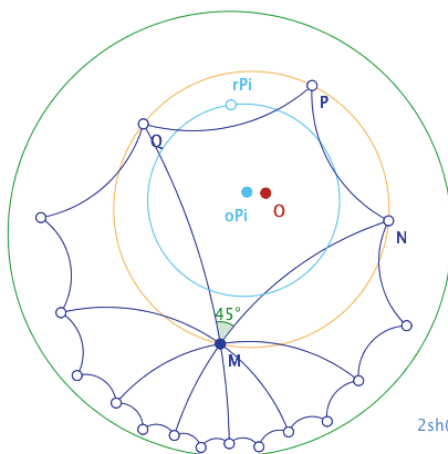
car ils sont très proches les uns des autres. L'environnement proposé est alors perçu comme instable. C'est pour cela qu'il faut trouver autre chose.

déplace, même légèrement³⁶, le centre O du cercle. On notera que toutes les variables sont initialisées à 0 dans une configuration générique.



Après avoir déplacé le point O, taille réapparaît, on se place alors dans une autre situation de pavage.

On réitère le procédé pour aller jusqu'au pavage P(4, 8) qui contient un petit bonus :



Le carré d'angle $\frac{\pi}{4}$ est l'exemple donné par Bolyai (1832) pour montrer qu'en géométrie hyperbolique, la quadrature du cercle est possible.

En effet l'aire de ce carré est π . Bolyai a montré que le côté x de ce carré est constructible (car il vérifie $\text{ch}(x) = \frac{2\sqrt{2}}{\sqrt{2}-\sqrt{2}}$ soit constructible) ainsi que le rayon r du cercle de d'aire 1 par une transformation entre un arc et un angle :

✓ Voir un cercle d'aire 1

$$2\text{sh}(r/2) = 1$$

$$r = d(oPi, rPi) = 0,96242$$

$$\text{Aire} = 4\pi \cdot \text{sh}^2\left(\frac{r}{2}\right) = \pi \cdot \tan^2(\theta)$$

constructible pour $\theta = \frac{\pi}{4}$

³⁶ On notera que l'ingénierie didactique est telle que c'est l'utilisateur qui choisit de rentrer dans le processus et qui a la main pour en sortir.

L'engagement direct est *augmenté* au sens où c'est la figure dans son ensemble – et non plus un élément – qui réagit à la sollicitation de l'utilisateur : celui-ci est temporairement installé dans un environnement où il peut *réaliser son intention et l'explorer*. Une combinaison du temps de l'utilisateur – utilisé de manière très discrète, à peine perceptible – et de l'aimantation permet, à faible coût, d'enrichir considérablement les environnements de travail ou de simulation que peut proposer la géométrie dynamique.

En dernière analyse, on remarquera que, dès que l'on se donne les moyens de réinitialiser un comportement de type temporel au sein d'une figure, on rétablit de fait, le déterminisme de la figure : quels que soient les îlots de non déterminisme dans les figures précédentes, on peut toujours revenir à la position initiale. *S'affranchir du déterminisme* avec ces outils-là ne crée donc pas un imprévisible dans les figures, comme pourrait le faire la continuité, mais au contraire permet de créer des environnements plus riches, et pourtant entièrement contrôlés et prévisibles. En définitive ces outils permettent de construire un *déterminisme enrichi*. Et qui plus est, *enrichi d'îlots de non déterminisme*.

4. Récursivité croisée en manipulation directe

Dans les figures précédentes, les réalisations récursives donnaient des points qui, par construction, ne pouvaient être en manipulation directe. Il est possible de produire une récursivité croisée accessible à la manipulation directe. Dans cette figure de modélisation d'un bâton de longueur AB glissant sur le sol et le long d'un mur, on peut agir sur le point Q, le point P étant traditionnellement obtenu par l'intersection de l'axe vertical et du cercle de centre Q de rayon AB. Mais ici ce n'est pas le cas, on peut aussi déplacer [PQ] par le point P.

Pour réaliser cela les coordonnées de P et Q sont définies de manières croisées, ainsi pour P (et de même pour Q) :

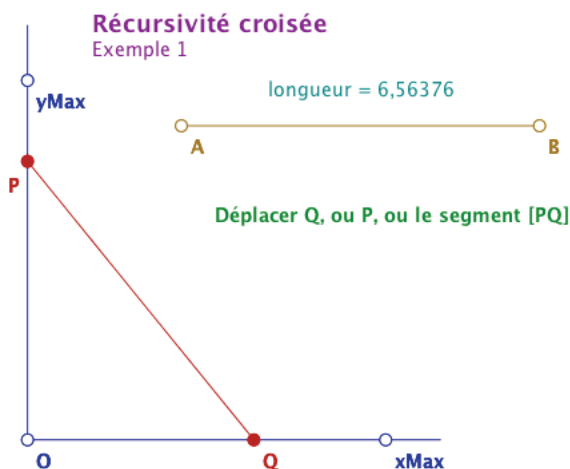
$$x(P) = x(@O)$$

$$y(P) = \text{if}(y(P) < y(@yMax); y(@O) + \text{sqr}t((x(@B) - x(@A))^2 - (x(@Q) -$$

$$x(@O))^2) + 0 * x(P); y(@yMax))$$

La règle principale est d'utiliser @ pour définir une récursivité croisée. Pour que le point soit manipulable à la souris, on notera³⁷ le « +0.x(P) ».

³⁷ Détail sur l'ensemble des règles – et exemples scolaires – dans cette page : <http://www.reunion.iufm.fr/recherche/irem/spip.php?article40>

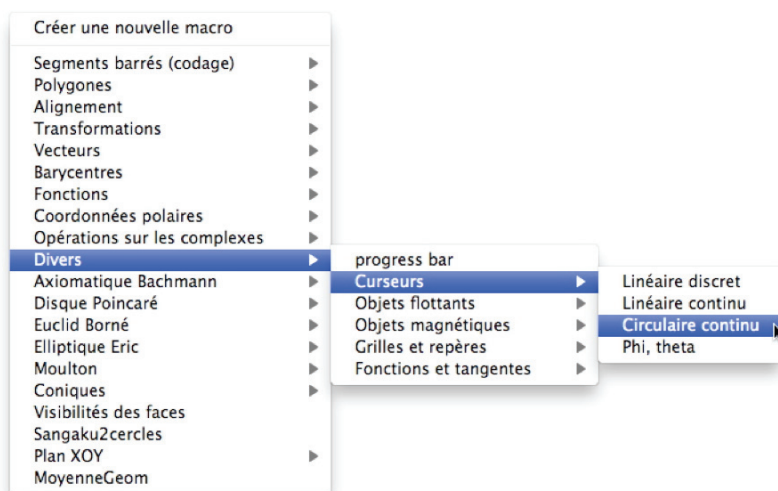


VI. Un autre traitement du temps

Dans la partie précédente nous avons vu comment la réversivité permettait de s'affranchir du déterminisme et créait ainsi, avec l'aimantation, un *déterminisme enrichi*. Mais celui-ci, tel qu'il a été proposé, ne permet pas de résoudre la question de l'enroulement continu sur un cercle qui est bien entendu un autre contexte d'affranchissement du déterminisme : on veut pouvoir compter le nombre de tours et pas seulement mesurer un angle sur une amplitude d'un seul tour.

Cette question – sans solution dans un environnement entièrement déterministe de géométrie dynamique – a été résolue par Éric Hakenholz, dès 2005, donc avant qu'il envisage même de commencer CaRMetal, dans ses explorations subtiles et fouillées de CaR. Il en a fait un article³⁸ (avec applet) sur son site CaRZine alors dédié à CaR. C'était la première fois qu'un logiciel de géométrie dynamique permettait de réaliser, en manipulation directe, l'uniformisation du cercle, et probablement ce résultat a-t-il participé à convaincre le futur auteur de CaRMetal à « donner un petit coup de pouce à ce logiciel », comme c'était son projet initial. Depuis, il a transformé cette possibilité en macro standard de CaRMetal, présente dès la première version, sous le nom de *Curseur/Circulaire continu*.

³⁸ <http://db-maths.nuxit.net/CARzine/articles/art91/>

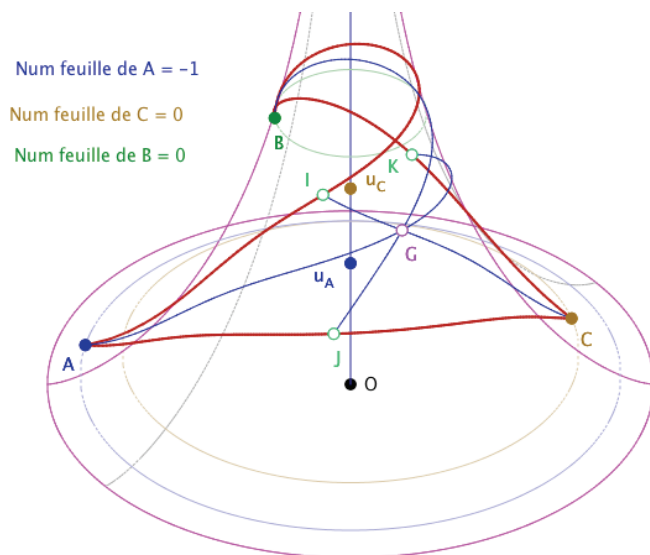


Nous ne reviendrons pas plus avant sur cette technique³⁹. Disons simplement qu'Éric Hakenholz utilise un subtil mélange de la fonction d de CaRMetal, déjà utilisée pour réinitialiser la récursivité, et d'une fonction sum qui, comme son nom l'indique, somme les valeurs d'une variable numérique.

Cette possibilité d'enroulement sur le cercle se transpose dans de nombreuses situations et, en particulier, pour ce qui nous concerne directement, dans les micromondes des géométries sur les surfaces de Beltrami. On peut donc reprendre un travail plus fin que celui mené sur la feuille principale pour construire un nouveau micromonde qui incorpore l'enroulement sur les surfaces pseudosphériques. Dans le cas de la pseudosphère, s'il faut refaire toutes les macros, elles peuvent toujours rester intrinsèques à la surface. On obtient alors des figures comme celle-ci avec un triangle ayant ses trois sommets sur trois feuilles différentes de la surface :

³⁹ Reprise, plus en détail, sur cette page :

<http://www.reunion.iufm.fr/recherche/irem/spip.php?article50>



Les médianes d'un triangle hyperboliques sont concourantes

VII. Les CarScripts

Depuis sa version 3.0 d'août 2009, le logiciel s'est doté d'un langage de scripts pour pouvoir travailler la programmation principalement dans un environnement graphique *dynamique*. CaRMetal n'est pas le seul logiciel de géométrie dynamique avec un langage de script, mais c'est le premier libre et multiplateforme⁴⁰.

Cette évolution n'est pas sans lien avec celle des programmes du lycée dont l'introduction de l'algorithmique et la pratique de la programmation. Du côté de l'élève il s'agit de proposer un environnement commun pour la géométrie et la programmation – économie de temps d'apprentissage – avec un langage non pas construit uniquement pour une utilisation scolaire, avec les artifices qui vont avec⁴¹, mais bien un langage contemporain, largement utilisé dans le monde Web 2.0, pour lequel les navigateurs rivalisent en temps d'exécution : le JavaScript (JS).

⁴⁰ Cinderella possède aussi son langage de scripts, mais il n'est ni libre ni gratuit.

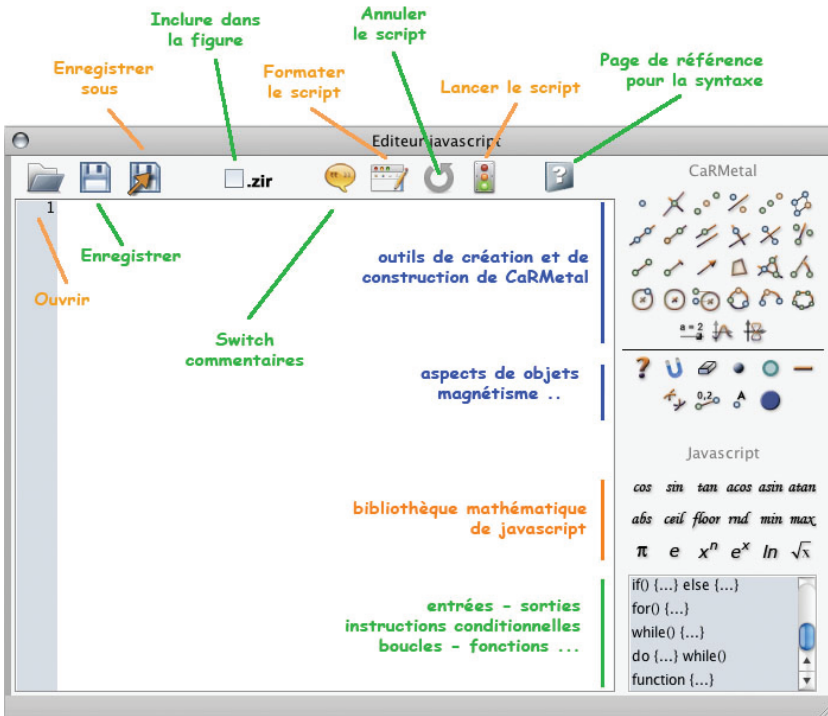
⁴¹ On se souvient de l'expérience du LSE.

Nous nous proposons dans cette partie, tout en présentant succinctement les possibilités de cet outil, d'analyser les nouveaux jeux de cadre que peut permettre une rencontre entre la géométrie dynamique et la programmation.

1. Introduction aux CarScripts

Pour l'auteur, le choix de JS relève d'un cahier des charges très précis : langage pérenne, intégrable dans un logiciel libre, utilisable en ligne, et qui plus est faiblement typé pour pouvoir ajouter les fonctionnalités géométriques de CaRMetal.

L'éditeur JavaScript de CaRMetal intègre toutes les fonctionnalités géométriques du logiciel sous forme d'icônes, ainsi que la bibliothèque Maths du JS et les principales instructions de programmation. L'utilisation du JS est ainsi simplifiée : il n'y a pas de syntaxe à apprendre, on ne fait que modifier des lignes pré-écrites. Les lignes sont numérotées (renvoi du numéro de la ligne dans les erreurs) et il y a une coloration syntaxique.



Pour les premiers usages en classe, l'utilisation peut se faire en « pur JS » avec une sortie en console texte. Cette sortie est simple d'accès et pratique pour tout ce qui peut toucher les statistiques ou l'algorithmique en arithmétique par exemple. On se centrera dans la suite sur la sortie dans une fenêtre de CaRMetal car c'est là qu'est la richesse du langage puisque son intégration au logiciel est *totale*ment dynamique comme on va le voir au cours des pages suivantes.

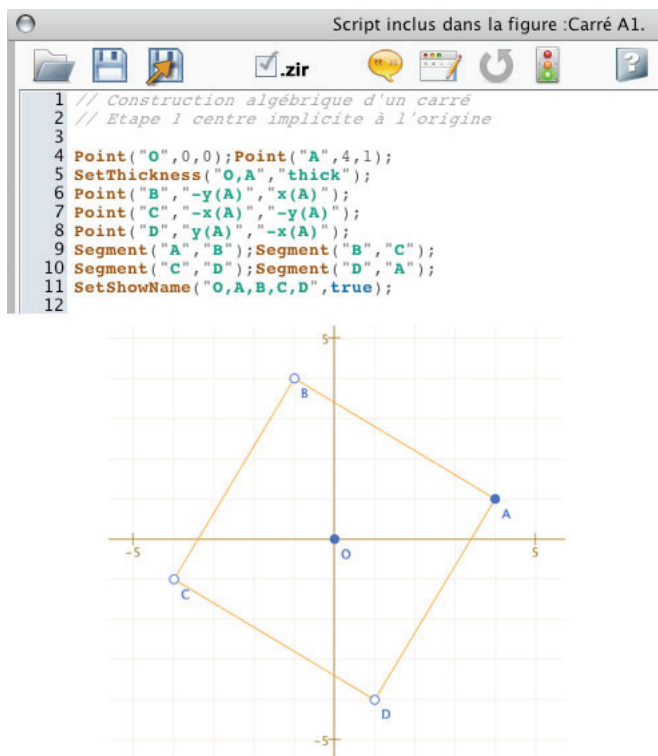
Un script de CaRMetal – on dira désormais un CarScript – peut s'appliquer à une figure vide ou à une figure préparée pour son exécution. Il y a alors deux points de vue différents. Du côté de la programmation, un CarScript est avant tout un programme qui admet comme paramètre d'entrée une figure de CaRMetal. Du côté de la géométrie, un CarScript est un outil qui va finaliser une figure que la géométrie dynamique usuelle ne pourrait pas faire – ou pas aussi rapidement. Dans un cadre scolaire, selon l'activité proposée, les choix didactiques, ses variables, l'utilisateur va être plongé, pour son objet d'étude, plutôt dans l'un des deux points de vue, l'autre servant d'outil pour la situation.

2. Géométrie repérée dynamique

Nous allons nous attarder sur ce premier thème, très élémentaire, mais significatif de l'évolution possible des pratiques scolaires dans le cadre des nouveaux programmes du lycée. En effet, avec les nouveaux outils qu'il propose, CaRMetal autorise un nouveau regard sur la géométrie repérée car il permet de faire – très simplement – des programmes autour de ce que nous appellerons de la *géométrie repérée dynamique*, c'est-à-dire de la géométrie repérée, tournée du côté de l'algèbre. Sans aucune technicité, nous utilisons d'emblée cette spécificité dynamique de l'interaction JS-logiciel. Voyons comment sur l'exemple de la construction algébrique d'un carré, en plusieurs étapes.

Tout d'abord, commençons par un carré particulier, centré à l'origine O du repère, et de sommet A . Pour l'essentiel, il suffit donc de savoir placer un segment orthogonal à $[OA]$ et de même longueur. Selon la classe, c'est un résultat bien connu sur l'orthogonalité, soit des droites soit des vecteurs. On peut aussi se placer en début de lycée, quand la propriété peut ne pas être connue. Une démarche (attitude mathématique du socle commun) consiste alors à utiliser une lecture graphique dans un premier temps et à engager une interprétation algébrique de cette lecture. Cette interprétation algébrique peut même être un objectif d'induction du numérique vers l'algébrique, partici-

pant à se représenter le champ algébrique comme conceptualisation du champ numérique. Le script suivant peut alors se construire par étapes :



Pour écrire les coordonnées du point B (ligne 5) sans connaître le résultat, on peut envisager la démarche *d'investigation algébrique* suivante, désormais possible grâce à cet outil :

Attitude d'investigation en géométrie repérée

- entrer la ligne 4 comme seul contenu du script,
- l'exécuter,
- lire les coordonnées $(-1, 4)$ du point à construire,
- et conjecturer l'interprétation algébrique de la ligne 5.
- Puis annuler le script précédent, ajouter la ligne 5, exécuter le script,
- ET valider par une manipulation directe du point A.

C'est la dernière ligne de cette séquence qui donne à la démarche proposée son statut d'investigation. En effet les coordonnées des points, données dans un script, *ne fixent pas les points* dans la figure, mais *les initialisent* seulement. En faisant cette manipulation, l'élève peut déplacer A sur d'autres coordonnées entières pour non seulement valider l'orthogonalité de manière perceptive mais aussi valider que sa compréhension algébrique a bien été prise en compte dans le script comme on peut le penser. On notera que le point O ne sert pas, mais souvent les élèves placent le centre d'une figure avant de commencer. On vérifie par manipulation que le carré n'est pas véritablement de centre O mais bien implicitement de centre l'origine du repère. L'objectif suivant est alors de faire un authentique travail algébrique – avec une fiche pour une trace dans l'environnement papier-crayon – non plus seulement pour explorer avec les scripts mais bien *pour anticiper et produire* les coordonnées correctes.

Géométrie repérée dynamique : travail algébrique validé par la manipulation directe

On a choisi depuis plusieurs années d'utiliser massivement le tableur comme outil d'intermédiation entre l'arithmétique et l'algèbre : le tableur a été retenu car il permet de travailler sur les variables sans pour autant faire de l'algèbre. D'un point de vue didactique, c'est de l'algèbre « sur des objets⁴²», ces objets étant les cellules. On peut rester dans une certaine opérationnalité du travail sur feuille de calcul, et peu progresser vers l'algèbre, en particulier si ce n'est pas dans le programme, ou au contraire, utiliser les cellules comme un travail « sur les objets ». Une cellule d'un tableur, avec son nom différent de son contenu, ayant été considéré par les concepteurs de programmes comme une première représentation acceptable de la variable algébrique.

La démarche que l'on propose ici, d'un travail algébrique sur la géométrie repérée avec validation par la géométrie dynamique à travers les scripts se situe clairement plus du côté de l'algèbre que le tableur. Mais elle garde aussi cette dimension de « calcul sur les objets » – réification de l'algèbre – car une fois le script exécuté, les élèves peuvent manipuler les points qu'ils ont eux-mêmes construits. La dimension algébrique est alors acceptée comme *embraquée dans ce projet* de travail sur les objets. Et par cette manipulation de

⁴² Au sens du « calcul sur les objets » d'auteurs cognitivistes comme Brissiaud ou Ouzoulias dans leurs ouvrages pour l'école primaire : le passage par des objets (ici déjà conceptualisés) est un *accélérateur de conceptualisation* de ce que l'on veut enseigner. Le succès des méthodes Tchou et Picbille en est une illustration.

la figure, avec les objets produits par leurs propres scripts les élèves acquièrent aussi⁴³ des connaissances algébriques. Un travail régulier sur ce type de scripts dans un environnement dynamique, outre qu'il est pleinement en accord avec les nouveaux programmes, s'inscrit dans une logique générale d'apprentissage qui a amené à inscrire le tableur dans de nombreux programmes, tout en restant dans une véritable démarche algébrique. D'un point de vue didactique, il y a des hypothèses de travail à préciser et tout un champ de recherche à étudier en situation.

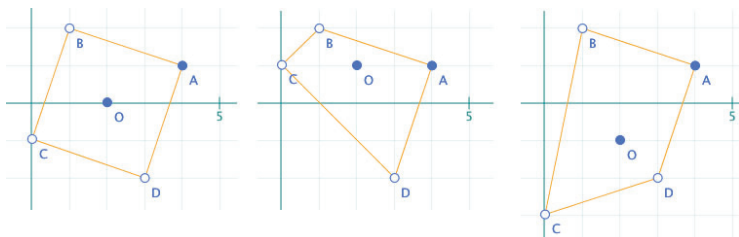
La situation n'est d'ailleurs pas simple pour les élèves. Ayant pris conscience de la non utilisation du point O dans le script précédent, on n'aboutit pas au carré algébrique en une seule étape, loin s'en faut. Parfois les élèves travaillent implicitement sur une seule coordonnée, sans même s'en apercevoir avant d'invalider leur travail par la manipulation des points de base.

```

1 // Construction algébrique d'un carré
2 // Etape 2 - première tentative d'intégration de O
3 // ici correct sur les abscisses
4
5 Point("O",0,0);Point("A",4,1);
6 SetThickness("O,A","thick");
7 Point("C","2*x(O)-x(A)","2*y(O)-y(A)");
8 Point("B","x(O)-y(A)","-x(O)+x(A)");
9 Point("D","x(O)+y(A)","x(O)-x(A)");
10 Segment("A","B");Segment("B","C");
11 Segment("C","D");Segment("D","A");
12 SetShowName("O,A,B,C,D",true);

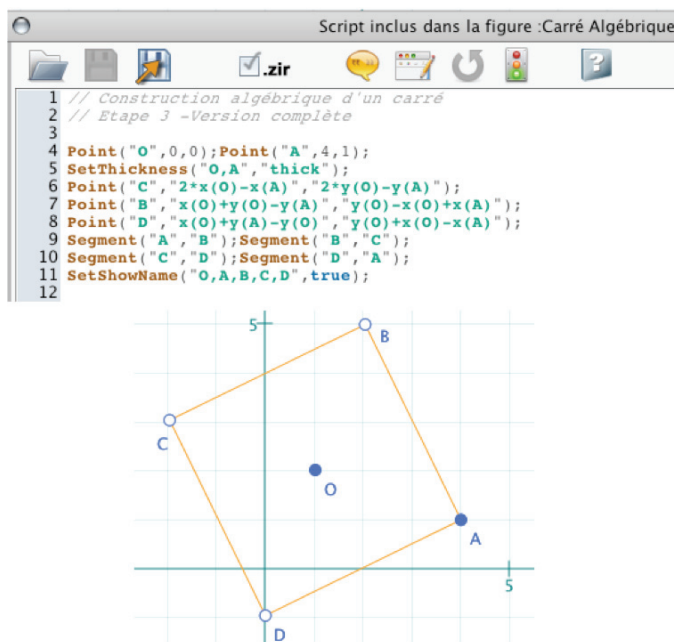
```

Un script qui produit un carré... seulement si O est sur l'axe des abscisses



Par manipulation sur O, et en plaçant O sur des valeurs entières on peut analyser ses erreurs et avoir une aide à l'organisation des calculs...

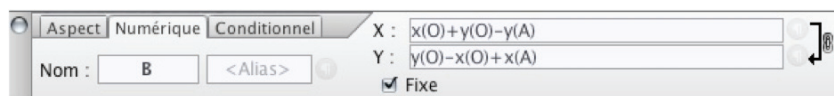
⁴³ Il faut relativiser. Parfois au contraire, devant les interactions langagières entre eux face à leurs productions à l'écran, mais aussi devant leur résistance à voir leur propre capacité à algébriser, il vient à l'idée que ce type de situation est un autre contexte qui valide cette affirmation de/et que Gérard Vergnaud aime rappeler régulièrement à propos des schèmes : « la connaissance opérationnelle est supérieure à la connaissance prédicative ».



... avant d'arriver à une version correcte

On retiendra de ce paragraphe l'enjeu que peut avoir, pour une entrée efficace dans l'algèbre, la manipulation d'objets, produits par les élèves, issus de leurs propres raisonnements algébriques, et les possibilités – selon les élèves, les classes et le projet d'apprentissage – d'une phase que l'on a dite *d'investigation algébrique* pour aider à l'induction vers l'algèbre.

Sur un plan plus technique, on aura observé dans les scripts précédents combien l'écriture mathématique dans les CarScripts est proche de l'écriture mathématique au tableau : pour le moment, c'est la même, elle est seulement envoyée aux points de la figure par une expression entre guillemets – par une chaîne de caractères. On aura bien compris qu'alors on n'envoie pas une initialisation comme quand on donne une valeur numérique, mais bien un calcul littéral. Voici par exemple les coordonnées du point B dans le logiciel :



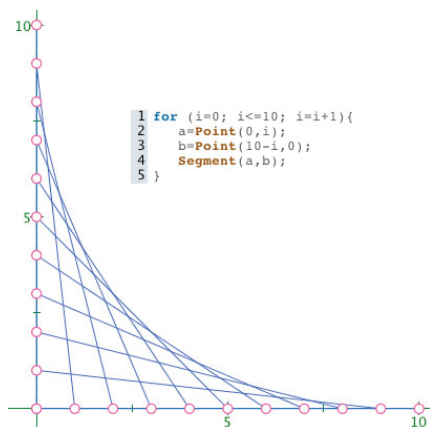
Cela signifie aussi que, dans des classes où l'algorithmique n'est pas au programme, la même démarche, mais simplement *dans l'inspecteur d'objet*, est aussi une pratique de géométrie repérée dynamique, elle aussi plus proche de l'algèbre que la pratique des tableurs. L'inspecteur d'objet, comme les autres outils, a un fonctionnement a-modal y compris dans l'écriture algébrique : chaque écriture (même partielle) qui a un sens algébrique est automatiquement interprétée et la figure actualisée, ce qui facilite l'investigation.

3. Insertion de variables JavaScript dans les figures – Exemples en analyse et statistique.

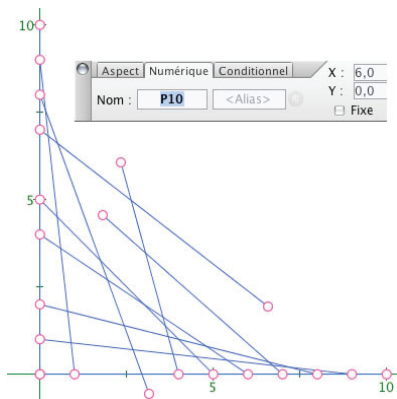
Les scripts précédents sont élémentaires au sens où ils sont écrits essentiellement du côté de la géométrie. En fait, ce sont plutôt des exercices de géométrie analytique que de programmation, auto-validés par l'utilisation des scripts et la manipulation directe sur la figure produite. Il y a donc, déjà, l'utilisation d'une interaction entre le JS et le logiciel puisqu'il y a transmission d'informations algébriques, mais ce n'est pas uniquement ce type d'interaction que l'on peut attendre d'une intégration réussie. Qu'en est-il du passage des variables JS dans les figures de CaRMetal ? Toujours dans un contexte scolaire, poursuivons notre exploration de la géométrie repérée dynamique en nous intéressant à la réalisation d'un *tableau de fils dynamique*.

Un préalable statique

Cette activité suppose avoir fait au préalable, un *tableau de fils statique*, par exemple sur les coordonnées entières dans le cadre d'un apprentissage de la boucle *for*.

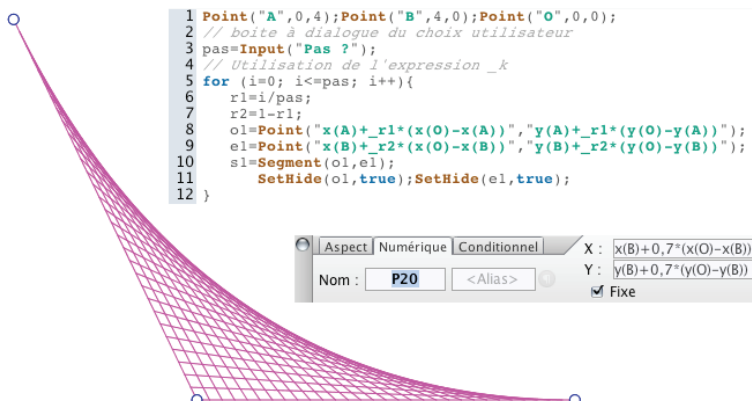


Mais puisque tous les points sont initialisés – certes à la bonne position – on n’a fait qu’un dessin – ce qui est déjà un objectif tout à fait honorable en seconde – ce n’est pas une figure « tableau de fil » au sens où il y aurait une cohérence globale interne : tous les points sont indépendants



Fils dynamiques par boucle d’itération

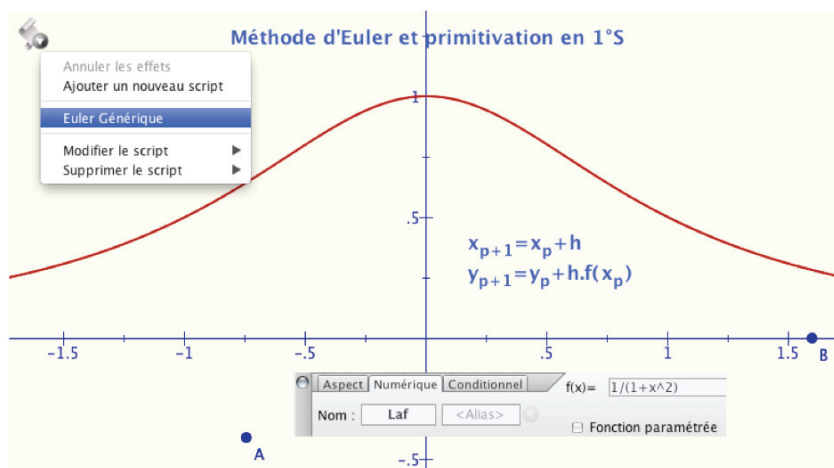
Faire un tableau dynamique à partir de trois points A, O, B nécessite de partager les segments [OA] et [OB], de parcourir ce partage par une boucle et de construire des points associés. Il va donc falloir envoyer une variable JavaScript à CaRMetal. La syntaxe est celle de l’adressage indirect des logiciels de programmations actuels : pour envoyer le contenu de la variable k , on utilise $_k$ dans les coordonnées des points, comme dans ce script :



En lisant la ligne 1 du script, on remarque que les segments [OA] et [OB] sont initialement orthogonaux. Sur l'illustration, le point A a été déplacé après le tracé du tableau de fils qui, parce qu'il est *dynamique*, devient en fait un tableau *affine*. À part l'underscore *_k* le reste de l'écriture est conforme à ce que l'on pratique en classe. Dans l'illustration on a placé les coordonnées de l'un des points : elles sont fonction de O et B, et du *contenu* de la variable *k*. Ce passage *par contenu* – plutôt que par valeur comme c'est le cas ici – va être plus explicite dès l'activité suivante, quand la variable ne sera pas uniquement une valeur numérique mais une structure un peu plus complexe.

4. Une figure CaRMetal comme entrée d'un script : scripts génériques

Jusqu'ici nous avons vu des scripts qui produisaient des objets dans une figure vide. Nous allons voir la réalisation des scripts génériques, qui s'appliquent à toutes les situations d'un même type, celles-ci étant considérées comme des entrées du script. Nous allons aborder cela sur un thème du lycée qui va nous permettre en même temps d'aller un peu plus loin sur l'échange entre les variables JS et CaRMetal. La situation retenue est la méthode d'Euler dans le contexte de la classe de 1^{re} S comme primitivation approchée de fonction dont « on » (les élèves) ne connaît pas de primitive. Nous avons choisi de prendre la fonction proposée dans les documents d'accompagnement des programmes.



Le script est déjà contenu dans la figure. Du côté de la figure, il va s'appliquer à deux points A et B et une fonction *Laf*. Du côté du programme,

celui-ci prend en compte ses valeurs d'entrée, A et B et *Laf* : la figure est une donnée du script pour son exécution.



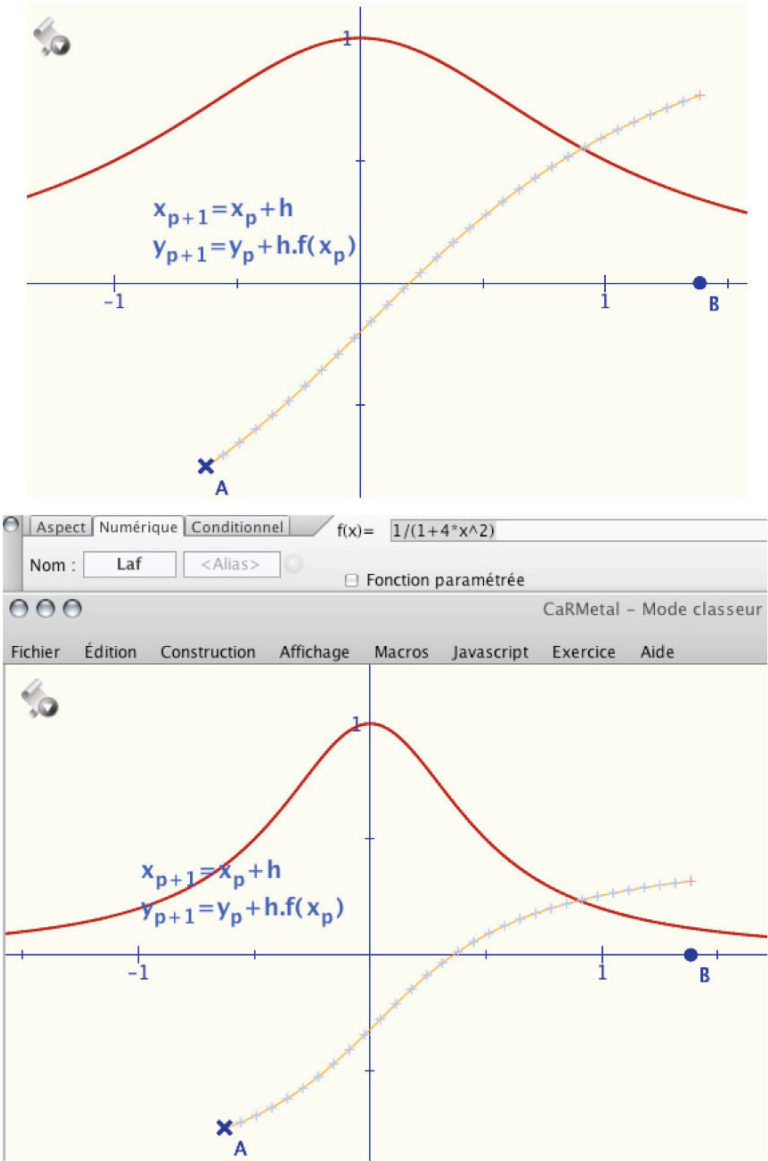
On notera l'écriture du pas avec A et B, pour que le résultat soit dynamique⁴⁴.

À propos de la syntaxe simplifiée : cette fois on passe par contenu non plus des variables numériques, mais des points et plus précisément leurs coordonnées. Il aurait fallu écrire x_m mais l'auteur a introduit dans ce cas-là, conformément à l'usage dans les langages de programmation ou les CAS, la simplification syntaxique x_a et y_a pour $x(a)$ et $y(a)$, écritures qui restent fonctionnelles. C'est pour cela que l'on garde l'expression *passage par contenu* plutôt qu'un traditionnel « par valeur ».

Dans l'illustration de la page suivante, en haut, l'application du script, en bas le changement de fonction, avec mise à jour immédiate⁴⁵ du résultat du script, puisqu'il est entièrement construit sur ses données initiales.

⁴⁴ Ceci n'est pas propre au script : si on voulait faire la construction géométriquement, par macro, pour que la figure soit dynamique, il faudrait aussi prendre le pas en fonction de A et B. Détails dans cette vidéo flash de Monique Gironce : <http://db-maths.nuxit.net/CaRMetal/tutoriels/ScriptEuler/ScriptEuler.htm>

⁴⁵ Rappelons l'importance didactique de l'absence de modalité dans les onglets de l'inspecteur d'objet : sur les fonctions par exemple, les élèves voient leurs modifications en temps réel, car dès que l'expression écrite a un sens algébrique, elle est aussitôt prise en compte. L'investigation algébrique peut se faire aussi dans cet onglet, sans les scripts.

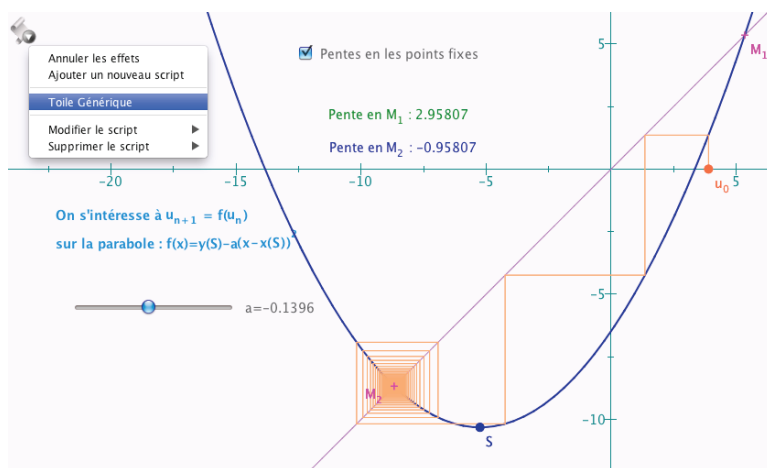


Script de recherche du point fixe $u_{n+1} = f(u_n)$

Toujours dans un contexte scolaire, voici un exemple dynamique, avec une fonction du second degré, avec paramètre et le sommet de la parabole sur l'axe des ordonnées, pour une écriture plus simple de la fonction $y = y(S) - ax^2$ que l'on continue d'appeler *Laf* dans le logiciel. On peut alors appliquer ce simple⁴⁶ script, lui aussi générique (à partir de u_0 et *Laf*) :

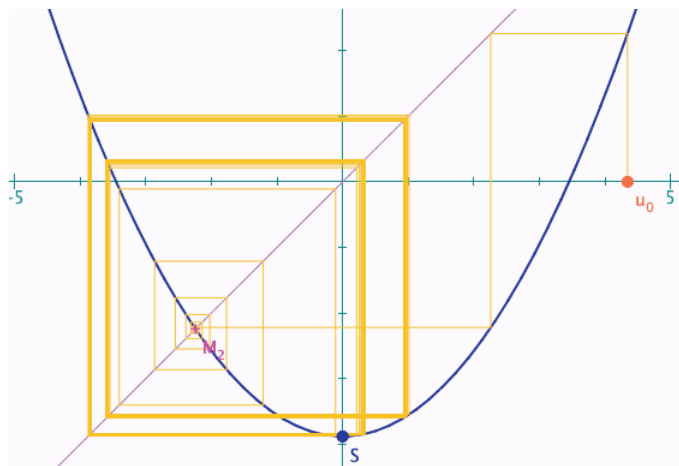
```

1 // fonctionne pour toute courbe nommée Laf avec u0
2 // Le script fait 100 "spirales"
3
4 u="u0";
5 a=Point("x_u", "Laf(x_u)");
6 SetHide(a, true);
7 s1=Segment(u, a);
8 for (i=0; i<200; i=i+1){
9   b=Point("y_a", "y_a");SetHide(b, true);
10  s=Segment(a, b);
11  c=Point("x_b", "Laf(x_b)");SetHide(c, true);
12  v=Segment(b, c);
13  a=Point("x_c", "y_c");SetHide(a, true);
14 }
```

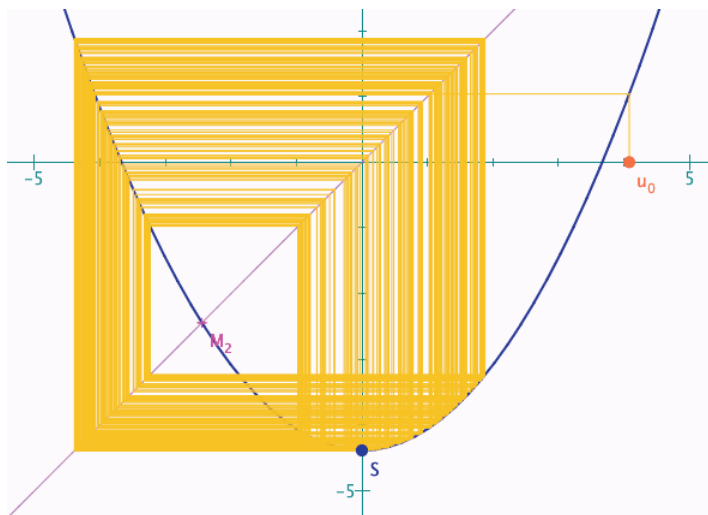


⁴⁶ En formation continue, un enseignant a eu cette remarque qu'il était « bien connu » que cet algorithme « de la toile d'araignée » n'a pas besoin de trois variables, mais que deux suffisent. C'est rassurant pour ce qui est de l'engagement des jeunes enseignants dans les nouveaux programmes. Pour des raisons didactiques évidentes, on peut maintenir trois variables même si ce n'est pas optimisé.

Selon la classe (lycée, BTS, IUT), cet outil *dynamique* permet d'aborder, d'une manière plus dynamique qu'avec un tableur même muni d'un curseur, soit simplement le phénomène de point répulsif, soit encore le concept de valeur d'adhérence (ci-dessous 4) tout ceci simplement *en déplaçant à la souris le sommet S*.

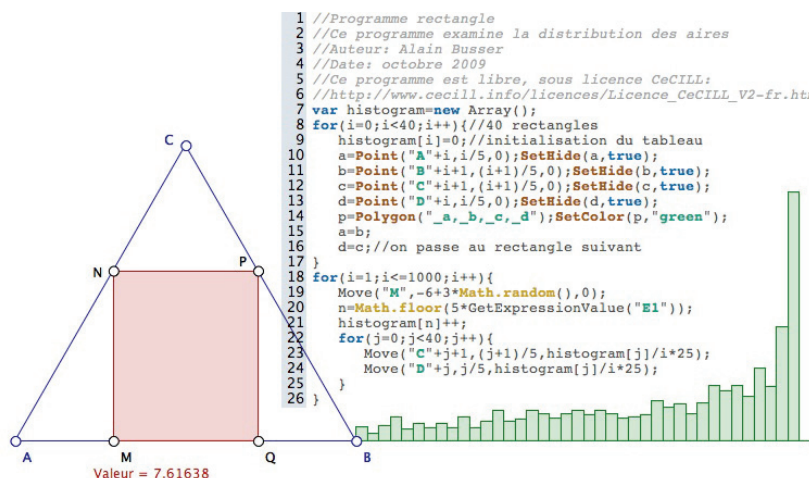


Et toujours en déplaçant S de quelques pixels, on rencontre un phénomène nouveau, le chaos...

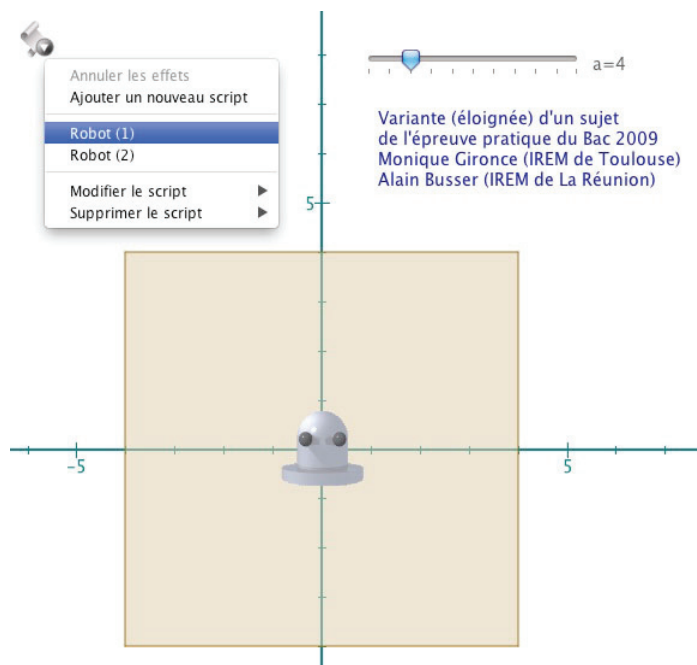


5. Utilisation en statistique

C'est un domaine à la fois simple d'accès, pour lequel l'usage des scripts est pertinent, et qui peut être plus dynamique que ce qui est proposé d'habitude – le forum de CaRMetal sur les CarScripts contient surtout des propositions en statistique. On peut bien sûr faire de simples programmes en « JS pur » avec une sortie texte, on est alors quasiment devant une calculatrice. La sortie graphique est quand même plus intéressante. Voici quelques exemples de contributions générées par les échanges entre l'IREM de Toulouse et celui de la Réunion. Tout d'abord les histogrammes sur les aires de rectangles inscrits dans un triangle.

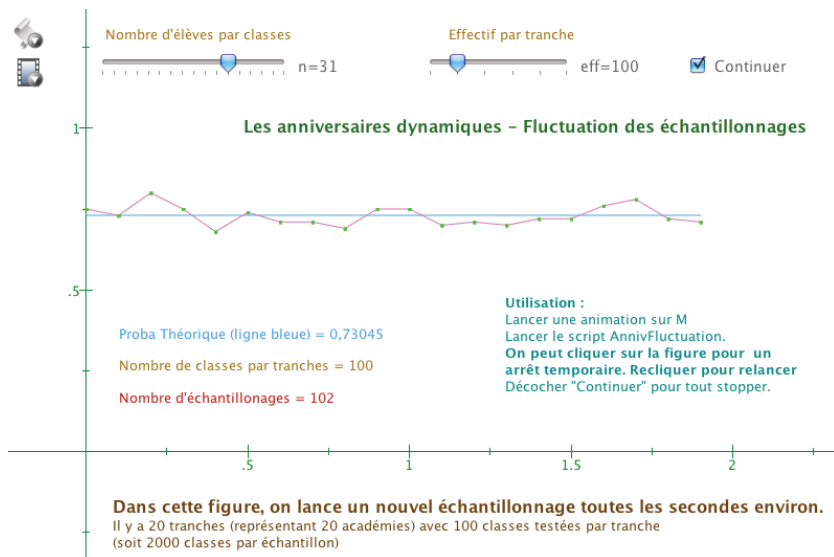


Le thème du parcours aléatoire est fort présent dans l'ingénierie scolaire. Ici il s'agit simplement de modéliser l'exercice, à savoir étudier le nombre de lancers d'un dé à quatre faces pour sortir du carré, et d'étudier statistiquement la situation. Le premier script fait lentement une partie, visible à l'œil, une par lancer du script. Le second fait 20 parties d'affilée pour des premiers résultats statistiques.



D'autres thèmes classiques peuvent être revisités d'une manière plus dynamique que ce qui est proposé habituellement aux élèves. C'est le cas par exemple du problème de la coïncidence des anniversaires dans une classe. Sur un plan technique, nous sommes désormais dans la préparation de figures plus sophistiquées pour les enseignants.

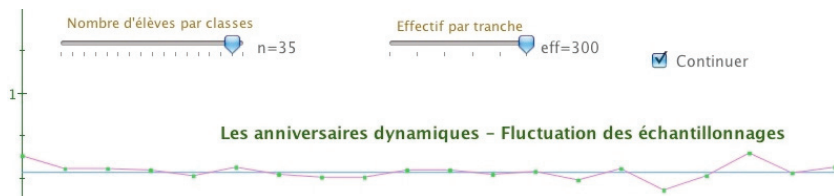
Dans l'illustration suivante, dans chacune des 20 tranches d'échantillons on effectue des tirages pour un effectif de classe donné, allant de 50 à 300 par pas de 50. Dans le même temps – et toujours en temps réel – on peut choisir par classe les effectifs allant de 20 à 36 élèves. C'est bien entendu ce point-là qui est nouveau par rapport aux autres pratiques : les scripts – et la manipulation aisée des booléens par le logiciel – permettent de mettre sur chaque point (comme une pile booléenne) les résultats pour tous les effectifs de classe, sous la forme d'une combinaison linéaire de booléens écrite par le script et mise à jour à chaque tirage.



Ci-dessous, l'ordonnée d'un point pour des échantillons de 50, pris en cours d'exécution du script⁴⁷ :

$$(n=20)*23/50+(n=21)*24/50+(n=22)*26/50+(n=23)*27/50+(n=24)*30/50+(n=25)*33/50+(n=26)*34/50+(n=27)*35/50+(n=28)*36/50+(n=29)*36/50+(n=30)*37/50+(n=31)*38/50+(n=32)*39/50+(n=33)*39/50+(n=34)*39/50+(n=35)*42/50+(n=36)*43/50$$

Le balayage du nombre d'élèves par classe au curseur ne fait que donner à cette expression la valeur correspondante. Le point est alors affiché en temps réel, toutes les données étant déjà dans ses coordonnées. On peut tout de suite observer l'effet des effectifs sur la fluctuation des échantillonnages et le positionnement autour de la valeur théorique.



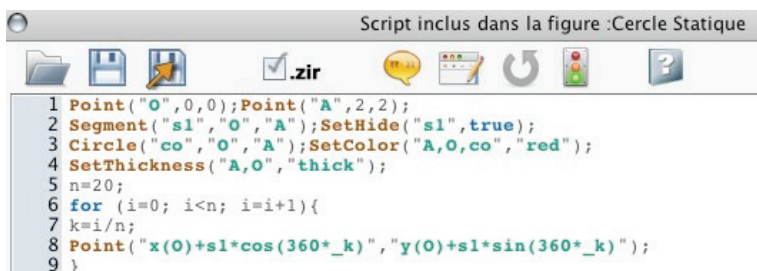
⁴⁷ Le script est détaillé sur cette page :

<http://www.reunion.iufm.fr/recherche/irem/spip.php?article326>

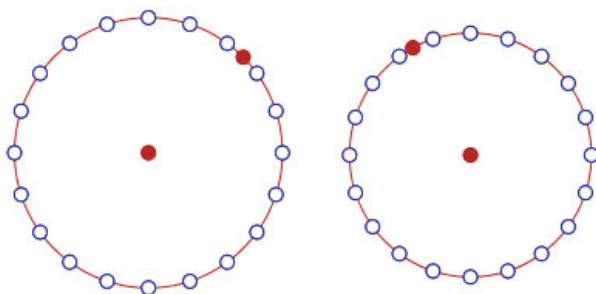
6. Statique/Absolu : un subtil mélange de statut

Nous revenons dans ce paragraphe sur la géométrie repérée dynamique. Nous allons ajouter un peu de géométrie à la démarche analytique mais le propos essentiel restera centré sur les représentations liées à la géométrie repérée. Le matériel de base va être le cercle trigonométrique, et l'enjeu sera la perception du statique et du dynamique qu'il y a dans les scripts que nous allons écrire. On pourrait penser que c'est une problématique – et une expertise – de géométrie dynamique, mais en réalité, comme nous l'avons vu au début de cette partie, la géométrie dynamique va servir, du côté de l'élève, d'outil d'auto-validation pour interroger et faire progresser les représentations de la programmation, voire éventuellement de l'algorithmique. Plusieurs ingénieries peuvent être construites avec les scripts proposés ici, selon le niveau de la classe, la pratique de la programmation. Nous ne proposons que des pistes de réflexion pour que chacun construise ses propres séquences : la pédagogie est elle aussi un micromonde dans lequel la didactique – comme ici – propose ses outils avec lesquels il faut construire.

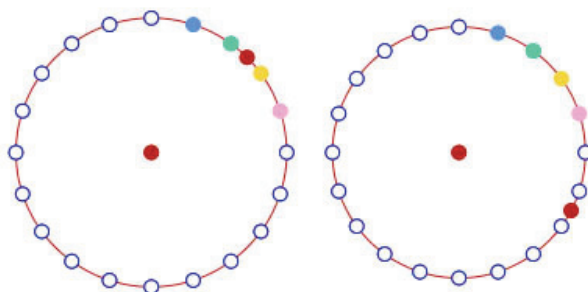
Commençons par ce script, *éventuellement que l'on propose à l'analyse en classe*, puisque c'est une des compétences mises en valeur par les programmes, sur les trois niveaux du lycée. Et ici, c'est bien dans l'analyse que le discours va être intéressant, que les représentations sur les implicites des repères – et les repères implicites – vont devoir s'exprimer.



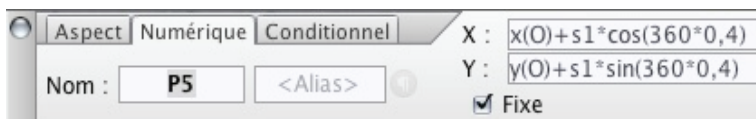
Clairement ce script construit un cercle de centre O passant par A (points gras) et sur ce cercle place 20 points. Divers commentaires techniques peuvent être possibles, en particulier on notera que l'on a construit le segment [OA], en lui donnant un nom dans CaRMetal, *s1* ce qui permet d'utiliser *s1* comme rayon du cercle dans la ligne 8 et éviter un calcul de distance. On peut aussi parler de la référence explicite au point O, et donc de la dépendance du cercle à ce point, ce que l'on voit ci-dessous.



Pendant le déplacement de O, les plus curieux des élèves peuvent se poser des questions, sur le mouvement des autres points. Ils paraissent, à réduction près du cercle, se déplacer en translation. Qu'en est-il en fait ? Avant de déplacer le point A, et pour voir le mouvement des points, on peut en colorier quelques-uns, ce qui donne :



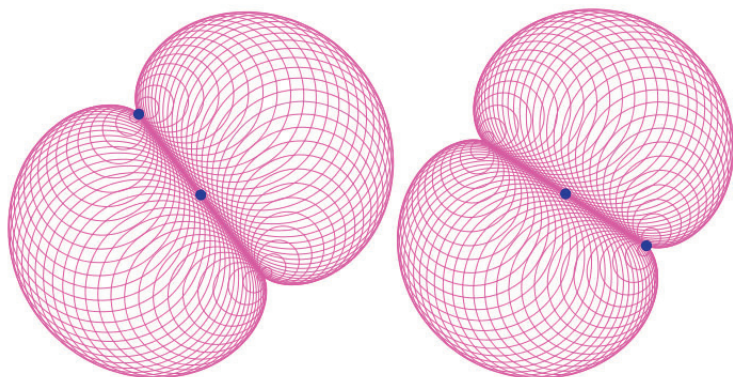
Manifestement les points, s'ils se déplacent à l'écran quand on manipule O ou A, restent figés sur le cercle de centre O, ils ne se déplacent pas sur ce cercle. *Ces points, construits géométriquement, conservent un comportement statique sur le cercle.* Selon la classe, selon si c'est une évidence ou non, un questionnement sur ce sujet peut être lancé. La raison en est évidemment élémentaire, et on peut confirmer sur les coordonnées des points que les angles utilisés sont effectivement des constantes et donc *absolus* :



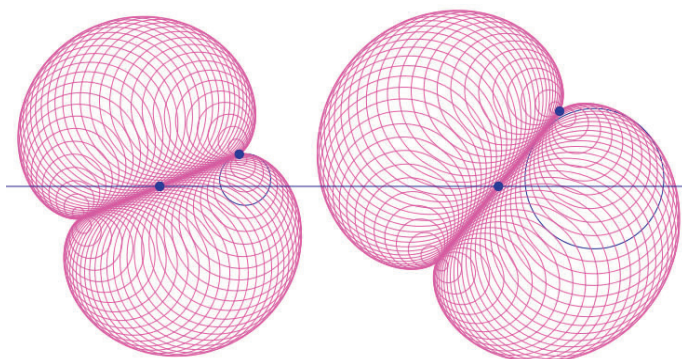
Pour avoir encadré plusieurs formations sur les CarScripts, un invariant émerge à cet endroit : ***la représentation que l'on a du dynamique est questionnée***. Elle l'est déjà sur l'articulation entre le statique et l'absolu. En effet, jusqu'ici le statique – la constante – était identifié au dessin que la manipulation directe cassait (les tableaux de fils statiques) alors qu'ici l'absolu, s'il contient une dimension statique, s'inscrit aussi dans une logique dynamique. Elle l'est d'autant que, dans de nombreuses situations, l'usage des angles absolus peut suffire. Voyons en un exemple.

Un absolu pas si statique qu'on pourrait le penser

En fait le script précédent est une partie d'un travail plus général. Le support de ce questionnement était la construction d'une néphroïde comme enveloppe de cercles. On se donne un cercle de diamètre $[AB]$ et à partir d'un point M du cercle (les points du script précédent) on construit le cercle de centre M tangent à $[AB]$. On obtient alors une néphroïde. Si on s'arrange pour la construire à partir de deux points, le centre O du cercle et un point A qui définit ce cercle, elle est manipulable facilement. Et elle fait totalement illusion – au sens où l'on oublie l'aspect absolu des angles – car en manipulant A , même avec ces angles absolus, le cercle de centre un point donné bouge avec A puisque le diamètre $[AB]$ bouge aussi.



Mais si, dans un contexte géométrique donné, on devait répondre à une contrainte dynamique spécifique, comme utiliser un cercle particulier en fonction de A – le quatrième cercle avant le point A dans l'illustration ci-dessous – clairement la figure ne convient pas, c'est-à-dire *l'algorithme ne convient pas*, les angles absolus ne suffisent plus.



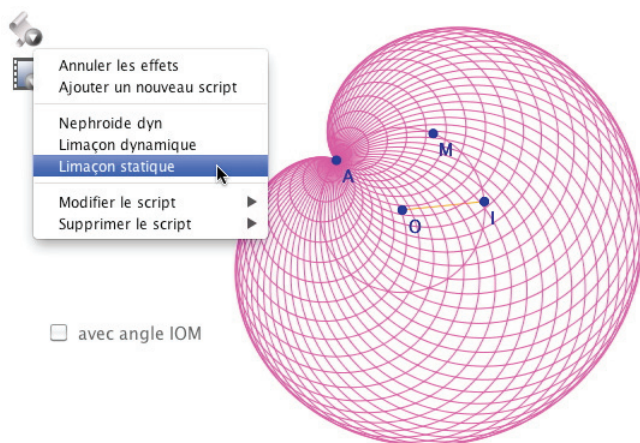
À gauche le quatrième cercle avant A est mis en bleu, à droite on voit que c'est le cercle de centre d'angle nul dans le script.

```

1 // Nephroïde
2 Point("O",0,0);Point("A",2,1);SetThickness("A,O","thick");
3 Segment("s1","O","A");Line("d","O","A");
4 Circle("co","O","A");SetHide("s1,d,co",true);
5 n=80;
6 for (i=0; i<n; i++){
7   k=i/n;
8   pi=Point("x(O)+s1*cos(360*_k)","y(O)+s1*sin(360*_k)");
9   hi=Perpendicular("d",pi);SetHide(hi,true);
10  mi=Intersection("d",hi);SetHide(mi,true);
11  Circle(pi,mi);SetHide(pi,true);
12 }

```

Cardioïde version formation (dont utilisation des booléens)



Sur cette figure, une animation de M laisse la cardioïde statique si la case n'est pas cochée, mais la cardioïde tourne avec M si la case est cochée. Le script doit inclure, dans l'angle, une constante (l'angle absolu) et une variable (le complément dynamique) en fonction de la case à cocher. C'est une occasion pour utiliser la multiplication par un booléen.

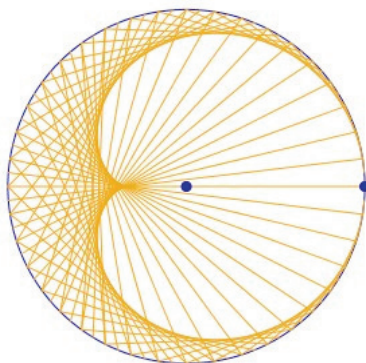
```

1 // on est sur un cercle centre O passant par I, M et A sont deux points
2 // de ce cercle. a est la variable associée à une boîte à cocher et
3 // al est l'angle (sur 360°) IOM
4
5 Segment("s1", "O", "I"); SetHide("s1", true);
6 n=50;
7 for (i=0; i<n; i++){
8   k=i/n
9   ci=Point("x(O)+s1*cos(al*(a==1)+360*_k)", "y(O)+s1*sin(al*(a==1)+360*_k)");
10  SetHide(ci, true);
11  Circle(ci, "A");
12 }

```

Une version scolaire du questionnement absolu/statique

L'exemple précédent a été proposé pour mettre en valeur cette relation entre statique/absolu/dynamique, mais dans registre flou pour que cela n'apparaisse qu'après la construction, en fonction de l'usage que l'on fera de la figure. En classe, pour provoquer ce questionnement, on choisirait une situation sans ambiguïté. Se plaçant dans un contexte où l'on n'a pas abordé la question du script introductif sur le cercle pour découvrir la problématique, on peut travailler avec cet exemple : il s'agit de réaliser une cardioïde par enveloppe de segments ; on place sur le cercle n points avec un angle au centre constant entre deux points consécutifs et, en parcourant deux fois le cercle, on joint les points d'indice k et $2k$.



Alors, avec ce script...

```

1 Point("O",0,0);Point("A",3,0);
2 Segment("s1","O","A");SetHide("s1",true);
3 Circle("co","O","A");SetColor("co","blue");
4 SetThickness("A,O","thick");
5 n=80;
6 for (i=0; i<n; i++){
7   k=i/n;
8   pi=Point("x(O)+s1*cos(360*_k)","y(O)+s1*sin(360*_k)");
9   pf=Point("x(O)+s1*cos(360*2*_k)","y(O)+s1*sin(360*2*_k)");
10  SetHide(pi,true); SetHide(pf,true);
11  Segment(pi,pf);
12 }

```

... il est clair qu'en déplaçant A, la cardioïde ne va pas suivre. Et pourtant... c'est souvent une surprise, et c'est elle qui fait surgir la question de l'angle absolu et fait ressortir des représentations de cette situation, soit sur les angles, soit tout simplement sur *les nombres réels* car après tout, utilise-t-on des angles ou des réels ici ?

Ce questionnement révèle l'un des sens de l'interrogation préalable repéré sur « notre représentation du dynamique ». C'est aussi, en fait, un questionnement sur le statut des variables. La « cardioïde de formation » a été proposée pour discuter de cette question et éclaircir la situation (avec de plus un visuel très convaincant). Les variables de CaRMetal ont toujours un comportement dynamique, celle du JS, passées algébriquement dans des objets sont dynamiques, mais passées par contenu, deviennent des nombres, et ont un comportement statique.

En définitive, à ce stade, on a envie de dire tout simplement⁴⁸ : le *statique* est du registre numérique, le *dynamique* du registre algébrique. Pour éclaircir cela, on peut proposer aux élèves la contrainte suivante : faire que la cardioïde construite par des segments suive le point A, plus précisément qu'elle soit de *sommet* le point A. Il n'y a pas grand-chose à modifier bien entendu mais cela peut être complexe pour les élèves car, pour faire cela, il est plus simple de se placer dans un autre contexte (que l'on peut préciser), celui où les objets sont déjà sur la figure.

On se donne donc un cercle de centre O passant par I, un point M sur objet du cercle. On note *ang*, l'angle (OI, OM)⁴⁹. De fait, on s'est donné une origine des angles. Alors ce script :

⁴⁸ Sans oublier le comportement affine : un point au tiers d'un segment n'est pas du registre numérique, car la manipulation directe respectant les relations affines, elle lui donne aussi, dans son contexte de point sur objet, un statut dynamique.

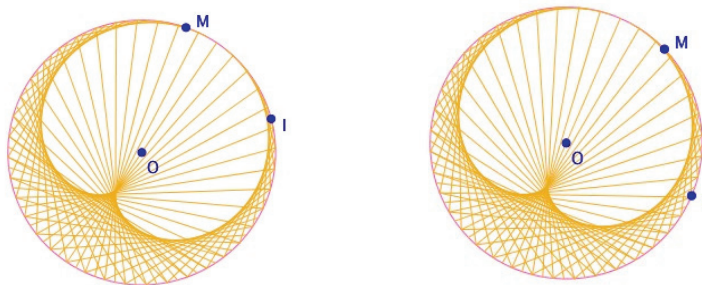
⁴⁹ Dans l'inspecteur d'objets, il faut donner à la mesure de *ang* une amplitude jusqu'à 360°.

```

1 n=80;
2 for (i=0; i<n; i++){
3   k=i/n;
4   pi=Point("x(0)+s1*cos(ang+360*_k)", "y(0)+s1*sin(ang+360*_k)");
5   pf=Point("x(0)+s1*cos(ang+360*2*_k)", "y(0)+s1*sin(ang+360*2*_k)");
6   SetHide(pi,true); SetHide(pf,true);
7   Segment(pi,pf);
8 }

```

donne bien ce qui est attendu. En particulier une animation sur M produit bien l'effet souhaité d'une rotation de la cardioïde (à gauche sur l'illustration suivante).



Une erreur de manipulation

Malencontreusement – mais plus certainement malicieusement – un élève⁵⁰ a fait une animation sur le point I (illustration de droite ci-dessus), il demande – avec un vrai plaisir de déstabilisation potentielle – qu'il « ne comprend pas ce qu'il se passe, M n'est plus là où il faut, et pourtant, on a bien mis l'angle IOM cette fois ». Encore une belle illustration que, définitivement, l'absolu des angles est toujours présent, caché mais présent...

Bilan : premières pratiques de trigonométrie dynamique ?

Dans un registre différent de la (simple) géométrie repérée dynamique, plus délicat en terme de concepts mathématiques engagés (le cercle trigonométrique et la mesure des angles), nous commençons à dégager cette hypothèse : *l'interaction entre la géométrie dynamique et la programmation, par la manipulation directe sur les objets produits, permet d'enrichir les deux cadres de travail.* Par ailleurs, à travers des problèmes concrets (on travaille sur des objets, produits par les élèves), on manipule, sans grande technicité,

⁵⁰ Un stagiaire en fait, et nous étions bien dans la seconde option.

les outils usuels de la géométrie analytique élémentaire et de la trigonométrie en posant des questions qui relèvent de la culture des élèves (quel type de manipulation directe sur les images suis-je en train de construire ?) tout en revisitant, de manière originale, les concepts mathématiques engagés.

VIII. Aller (un peu) plus loin avec les scripts

1. Utilisation pour réaliser des figures 3D

Les scripts sont aussi une opportunité pour faire rapidement des figures complexes dans l'espace. C'est à Jérôme Caré, régulièrement présent sur les forums de CaRMetal, que l'on doit l'idée d'utiliser directement les points du repère 3D sachant qu'un point $M(x, y, z)$ dans le repère (O, X, Y, Z) de la feuille d'espace est représenté dans le plan de la figure par le point :

$$M_e(x_O + x_M(x_X - x_O) + y_M(x_Y - x_O) + z_M(x_Z - x_O), \\ y_O + x_M(y_X - y_O) + y_M(y_Y - y_O) + z_M(y_Z - y_O))$$

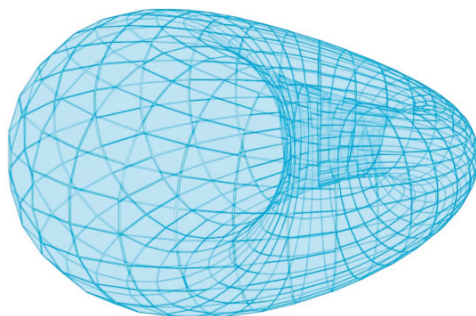
On utilise alors simplement une fonction⁵¹ générique pour les scripts comme celle-ci :

```
5 function TracePoint(Nom,ux,uy,uz){
6   vx=ux;vy=uy;vz=uz;
   up=Point(Nom,"
   x(O)+(x(X)-x(O))*_vx+(x(Y)-x(O))*_vy+(x(Z)-x(O))*_vz",
   y(O)+(y(X)-y(O))*_vx+(y(Y)-y(O))*_vy+(y(Z)-y(O))*_vz");
7 }
```

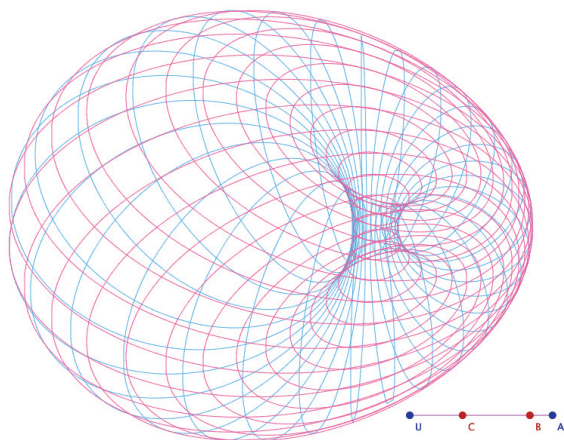
On arrive alors facilement à produire des figures 3D, *statiques* au sens de la partie précédente, c'est-à-dire qu'on ne manipule que par le trièdre, comme cette cyclide de Dupin⁵² :

⁵¹ C'est parce que les variables sont passées en adressage indirect à CaRMetal qu'il faut les recopier dans la fonction. On remarquera que dans le script de Cesaro, il n'y a pas de copie car les variables sont utilisées de manière standard, comme simples variables JavaScript. Cette présentation de la fonction *TracePoint* est à caractère didactique pour mettre en évidence – en formation des enseignants – la place des coordonnées 3D du point dans le calcul des coordonnées 2D. Une fois cette présentation faite, une autre option plus efficace consiste à tronquer la chaîne de caractères pour utiliser directement la variable JS *ux, uy, uz* sans avoir à les recopier localement, ce qui donne : “*x(O) + (x(X) - x(O))*” + ux + “+ x(X) - x(O))...*”.

⁵² <http://www.reunion.iufm.fr/recherche/irem/spip.php?article210>



La figure précédente est construite avec plus de 960 quadrilatères. On peut y ajouter des paramètres et rendre le script bien plus rapide en choisissant de tracer des ellipses à la place des polygones. Le partage en 31 parties, comme dans le cas suivant, n'utilise que 62 coniques. Les points B et C définissent les rayons des sphères intervenant dans la cyclide.



Mais ce n'est pas véritablement de la manipulation directe sur la figure comme on l'entend usuellement. Pour cela, comme le logiciel n'est pas un logiciel de 3D⁵³, on ne peut travailler que dans le cadre de relations entre les

⁵³ On voit bien qu'une gestion des coordonnées 3D permettrait une manipulation directe 3D réellement significative. Cela va être possible par script : dans la partie suivante nous allons récupérer les véritables coordonnées 3D des points à partir de leurs coordonnées 2D à l'écran et donc construire une injection de $\mathbf{R}^3$ sur $\mathbf{R}^2$!

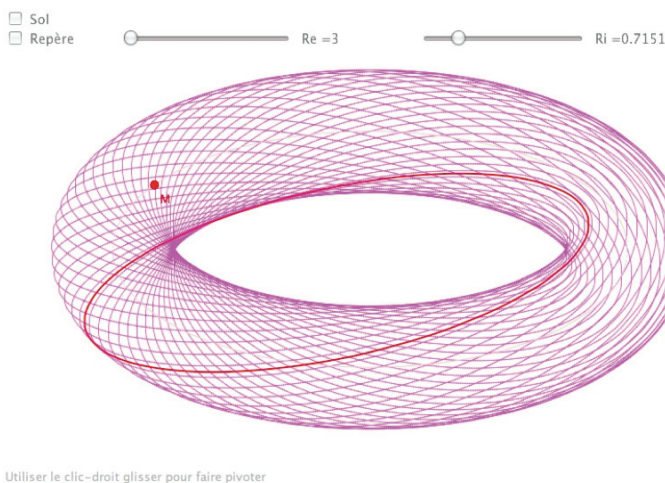
coordonnées, typiquement des relations linéaires, dans un plan. Pour donner un exemple, voyons comment manipuler un cercle à partir d'un point du sol. Pour cela il faut récupérer les coordonnées 2D dans le plan (XOY) du point M. Dans ce cas précis, il serait très simple d'utiliser un parallélisme aux axes et d'utiliser des points intermédiaires pour déterminer ces coordonnées.

Mais pour montrer que la démarche peut être plus générale, exprimons les coordonnées planes (x_p, y_p) d'un point M du plan du sol dont on a les simples coordonnées (x_M, y_M) dans le plan de la feuille. Il s'agit de résoudre un système d'ordre 2, et ces coordonnées sont, dans une notation utilisant les fonctions de CaRMetal x et y :

$$x_p = \frac{((y(M) - y(O))(x(Y) - x(O)) - (x(M) - x(O))(y(Y) - y(O)))}{(x(Y) - x(O))(y(X) - y(O)) - (x(X) - x(O))(y(Y) - y(O))}$$

$$y_p = \frac{(-(y(M) - y(O))(x(X) - x(O)) + (x(M) - x(O))(y(X) - y(O)))}{(y(X) - x(O))(y(X) - y(O)) - (x(X) - x(O))(y(Y) - y(O))}$$

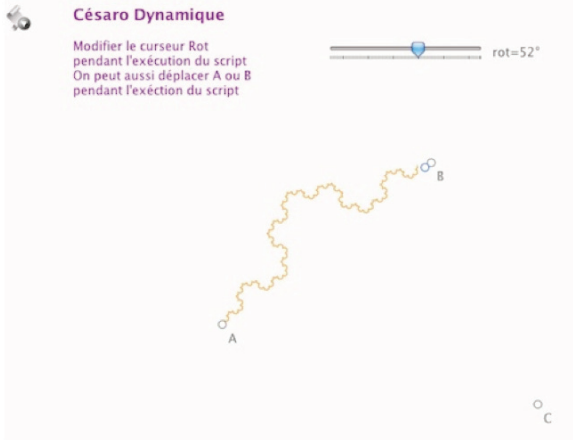
On peut alors entreprendre des figures de l'espace, entièrement construites par script, qui ont un degré de manipulation directe *sur la figure*. Par exemple, voici un tore⁵⁴ construit par 40 cercles de Villarceau, de rayon paramétrable, avec l'un d'eux accessible à la manipulation directe par le point M.



⁵⁴ Script et figure manipulable en ligne à l'adresse de la note 52.

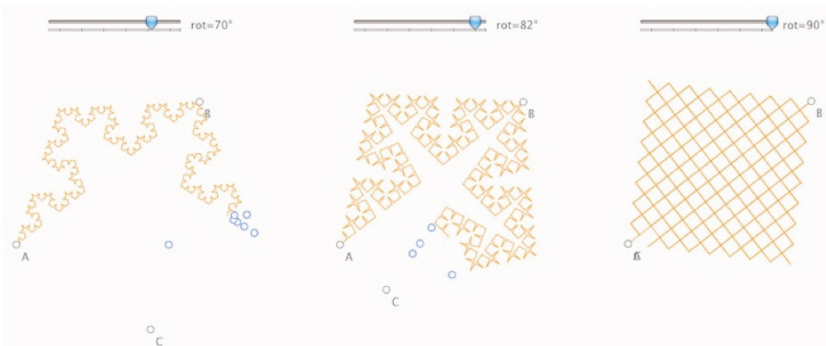
Poursuivons par deux scripts qui illustrent de manière plus profonde l'intégration réalisée au sein de ce logiciel entre le JavaScript et les outils internes à CaRMetal.

2. La récursivité dynamique (sur la courbe de Cesaro)

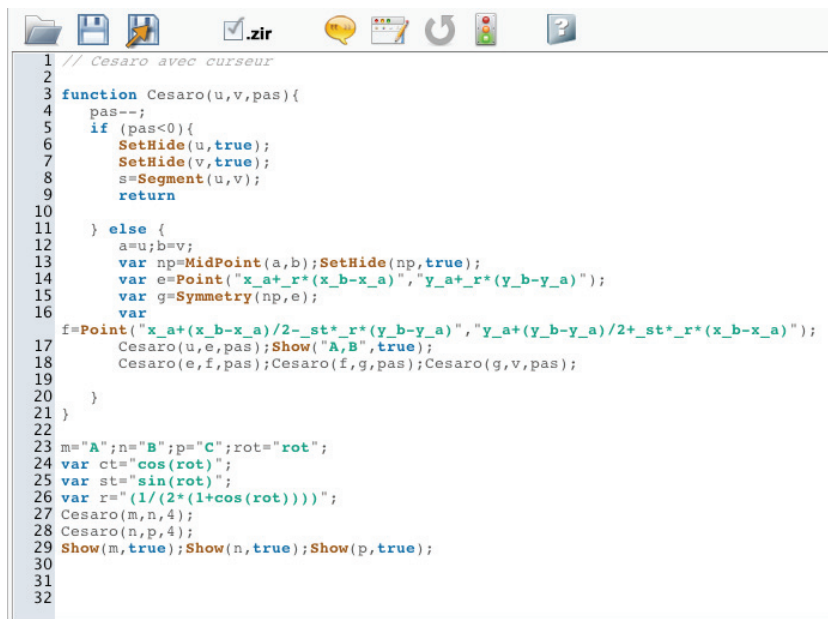


La courbe de Cesaro généralise celle de Koch avec un angle variable

La construction est récursive et la manipulation directe sur l'angle de la courbe de Cesaro est disponible *pendant l'exécution du script*. L'effet est alors saisissant puisque tout est modifié en temps réel avec le nouvel angle. En voici des illustrations prises en cours d'exécution.



*Manipulation directe sur l'angle de la courbe de Cesaro
pendant l'exécution du script*



```

1 // Cesaro avec curseur
2
3 function Cesaro(u,v,pas){
4   pas--;
5   if (pas<0){
6     SetHide(u,true);
7     SetHide(v,true);
8     s=Segment(u,v);
9     return
10  } else {
11    a=u;b=v;
12    var np=MidPoint(a,b);SetHide(np,true);
13    var e=Point("x_a+_r*(x_b-x_a)","y_a+_r*(y_b-y_a)");
14    var g=Symmetry(np,e);
15    var
16    f=Point("x_a+(x_b-x_a)/2+_st*_r*(y_b-y_a)","y_a+(y_b-y_a)/2+_st*_r*(x_b-x_a)");
17    Cesaro(u,e,pas);Show("A,B",true);
18    Cesaro(e,f,pas);Cesaro(f,g,pas);Cesaro(g,v,pas);
19  }
20 }
21
22
23 m="A";n="B";p="C";rot="rot";
24 var ct="cos(rot)";
25 var st="sin(rot)";
26 var r="(1/(2*(1+cos(rot))))";
27 Cesaro(m,n,4);
28 Cesaro(n,p,4);
29 Show(m,true);Show(n,true);Show(p,true);
30
31
32

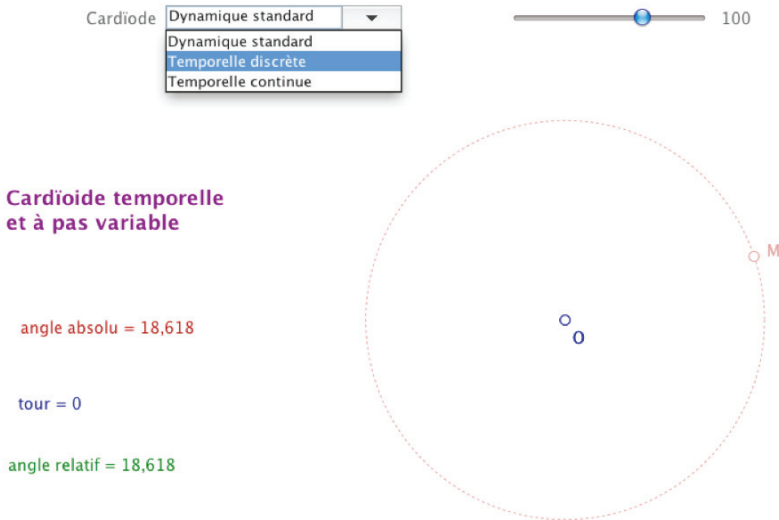
```

C'est un script de figure, il utilise les points A, B et C et l'angle *rot* du curseur. Dans la figure C est le transformé de A dans la rotation de centre B et d'angle $\pi - 2*rot$.

3. Animation, script et temporalité

Retour à la question de la temporalité dans les figures : qu'en est-il des scripts et de la gestion du temps ? La nouvelle animation de la version 3.5 est-elle capable de respecter cette temporalité ?

Pour observer cela, nous poursuivons sur la cardioïde par segments déjà rendue dynamique dans la partie précédente. Nous allons y ajouter, d'une part, un pas variable en temps réel (*i. e.* il y aura de 60 à 120 segments), mais surtout une rotation de la cardioïde qui pourra prendre trois aspects : standard, temporel discret, temporel continu.



La figure avant le script, est déjà construite dans un environnement temporel

On a utilisé la macro-construction *Curseur circulaire* pour construire M. C'est un point qui s'enroule sur le cercle avec son compte-tours. La macro renvoie l'angle absolu et le nombre de tours. Les menus pop-up et le curseur sont inactifs, ce sont des données qui seront utilisées par le script suivant :

Script inclus dans la figure :CardioTempetVar

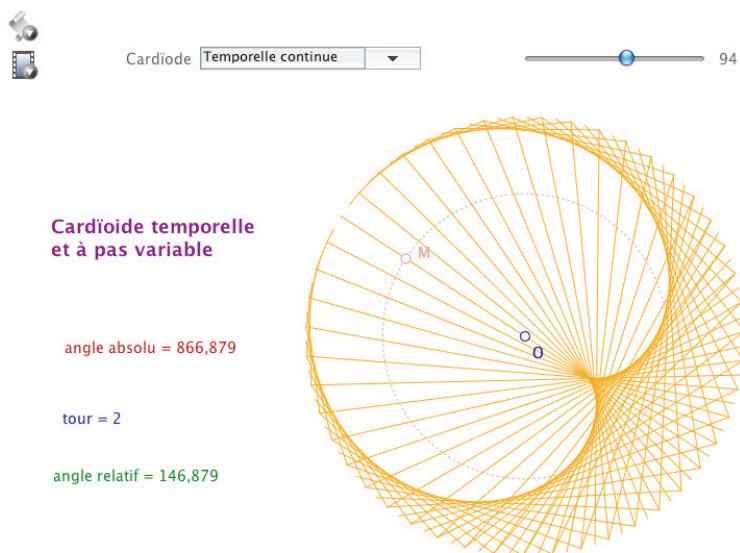
```

1 n=120;
2 for (i=0; i<n; i++){
3   pi=Point("x(0)+(sA+(chx>1)*tour+(chx==3)*anrel/360)*cos(angl+360*_i/nseg)", "
4   y(0)+(sA+(chx>1)*tour+(chx==3)*anrel/360)*sin(angl+360*_i/nseg)");
5   pf=Point("x(0)+(sA+(chx>1)*tour+(chx==3)*anrel/360)*cos(angl+360*2*_i/nseg)", "
6   y(0)+(sA+(chx>1)*tour+(chx==3)*anrel/360)*sin(angl+360*2*_i/nseg)");
7   SetHide(pi,true); SetHide(pf,true);
8   sg=Segment(pi,pf);
9   Conditional(sg,"hidden", "_i>nseg");
10 }

```

Le script construit le segment [pi pf]. L'écriture paraît complexe mais c'est seulement parce que le rayon du cercle est un peu long à écrire et qu'il est utilisé quatre fois. *sA* est le rayon du cercle initial, *nseg* est le nombre de segments, donné par le curseur. On remarquera que la boucle va jusqu'à 120, tous les segments sont construits, même si *nseg* est plus petit. Leur affichage

est traité dans la ligne 5, la dernière ligne de la boucle, le segment est caché si son indice est trop élevé. Ainsi les points ne sont calculés qu'une fois, seul leur affichage dépend du curseur. La variable chx est bien sûr le choix du popup. Le rayon du cercle par défaut vaut sA . On lui ajoute une expression si $chx > 1$ (donc si on fait un choix de cardioïde temporelle). Dans ce cas on lui ajoute le nombre de tours, c'est le sens de $(chx > 1) * tour$. Si de plus on a fait le choix *temporel continu*, il faut ajouter une variation continue du rayon : c'est l'expression $(chx == 3) * anrell / 360$. On remarquera aussi l'économie considérable du traitement par les booléens qui évitent le recours à des conditionnels imbriqués qui seraient difficiles à mettre en œuvre dans ce cas.



La cardioïde après deux tours, en mode animation...

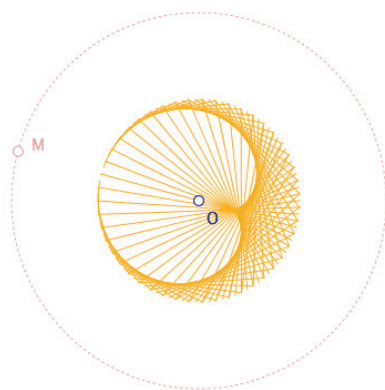
Bien entendu, pendant l'animation, on peut faire sauter la cardioïde d'un mode à l'autre en changeant de choix dans le pop up et faire varier le nombre de segments. Une dernière expérience illustrera, une nouvelle fois, la profondeur de l'intégration des outils entre eux, et donc la sensibilité de l'auteur à la recherche de cohérence du logiciel utilisé comme EIAH. Si avant de lancer l'animation sur M, on fait faire trois tours en arrière à M, en mode temporel, et qu'on lance l'animation, celle-ci démarre bien dans l'état temporel de la cardioïde. Il en résulte que *l'animation de CaRMetal respecte la temporalité et le déterminisme enrichi induit*.

Cardioïde temporelle et à pas variable

angle absolu = -915,21

tour = -3

angle relatif = 164,79



... puis en inversant le sens de l'animation (ici à 60 segments)

IX. Aller (nettement) plus loin avec les scripts.

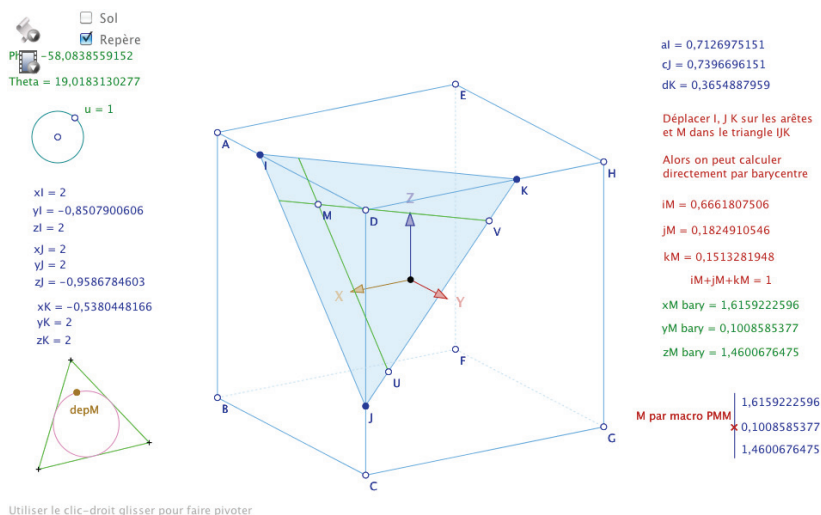
Pour terminer cette présentation des CarScripts et du logiciel, nous abordons des travaux récents (juillet 2010) de deux collègues, Jérôme Caré et Pierre-Marc Mazat. Ces travaux ouvrent de nouvelles perspectives pour l'utilisation du logiciel en classe. Et comme tout travail novateur, ils posent aussi de nouvelles questions. Les deux thèmes présentés ici ont cette particularité d'avoir été initiés par Jérôme Caré, et retravaillés par Pierre-Marc Mazat pour leurs aspects ergonomiques.

1. Retrouver les coordonnées 3D d'un point à l'écran

Le logiciel n'étant pas un logiciel 3D, les points ne sont pas stockés avec leurs coordonnées 3D. On peut naturellement penser qu'il n'y a aucune raison de rechercher les coordonnées 3D d'un point d'une figure : ce serait, selon le point de vue, construire une injection de $\mathbf{R}^3$ sur $\mathbf{R}^2$ (les points 3D depuis leurs représentations 2D) ou une surjection de $\mathbf{R}^2$ vers $\mathbf{R}^3$ (les coordonnées 2D décrivant tout l'espace 3D). Nous sommes nombreux à avoir déjà argumenté que ce n'est même pas la peine d'y réfléchir.

Mais cela n'était pas l'avis de Jérôme Caré, car les arguments des mathématiques statiques ne s'appliquent peut-être pas en géométrie *dynamique* quand ce dynamisme est à l'œuvre. Il a remarqué (en octobre 2009) que la

rotation du trièdre de CaRMetal permettait d'obtenir une bonne approximation des coordonnées des points placés dans le trièdre. Cela avait abouti à une première macro *xyr.mcr* diffusée avec un fichier explicatif. Toutefois, cette macro, qui donnait théoriquement des valeurs exactes, était peu pratique à l'utilisation et surtout ne donnait des résultats qu'avec une précision de l'ordre de 10^{-2} , ce qui n'était pas suffisant pour une utilisation régulière. Un travail sur ce thème, avec une autre approche, fut relancé en juin 2010 par Pierre-Marc Mazat et, après de nombreux échanges de courriels pour affiner les démarches de chacun, une contribution fut publiée⁵⁵ par l'initiateur de cette *injection dynamique* de $\mathbf{R}^3$ sur $\mathbf{R}^2$ et enrichie par une procédure plus complète – mais qui demande un aménagement des figures – par le second auteur. C'est cette seconde procédure que nous allons rapidement présenter.



Dans cette figure test, M est un point de l'ellipse de Steiner du triangle IJK⁵⁶ obtenu par transformation affine du point *depM* du cercle inscrit d'un triangle équilatéral (procédé géométrique classique). En vert : le calcul des coordonnées de M par barycentre, et en bleu : les valeurs produites par la macro *point3D* utilisant une vibration du trièdre. On peut déplacer I, J, K, *depM*, le calcul de M est aussitôt mis à jour, il coïncide bien avec les calculs.

⁵⁵ <http://db-maths.nuxit.net/CaRMetal/forums/viewtopic.php?t=358>

Le post de J. C. date du 30 juin, l'amélioration de P.-M. M., plusieurs fois affinée, du 31 juillet. Entre-temps les deux auteurs ont travaillé sur la partie suivante...

⁵⁶ Mais cela aurait pu être tout point du triangle par la macro « point sur plan ».

Dans le projet initial, il s'agissait de provoquer une rotation du trièdre (O, X, Y, Z) pour avoir les coordonnées d'un point, démarche suffisante pour le traitement des problèmes de géométrie dans l'espace au lycée, qui sont essentiellement numériques. Pierre-Marc Mazat a voulu aller plus loin, et a cherché à automatiser cette démarche pour arriver à la rendre utilisable en manipulation directe. Si une rotation du trièdre donne les coordonnées au moment de la rotation, pour avoir en permanence les coordonnées en manipulation directe, il faut faire tourner le trièdre en permanence... sans que cela soit perceptible. Le principe est donc de faire vibrer le trièdre en ϕ et en θ , d'une faible quantité ($0,01^\circ$) pour que cela reste imperceptible à l'œil, et alternativement positivement et négativement pour que le trièdre ne tourne pas.

Les premiers essais dans une boucle *while (true) ... end while* ne donnent pas satisfaction en terme d'ergonomie. La manipulation directe pendant l'exécution d'un script, bien qu'elle soit possible – on l'a vu avec Cesaro – n'est pas aussi immédiate que directement sur la figure. L'idée est alors d'arriver à exporter dans la figure la vibration du trièdre et de n'utiliser le script que pour placer les expressions associées dans la figure, ces expressions n'étant évaluées qu'avec la vibration du repère. Par ailleurs, l'usage du script permet d'automatiser plus finement toute une partie de la figure.

```

30 Expression("y1"+p,
  "(-(sin(E10)-sin(old(E10)))*(y("p+")-old(y("p+"))))/(-(sin(E10)-sin(
  old(E10)))^2*sin(E11)-(cos(E10)-cos(old(E10)))^2*sin(E11))-((cos(E10)-cos
  (old(E10)))*sin(E11)*(x("p+")-old(x("p+"))))/(-(sin(E10)-sin(old(E10))
  )^2*sin(E11)-(cos(E10)-cos(old(E10)))^2*sin(E11)))*(x("p+")==x(0))",
  "x(_q)/pixel", "y(_q)-4/pixel");
31 Expression("y"+p, 0, "x(_q)+7/pixel", "y(_q)-4/pixel");
32 SetExpressionValue("y"+p, "if(d(Rot)==0,0,if(u==2,y1"+p+",y"+p+"))");
33
34 Expression("z1"+p,
  "(((sin(old(E10))*(sin(E10)-sin(old(E10)))*(sin(E11)-sin(old(E11)))+cos(
  old(E10))*(cos(E10)-cos(old(E10)))*(sin(E11)-sin(old(E11)))*(y("p+")-o
  ld(y("p+"))))/((cos(E11)-cos(old(E11)))*(-(sin(E10)-sin(old(E10)))^2*si
  n(E11)-(cos(E10)-cos(old(E10)))^2*sin(E11))+old(y("p+"))-old(old(y("
  p+"))))/((cos(E11)-cos(old(E11)))+(sin(old(E10))*(cos(E10)-cos(old(E10))
  )*sin(E11)*(sin(E11)-sin(old(E11)))-cos(old(E10))*(sin(E10)-sin(old(E10)
  ))*sin(E11)*(sin(E11)-sin(old(E11)))*(x("p+")-old(x("p+"))))/((cos(E1
  1)-cos(old(E11)))*(-(sin(E10)-sin(old(E10)))^2*sin(E11)-(cos(E10)-cos(ol
  d(E10)))^2*sin(E11)))*(!x("p+")==x(0)+(x("p+")==x(0))*(y("p+")-y(0)
  )/(y(Z)-y(0))", "x(_q)+7/pixel", "y(_q)-30/pixel");
35 Expression("z"+p, 0, "x(_q)+7/pixel", "y(_q)-30/pixel");
36 SetExpressionValue("z"+p, "if(d(Rot)==0,0,if(u==2,z1"+p+",z"+p+"))");

```

E10 et E11 sont respectivement ϕ et θ de la figure précédente, une expression comme $\sin(E10) - \sin(\text{old}(E10))$ permet d'avoir la variation du

sinus de l'angle ϕ . Les calculs⁵⁷ montrent que cela permet d'avoir la valeur exacte de chaque coordonnée à partir d'un mouvement du trièdre avec un seul $\Delta\phi$ et $\Delta\theta$. Il faut prendre des précautions pour les coordonnées sur l'axe Oz comme on le voit à la fin de la ligne 34. La mise en place d'une vibration complexifie la procédure, et nécessite de la séquentialiser avec précision. Il en résulte que l'on a un cycle de trois changements des variables au fur et à mesure que le repère vibre, et que les valeurs calculées ne sont correctes qu'au troisième changement. L'introduction de la variable u – présente dans la figure – qui prend successivement les valeurs 0, 1, 2, 0, 1... permet de n'afficher les valeurs correctes que lorsque le repère a effectué une vibration complète ($u = 2$) depuis la situation initiale ($u = 0$).

Voici un tableau récapitulant les différents stades :

Valeurs de u	Valeurs des angles	Expressions CaRMetal associées aux angles	Valeurs des coordonnées	Expressions CaRMetal associées aux coordonnées
0	$(\phi ; \theta)$	$(\text{old}(E10) ; \text{old}(E11))$	$(x ; y)$	$(\text{old}(x) ; \text{old}(\text{old}(y)))$ ⁵⁸
1	$(\phi ; \theta + \Delta\theta)$	$(\text{old}(E10) ; E11)$	$(x ; y + \varepsilon)$	$(\text{old}(x) ; \text{old}(y))$
2	$(\phi + \Delta\phi ; \theta + \Delta\theta)$	$(E10 ; E11)$	$(x + \delta ; y + \varepsilon + z)$	$(x ; y)$

Le script affiche alors l'expression calculée (de x , y en ligne 32, et z en ligne 36) seulement si u vaut 2 et sinon garde la valeur précédente (et donc l'expression générale est de plus récursive⁵⁹), ce qui évite d'afficher les valeurs intermédiaires erronées qui sont cachées dans des variables x_1 , y_1 et z_1 dont l'existence sert à éviter une récursivité sur des expressions aussi longues à interpréter que la ligne 34.

Le choix d'utiliser un script permet d'automatiser plusieurs tâches comme la création de l'animation nécessaire à la vibration du trièdre et de la variable u si elles n'existent pas dans la figure, ou encore d'effectuer quelques considérations ergonomiques : les coordonnées prennent le nom du point cliqué (d'où le " z " + p en ligne 35 par exemple), ce qui est très pratique pour une utilisation ultérieure comme ci-dessous. Pourtant, comme on le voit, le

⁵⁷ Non reproduits ici, ils seront ultérieurement publiés dans un numéro de *MathémaTICE* (n° 21 ou 22).

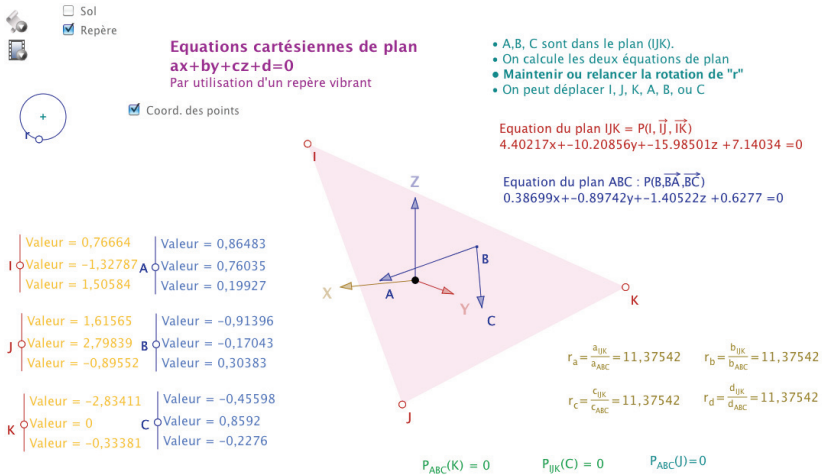
⁵⁸ Voir à la fin de la 5^e ligne de la ligne 34.

⁵⁹ Ce qui l'oblige à être initialisée indépendamment – par $d(Rot)=0$ – car, comme l'a déjà vu et contrairement aux points, la création des expressions ne les initialise pas pour la récursivité.

script ne sert qu'à la création et à l'initialisation des variables, ensuite tous les rafraichissements ont entièrement lieu dans la figure.

On retiendra de ce paragraphe qu'une rotation du trièdre en géométrie dynamique permet d'avoir les coordonnées réelles d'un point 3D depuis sa projection en perspective et qu'une vibration permanente du trièdre permet d'avoir les coordonnées des points en manipulation directe, et donc de faire de la géométrie 3D dans un logiciel dit de 2D+. Bien entendu, ces points doivent d'une façon ou d'une autre – par construction ou comme point sur objet (y compris de lieu ou de plan) – être liés au repère 3D de la figure.

Voici un exemple d'utilisation en cours, sur la géométrie repérée dans l'espace et les équations de plan. On dispose d'un plan I, J, K, et dans ce plan de trois points A, B, C. On calcule une équation cartésienne du plan passant par I et de vecteurs directeurs $\vec{IJ}$ et $\vec{IK}$, puis on refait de même avec le plan passant par B et de vecteurs directeurs $\vec{BA}$ et $\vec{BC}$. On peut alors comparer les coefficients des deux plans.



Exemple d'utilisation du script de repère vibrant : I, J, K, A, B et C sont en manipulation directe et on dispose de leurs coordonnées 3D en temps réel.

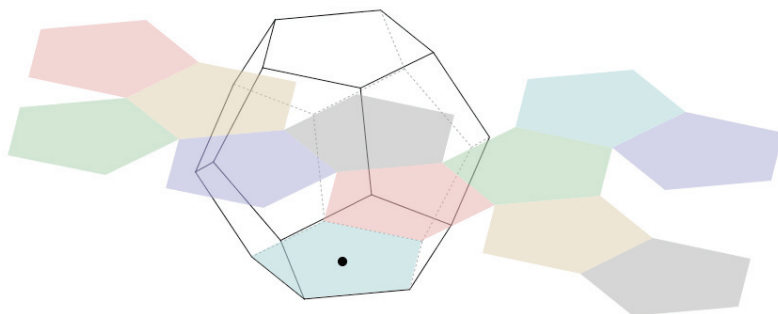
On retiendra aussi que, après 20 ans d'existence, nous avons encore à découvrir et à mettre en œuvre des propriétés assez surprenantes qu'offre la géométrie dynamique. La démarche initiale de Jérôme Caré, presque contre

une évidence mathématique, est un exemple de ce que peut être une profonde perception de la géométrie dynamique.

2. Premiers scripts objets – Patrons dynamiques de polyèdres

Lors de la création des CarScripts, l’auteur a proposé une instruction, *interactiveInput()*, qui permet de saisir des objets au cours d’un script. L’idée était de pouvoir permettre aux scripts d’être aussi de véritables extensions des macro-constructions. Et c’est de belles extensions de macros dont nous allons parler maintenant.

C’est encore à Jérôme Caré que nous devons le premier CarScript objet. Son projet initial était de produire le patron d’un cube, à partir d’un cube, l’utilisateur cliquant sur les arêtes du cube – avec le *interactiveInput()* – pour construire son patron. On pouvait ainsi construire les 11 patrons du cube, à partir d’un seul script⁶⁰. Le lendemain, il proposait la même chose pour le dodécaèdre⁶¹, et désormais chacun peut construire, avec un seul script, en manipulation directe, l’un des 43 380 patrons du dodécaèdre.



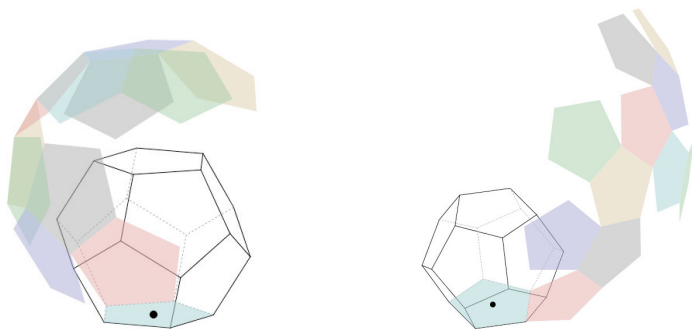
Non seulement on peut choisir les arêtes, mais l’ouverture est accessible pendant la construction – ce qui aide à choisir les arêtes.

⁶⁰ Le script est téléchargeable sur cette contribution :

<http://db-maths.nuxit.net/CaRMetal/forums/viewtopic.php?t=359>

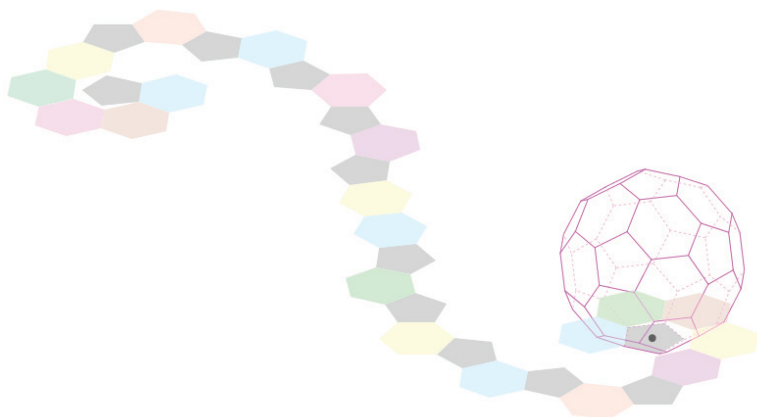
dans laquelle l’auteur donne de nombreuses explications et illustrations autres que ses commentaires détaillés déjà présents dans le script.

⁶¹ <http://db-maths.nuxit.net/CaRMetal/forums/viewtopic.php?t=360>



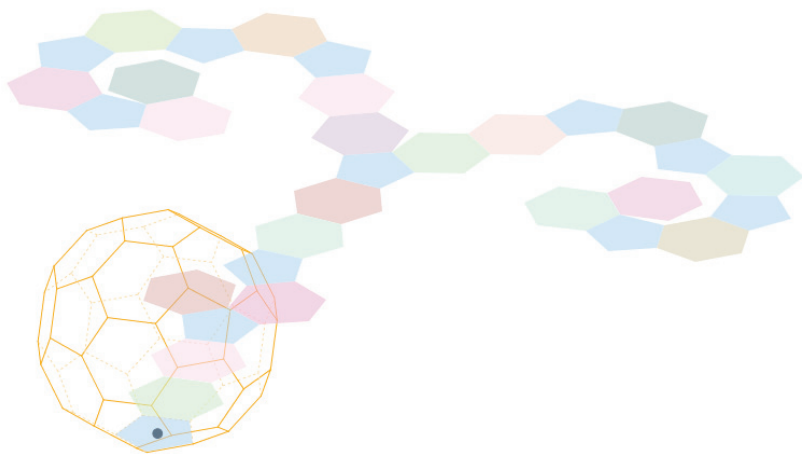
Les 43380 patrons du dodécaèdre au bout des clics de l'utilisateur

Encore une fois, Pierre-Marc Mazat a proposé une variante de ces scripts, en particulier en proposant, sur une figure plus complexe⁶², la possibilité de cliquer sur les arêtes, non plus du polyèdre de départ, mais directement sur les faces du patron. Cela permet de chercher des patrons spécifiques :

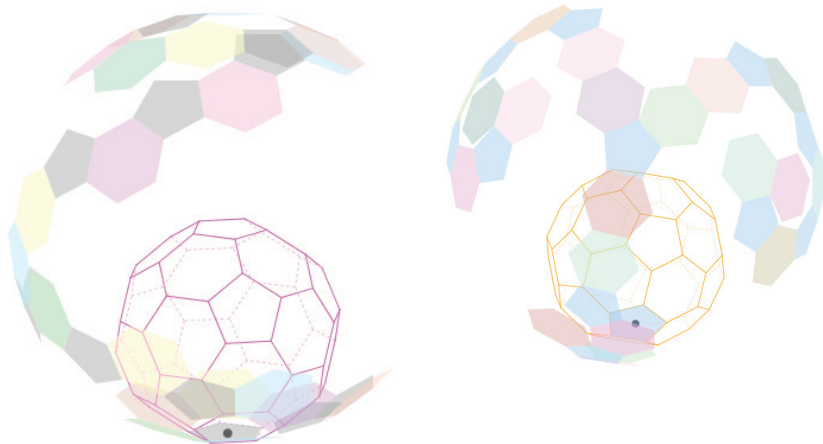


⁶² L'icosaèdre tronqué. Figures et vidéos explicatives disponibles à cette adresse : <http://db-maths.nuxit.net/CaRMetal/forums/viewtopic.php?t=363>

On utilisera l'archive du 20 juillet, plus complète que la première. Elle permet d'enregistrer des patrons déjà réalisés et d'en produire aléatoirement. Pour voir rapidement le fonctionnement du script principal, on peut consulter la première des six vidéos proposées. Elle reprend les différentes possibilités pour construire ces patrons.



Combien de patrons de l'icosaèdre tronqué ?



On comprendra que nous ne détaillons pas plus avant les scripts de ces constructions. La seconde version est bien plus lisible, même si elle est plus compacte ; plusieurs paramètres placés dans les tableaux de points sont explicitement dans la structure objet :

```

1 //Objets
2 Hexagone = function(art0, art1, art2, art3, art4, art5){
3   this.etat = 0;
4   this.arete = new Array(art0, art1, art2, art3, art4, art5);
5   this.segment = new Array(6);
6   this.point = new Array(6);
7   this.o;
8   this.z;
9 }
10 Pentagone = function(art0, art1, art2, art3, art4){
11   this.etat = 0;
12   this.arete = new Array(art0, art1, art2, art3, art4);
13   this.segment = new Array(5);
14   this.point = new Array(5);
15   this.o;
16   this.z;
17 }

```

Les objets de base utilisés dans ce script (largement commenté par l'auteur)

Comme souvent, ces premiers scripts objets⁶³ ont été mis en œuvre pour réaliser des figures complexes qu'on ne réaliserait pas sans eux. Mais des utilisations plus élémentaires sont tout aussi possibles et didactiquement pertinentes (à la fois pour une utilisation en classe et en formation des enseignants). C'est par exemple le cas d'un premier travail mené sur les nombres complexes⁶⁴. Après la vibration du repère 3D pour récupérer les coordonnées des points, avec les scripts objets et la puissance de cette programmation, le travail conjoint de ces deux collègues semble ouvrir un nouveau chantier pour la géométrie dynamique, avec ses premières illustrations dont on ne se lasse pas...

⁶³ On trouvera un script pour les patrons de l'icosaèdre à cette adresse :

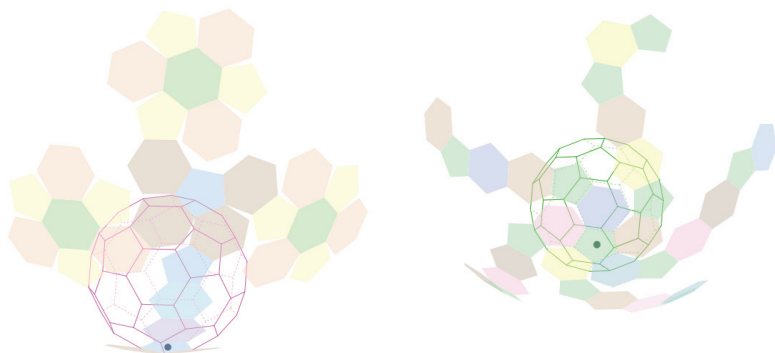
<http://db-maths.nuxit.net/CaRMetal/forums/viewtopic.php?t=364>

⁶⁴ Disponible dans ce wikibook :

http://fr.wikibooks.org/wiki/Programmation_objet_et_g%C3%A9om%C3%A9trie/CaRScripts_et_nombres_complexes

Voir aussi cette page, initiée par J. Caré, qui explore autrement les possibilités 3D :

http://fr.wikibooks.org/wiki/Programmation_objet_et_g%C3%A9om%C3%A9trie/Poins_3D_avec_CaRScript



X. Conclusion et perspectives

La naissance de l'équipe EREDIM au sein du LIM de l'université de la Réunion a été l'occasion de proposer un tour d'horizon des potentialités didactiques du logiciel CaRMetal pour un usage de la géométrie dynamique et/ou de la programmation. Ce tour d'horizon a été composé à partir de réflexions issues d'une pratique personnelle de la GD depuis 1989 et de nombreuses formations initiales et continues animées depuis plusieurs années avec ce logiciel, dans le premier degré et dans le second degré.

Dans cette partie nous ciblons quelques-unes des possibilités de recherches en didactique que l'on peut dégager de ces premières réflexions. Certaines seront menées au sein de l'EREDIM dans le cadre de masters « enseignement » ou de masters « recherche ». Nous invitons les collègues didacticiens à s'emparer eux aussi de ces thèmes pour les faire vivre à leur façon, dans leurs structures, au sein de leurs parcours didactiques, et bien entendu à en dégager d'autres sur cette géométrie dynamique, effectivement enrichie.

Rappelons les quelques points qui émergent de notre texte. Tout d'abord, nous avons vu que l'anticipation systématique de tous les outils provoque de fait une nouvelle instrumentation de la géométrie dynamique. Celle-ci peut être expérimentée et explorée dans des ingénieries précises qui vont instrumentaliser quelques particularités. On pense en premier lieu à l'étude des constructions des représentations chez les jeunes élèves (cycle 3 de l'école primaire, cycle d'adaptation du collège) par appropriation des schèmes

d'usage⁶⁵ des outils. Pour ces recherches, le tout nouvel outil « Monkey », outre qu'il favorise lui aussi l'instrumentation de la GD en tant que telle – et donc le concept de figure géométrique – devrait pouvoir être instrumentalisé dans des ingénieries adaptées, plus particulièrement pour étudier l'investigation sur « la géométrie des propriétés des objets » de l'équipe ERMEL.

Une autre piste à ce niveau d'expérimentation pourrait être un travail de réflexion sur les problématiques et les hypothèses de recherches, la mise en œuvre d'ingénieries sur des configurations finies, avec la manipulation directe discrète que nous avons dégagée comme première application didactique de l'aimantation. Les problématiques peuvent être variées, sur l'apprentissage des configurations ou de leurs propriétés – par la perception spécifique du discret comme on l'a vu avec le quadrilatère orthocentrique, mais aussi l'appropriation des schèmes d'action dans un contexte discret pour mieux s'approprier le contexte naturel, etc.

Dans tous les cas une variable didactique importante est l'environnement restreint utilisé. Une réflexion doit être engagée pour que cet environnement ne transforme pas la richesse et la richesse d'investigation d'un micromonde en simple exerciciel à environnement dynamique.

Dans le cycle d'orientation du collègue, sur la base d'hypothèses précises, des ingénieries peuvent être développées pour tester l'introduction des fonctions dans le passage du registre discret (les entiers les demi-entiers) au registre continu. Cela n'a pas été présenté, mais il n'est pas nécessaire d'utiliser l'aimantation pour cela, on peut utiliser la grille du logiciel qui peut être aimantée sur l'unité et la demi-unité.

Au lycée, l'a-modalité de tous les outils, mais plus particulièrement de l'inspecteur d'objet, a été soulignée pour entreprendre, avec l'onglet Numérique de l'inspecteur d'objet, d'étudier ce que nous avons appelé la géométrie repérée dynamique (GRD) avec son engagement du côté de l'algèbre tout en permettant un travail, une manipulation – et une expérimentation par les élèves – sur les objets (la GRD comme objet d'étude). On a vu que le même travail, mais avec des objectifs de recherche largement différents, peut-être envisagé avec un emploi spécifique des scripts pour utiliser la géométrie

⁶⁵ La même chose serait passionnante à travailler sur des géométries non standard comme le plan de Moulton, pour explorer comment des étudiants en mathématiques déjà chevronnés (M1 par exemple) utilisent l'anticipation des outils pour s'approprier plus rapidement le comportement des objets de cette géométrie dans ce modèle et en tirent des conclusions sur les possibilités de configuration. Il n'est pas certain que cela puisse être mis en place, mais c'est un projet de recherche personnelle potentielle.

repérée dynamique à des fins de validation dans un travail de programmation (la GRD comme outil pour les élèves).

Dans le cadre de l'utilisation des scripts, nous avons vu comment ce contexte, à travers des ingénieries adaptées, peut précipiter la réflexion sur le statut de variables (y compris chez les enseignants) par la confrontation de deux types de variables, celles du logiciel, naturellement dynamiques et celle du JavaScript qui peuvent être passées algébriquement à la figure ou par valeur numérique, et donc avec un statut de nombre. Nous avons vu – sur une pratique de formation continue, qu'il n'y avait aucune ambiguïté sur le partage d'un segment, et comment l'ambiguïté naissait sur le partage du cercle. Il y a là tout un champ de recherches didactiques possibles.

Toujours au lycée, des recherches peuvent être entreprises sur l'usage de la logique dans des figures, notamment dans un cadre de « logique-outil » – d'abord avec l'onglet Conditionnel de l'inspecteur d'objet, sur des ingénieries accessibles à de nombreuses classes, puis ensuite avec les booléens pour un usage réservé aux classes scientifiques. Nous avons en particulier vu comment l'usage de la logique réifiait l'unicité d'une solution quand elle se décline en plusieurs cas à traiter dans le champ des configurations. Des recherches sur la logique-outil et éventuellement sur cette réification peuvent être entreprises aussi dans les deux premières années de l'université.

Sans que nous ayons véritablement de pistes bien précises pour une recherche didactique en classe, nous envisageons à l'EREDIM de poursuivre la réflexion sur le *déterminisme enrichi*, plus particulièrement sur l'engagement direct augmenté, quand celui-ci propose à l'utilisateur un environnement global riche, recherché au cours d'une phase d'investigation, dans lequel l'utilisateur peut entrer et en sortir.

L'interaction entre le langage de script et cette implémentation de la géométrie dynamique avec toute la richesse que l'on a vue dans la dernière partie invite à entreprendre des développements propres, pour l'étude de cette intégration elle-même. Quelles applications imaginer pour la récursivité croisée au sein des scripts ? Comment intégrer la temporalité dans des scripts ? Quelles orientations didactiques aborder avec la vibration du trièdre et cette nouvelle *injection dynamique* de $\mathbf{R}^3$ sur $\mathbf{R}^2$? Quelles ingénieries spécifiques va-t-on développer, mais aussi quelles nouvelles problématiques va-t-on rencontrer ? Enfin, avec les scripts objets, que les objets soient d'essence scolaire ou non, de nombreux champs d'investigation sont envisageables même s'ils peuvent être, dans un premier temps, éloignés de la didactique.

Bibliographie

Textes historiques

- BELTRAMI Eugène (1868), « Essai d'interprétation de la géométrie non euclidienne », trad. par Jules Hoüel, *Annales Scientifiques de l'École Normale Supérieure*, 6, p. 251-288.
- BOLYAI Janos (1867), « La science Absolue de l'Espace indépendante de la vérité ou de la fausseté de l'axiome XI d'Euclide (que l'on ne pourrait jamais établir à priori) ; suivi de la quadrature géométrique du cercle dans le cas de la fausseté de l'Axiome XI », trad. par Jules Hoüel, *Mémoires de la Société des sciences physiques et naturelles de Bordeaux*, 5, p. 189-246.
- COXETER Harold Scott MacDonald (1989), *Introduction to Geometry*, 2nd ed., Wiley.
- HILBERT David (1971), *Les Fondements de la géométrie*, éd. critique et trad. par Paul Rossier, Dunod (1^{re} éd., *Grundlagen der Geometrie*, Teubner, 1899).
- MOULTON Forest Ray (1902), « A simple non-Desarguesian geometry », *Transactions of the American Mathematical Society*, 3, p. 192-195.

Géométrie dynamique

- ACOSTA Martin (2008), *Démarche expérimentale, validation, et ostensifs informatisés. Implications dans la formation d'enseignants à l'utilisation de Cabri en classe de géométrie*, Thèse de doctorat de l'université de Genève.
- BELLEMAIN Franck (1992), *Conception, réalisation et expérimentation d'un logiciel d'aide à l'enseignement de la géométrie : Cabri-géomètre*, Thèse de doctorat de l'IMAG, Grenoble.
- GALWICK Thomas (2002), « Dynamic Notions for Dynamic Geometry », *5th International Congress on Teaching Mathematics with Technology (ICTMT 5)*.
- GENEVÈS Bernard (2004), *Vers des spécifications formelles : Fondements mathématiques et informatique pour la géométrie dynamique*, Thèse de doctorat de l'université Joseph Fourier Grenoble 1.
- KORTENKAMP Ulli (1999), *Foundations of Dynamic Geometry*, Thèse d'habilitation de l'ETH Zürich.

- KORTENKAMP Ulli & RICHTER-GEBERT Jürgen (2001), « Grundlagen dynamischer Geometrie », in *Mathematische und didaktische Aspekte Dynamischer Geometrie Software*, Hildesheim & Berlin : Franzbecker.
- LABORDE Jean Marie (1985), « Spécification informelle pour un Cahier de Brouillon Informatique pour la géométrie », in *Environnements Interactifs d'Apprentissage avec Ordinateurs*, Eyrolles, Paris.
- LABORDE Jean Marie (1996), « Explorations en géométries non euclidiennes », in *Actes de l'Université d'été « Cabri-géomètre, de l'ordinateur à la calculatrice, De nouveaux outils pour l'enseignement de la géométrie »*, Grenoble
- MARTIN Yves (2003), *Conception et mise en œuvre de micromondes de géométries non-euclidienne dans le cadre de la géométrie dynamique, illustrées avec Cabri-géomètre. Expérimentation en formation des maîtres*, Thèse de l'université Joseph Fourier-Grenoble 1.
- PAPERT Seymour (1980), *Jaillissement de l'esprit. Ordinateurs et apprentissage*, Paris, Flammarion.

EIAH

- BALACHEFF Nicolas (2002), « Contribution à la réflexion sur la recherche sur les Environnements Informatiques pour l'Apprentissage Humain », in *Les technologies en éducation : Perspectives de recherche et questions vives*, Paris, INRP, MSH & IUFM de Basse Normandie, p. 193-201.
- BEAUDOUIN LAFON Michel (1997), *Interactions instrumentales : de la manipulation directe à la réalité augmentée*, Orsay, LRI-CNRS.
- SHNEIDERMAN Ben (1983), « Direct manipulation : a step beyond programming languages », *IEEE Computer*, 16 (8), p. 57-69.
- NANARD Jocelyne (1990), *La manipulation directe en Interface Homme Machine*, Thèse d'État de l'U.S.T., Montpellier.
- TCHOUNIKINE Pierre (2002), « Pour une ingénierie des EIAH », *Revue I3 information-interaction-intelligence* (www.revue-i3.org).

Mathématique et Didactique

- BRISSIAUD Rémi (2003), *Comment les enfants apprennent à calculer*, Paris, Retz.
- BROUSSEAU Guy (1986), *La théorisation des phénomènes d'enseignement des mathématiques*, Thèse d'État de l'université de Bordeaux 1.

CUPPENS Roger (1996-1999), *Faire de la géométrie... avec Cabri*, 4 vol., Paris, APMEP.

POLSTER Burkard (1998), *A Geometrical Picture Book*, New York, Springer

TOURNÈS Dominique (2004), *Autour d'un mémoire de Vincenzo Riccati : histoire de la construction tractionnelle des équations différentielles*, Thèse d'habilitation de l'université Pierre et Marie Curie-Paris 6.

Webographie

Cet article a son complément en ligne, avec toutes les figures citées – et bien d'autres – manipulables en ligne et téléchargeables, ainsi que tous les scripts :

Pour télécharger les classeurs de géométrie :

<http://db-maths.nuxit.net/IREMrun/art372/a372inter.html>

Pour télécharger les classeurs de scripts :

<http://db-maths.nuxit.net/IREMrun/art375/a375galerie.html>

Le site de CaRMetal pour utiliser les classeurs directement

<http://db-maths.nuxit.net/CaRMetal/>

(téléchargement, tutoriaux vidéo, ressources, forums)

Sur le site de l'IREM de la Réunion :

Présentation de la version 3.5 (avril 2010)

<http://www.reunion.iufm.fr/recherche/irem/spip.php?article347>

Le groupe de travail Algorithmique et CarScripts :

<http://www.reunion.iufm.fr/recherche/irem/spip.php?rubrique58>

avec un manuel de référence de 67 pages (Alain Busser), des articles de prise en main pour l'enseignant, et de propositions d'utilisations dans lesquelles les exemples vus ici sont repris et largement détaillés, avec plusieurs dizaines de scripts. On trouvera aussi deux rubriques très fournies d'Alain Busser :

- Narration de recherche sur les [TP d'algorithmique en classe](#) de seconde.
- Sujets de l'épreuve pratique TS 2009 [avec de nombreux scripts](#).

Le groupe de travail Abaques et nomogrammes

<http://www.reunion.iufm.fr/recherche/irem/spip.php?rubrique49>

utilise les CarScripts. Les scripts (Alain Busser) sont essentiellement dans les quatre sous-rubriques.