

APLUSIX, un EIAH de l'algèbre

Denis Bouhineau, Jean-François Nicaud

► **To cite this version:**

Denis Bouhineau, Jean-François Nicaud. APLUSIX, un EIAH de l'algèbre. Grandbastien, Monique and Labat, Jean-Marc. Environnements Informatiques et Apprentissage Humain, Hermès-Lavoisier, pp.333-350, 2006, Collection IC2. <hal-00962040>

HAL Id: hal-00962040

<https://hal.archives-ouvertes.fr/hal-00962040>

Submitted on 28 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Chapitre 15

APLUSIX, un EIAH de l'algèbre

15.1. Introduction

Ce chapitre décrit la (re-)conception de l'EIAH APLUSIX, micromonde d'apprentissage de l'algèbre pour les collèges et les lycées, ainsi que les premières expérimentations effectuées et les premiers usages observés en classe avec le produit obtenu. En particulier, ce chapitre témoigne de la tension qui existe au moment de la conception d'un EIAH entre deux problématiques, la première est liée aux objectifs d'apprentissage, la seconde concerne le développement d'une interface homme-machine (IHM). À l'intersection de ces deux domaines de recherche se trouvent les notions de micromonde et d'ergonomie notions centrales de ce chapitre.

Avant 2000, le projet APLUSIX a produit des tuteurs pour les factorisations d'expressions polynomiales, cf. [NIC 90]. Ces tuteurs étaient basés sur un petit nombre de commandes (regroupant chacune plusieurs opérateurs, comme la commande réduire) présentées dans des menus. Des expérimentations ont été conduites avec ces tuteurs. Le lecteur en trouvera la synthèse dans [NGU 02]. En 2000, s'est opéré un changement radical par adoption d'une approche 'micromonde' décrite dans ce chapitre et qui a donné lieu à la conception d'un nouvel APLUSIX. Ce changement a été motivé par la faible utilisation des systèmes basés sur des commandes : bien que ces systèmes aient prouvé qu'ils peuvent avoir un apport pour l'apprentissage, ni les professeurs, ni les élèves ne se sont montrés vraiment attirés par eux. C'est donc avec cet objectif « être réellement utilisé en classe » que la conception du nouvel APLUSIX s'est déroulée.

Version auteur (prépublication) d'un chapitre rédigé en 2006 par D. Bouhineau et J.-F. Nicaud, enseignants-chercheurs au Laboratoire d'Informatique de Grenoble.

2 Environnements Informatiques pour l'Apprentissage Humain

La section 15.2 opérera un rapide tour d'horizon des utilisations en classe de l'ordinateur pour l'enseignement de l'algèbre. En 15.3 les objectifs et les principes de la conception d'APLUSIX seront exposés. En particulier, les aspects liés à la notion de micromonde (15.4), à l'ergonomie (15.5) et aux rétroactions (15.6) seront approfondis, en prenant en compte différents aspects liés à l'apprentissage, aux cursus, et à l'épistémologie du domaine de l'algèbre qui nous intéresse. Enfin, la section 15.7 comportera un aperçu des expérimentations et retours d'usages.

15.2. Rapide tour d'horizon sur les utilisations en classe de l'ordinateur pour l'enseignement de l'algèbre

Afin de situer notre travail parmi les EIAH des mathématiques, commençons par un rapide tour d'horizon du domaine. Intéressons-nous plus particulièrement à l'utilisation d'environnements informatiques de tout types lors des premiers enseignements de l'algèbre dans l'enseignement secondaire (lycée et collège). Nous laisserons donc de côté les travaux en analyse et en géométrie (lire le chapitre 16 pour la géométrie), en notant cependant que les environnements produits dans ces domaines peuvent servir pour confronter l'algèbre avec d'autres cadres de référence des mathématiques.

Côté EIAH, complémentaire des enseignements techniques de l'algèbre, le problème de la modélisation en formules algébriques des énoncés donnés en langue naturelle a été abordé par l'équipe constituée autour de Ken Koedinger et John Anderson de la Carnegie Mellon University. Les résultats obtenus concernent le sous domaine des problèmes donnant lieu à des équations linéaires. Un EIAH de type tuteur a été produit pour la résolution de ce type d'équations. Il a été largement utilisé en classe [KOE 97]. D'autres sous domaines de l'algèbre ont été abordés par des EIAH. Pour exemple, nous citerons nos travaux précédents sur les factorisations [NIC 90]. D'autres exemples peuvent être donnés pour l'arithmétique ou la résolution des équations du second degré par la méthode du discriminant ou la résolution de systèmes linéaires par une élimination à la Gauss. La réduction du domaine a souvent pour avantage l'obtention d'un EIAH ayant une connaissance fine d'un domaine particulier. Elle comporte des inconvénients, principalement, la réduction du domaine d'utilisabilité. Quelques EIAH ont essayé de couvrir un domaine large. Par exemple, intermédiaire entre le tuteur, l'environnement de calcul formel et l'exerciceur, se trouve le travail de Michael Beeson. L'EIAH produit permet la résolution de problèmes par application de règles de calculs algébriques disponibles dans l'environnement [BEE 96]. À la frontière des EIAH, se trouvent des exercices des mathématiques plus ou moins perfectionnés, allant du simple QCM aux environnements effectuant une véritable analyse mathématique des réponses. Citons en vrac, Lilimath, Wims, TreeFrog que le lecteur retrouvera aisément sur le web. Enfin, notons également l'atelier « Pépite » [JEA 97] appartenant au domaine des EIAH de diagnostic des compétences en algèbre.

Du côté des usages, nous remarquons la présence de logiciels non spécifiquement prévus ni pour une utilisation en classe, ni pour les premiers apprentissages en algèbre. Nous notons tout d'abord les logiciels de calcul formel. Signalons que leur utilisation est très controversée [TRO 02]. Parmi eux, mentionnons l'exemple de Mathematica, dont les capacités de traitement de texte et de réalisation d'interface a permis le développement d'EIAH. Ainsi, des travaux sur l'enseignement de la preuve et la rédaction de la preuve en mathématiques ont été produits dans le projet Theorema de Bruno Buchberger [BUC 02]. Notons également, mu-Pad qui sert comme moteur de calcul formel dans un certain nombre d'EIAH ; pour exemple [LAG 02, ACT 02]. À côté des environnements de calculs formels, remarquons aussi les logiciels de bureautique (tableurs surtout, mais aussi traitements de texte) et certaines calculatrices scientifiques dont la puissante calculatrice graphique disponible dans la gamme Macintosh de Ron Avitzur [ERI 02].

Première remarque concernant la plupart de ces environnements : en général l'ordinateur apporte la rigueur du calcul numérique et la rigueur dans l'application de règles de transformations algébriques. En outre, ce sont souvent des systèmes basés sur un système de commandes algébriques (proche des logiciels de calcul formel), c'est-à-dire des systèmes 'orientés commandes'. Leur utilisation suppose que l'arithmétique, les notions d'appariement et d'utilisation des règles de transformations algébriques soient acquises par l'apprenant et que l'apprentissage de l'algèbre se trouve au delà, dans la reconnaissance des identités remarquables et des algorithmes à employer. Ces *a priori* ne sont pas légitimes pour les premiers enseignements en algèbre.

Seconde remarque : la plupart du temps pour les EIAH, et radicalement pour les environnements à QCM, l'apprenant est très limité dans ses productions ; il possède rarement la liberté de développer ses propres calculs comme il l'aurait fait dans un environnement papier-crayon. Il n'explore plus le même domaine d'expressions et n'envisage pas les mêmes raisonnements. En particulier, il n'est plus confronté à la plupart des erreurs qu'il pourrait faire, l'ordinateur ne lui autorisant que des calculs justes.

Dernière remarque : il existe une différence importante entre les EIAH et les autres logiciels, elle se trouve au niveau de l'ergonomie. Chaque type de logiciel a une ergonomie propre, celle des logiciels professionnels n'est pas celle des logiciels pour l'enseignement. Cela justifie un besoin d'EIAH et montre bien l'importance de l'ergonomie dans le champ des EIAH.

15.3 Principaux objectifs et principes de conception d'APLUSIX

Depuis le renouveau du projet APLUSIX en septembre 2000, APLUSIX est conçu par deux enseignants-chercheurs en informatique de l'enseignement supérieur, les auteurs de ce chapitre. L'objectif principal est de produire un environnement

4 Environnements Informatiques pour l'Apprentissage Humain

d'apprentissage de l'algèbre qui puisse réellement être utilisé en classe, en milieu écologique. Cela signifie, d'une part, que l'utilisation de ce logiciel doit se faire effectivement hors du laboratoire de conception, sous la conduite d'enseignants de mathématiques indépendants du projet, sans formation spécifique sur le logiciel, avec leurs élèves habituels. Cela signifie, d'autre part, que la conception doit permettre que le produit réalisé soit largement diffusé nationalement et internationalement.

Cet objectif premier s'accompagne d'autres objectifs importants, plus classiques dans le monde universitaire. Tout d'abord, bien entendu, il s'agit de fournir un environnement qui favorise les apprentissages en algèbre. Mais, il faut aussi qu'il permette d'étudier ces apprentissages d'un point de vue épistémologique, didactique, psychologique pour que se poursuivent les études universitaires auxquelles nous participons. Également, à travers le développement d'APLUSIX, nous devons pouvoir poursuivre un travail de recherche en informatique sur la conception des EIAH. Enfin, nous avons aussi comme objectif le réinvestissement de l'expertise dans le domaine de l'apprentissage de l'algèbre acquise par le long travail en laboratoire du précédent projet APLUSIX.

De manière plus pragmatique, très concrètement, l'objet du projet est la conception d'un environnement de type 'micromonde' où l'apprenant est libre de construire des expressions algébriques et de mener ses propres raisonnements, de pouvoir les modifier librement, en ayant à sa disposition des rétroactions épistémiques. Le domaine visé est constitué des premiers enseignements en algèbre au collège et au lycée : manipulations algébriques sur des formules polynomiales (factorisation, développement, réduction), résolution d'équations, d'inéquations et de systèmes d'équations. Dans le contexte de l'enseignement de l'algèbre avec un ordinateur décrit à la section précédente, l'environnement visé est original, il s'apparente à un cahier de brouillon informatique¹ pour les raisonnements par équivalence en algèbre où l'apprenant a la même liberté qu'avec l'environnement habituel papier-crayon. Le choix d'une approche 'micromonde' nous semble le plus judicieux pour favoriser des usages étendus. Ce choix est compatible avec une théorie de l'apprentissage constructiviste. La présence de rétroactions peut faire penser à l'emploi d'une théorie SR de l'apprentissage. En fait, la conception de l'EIAH décrite ici ne repose pas sur une théorie de l'apprentissage en particulier. Globalement, le noyau de l'environnement envisagé est constitué de deux micromondes, l'un pour les raisonnements par équivalence et l'autre pour les expressions algébriques. En tant que micromondes chacun de ces deux objets définissent un univers mathématique particulier. Ensemble ils constituent un monde algébrique complet possédant une connaissance fine de la syntaxe et de la sémantique des expressions permettant des rétroactions épistémiques. L'idée de départ du projet est de fournir un environnement où à chaque instant, l'apprenant

¹ Avant goût du chapitre 16, **Cabri**, le logiciel qui a inventé la géométrie dynamique et qui est leader dans ce domaine signifie **C**ahier de **B**rouillon **I**nformatique.

doit avoir à sa disposition des rétroactions qui lui indiquent si son expression algébrique est syntaxiquement correcte et bien définie, si le raisonnement qu'il poursuit est sémantiquement correct, c'est-à-dire, s'il est juste, ainsi que des indicateurs stratégiques relatifs à l'avancement de la résolution. Avec ces trois guidages, syntaxique, sémantique et stratégique, et la liberté associée au concept de micromonde l'environnement visé combine ce que l'on a habituellement en papier-crayon et ce que l'on peut faire avec un précepteur. Ces trois guidages, associés à trois groupes de rétroactions seront décrites dans la section 15.6 qui leur est consacrée.

Revenons sur l'objectif principal « être largement utilisé en milieu écologique ». Pour chaque type d'utilisateurs (apprenants et enseignants) cet objectif doit être considéré avec attention au cours de la conception. Du côté de l'utilisateur apprenant, l'EIAH doit être construit autour d'un noyau avec une ergonomie particulièrement travaillée et capable de prendre en compte un domaine aussi large que possible. L'interface doit comporter en particulier des éditeurs évolués et performants qui concrétisent un monde algébrique dont la nature est claire et proche de celui étudié en classe. Le choix d'une approche de type 'micromonde' doit donc s'enrichir d'un travail conséquent sur l'ergonomie. Selon nous, ces deux points sont les éléments principaux susceptibles de favoriser l'appropriation de l'outil par le plus grand nombre. Du côté de l'utilisateur enseignant, il doit y avoir un grand nombre de services associés à l'EIAH, par exemple : pour paramétrer l'EIAH selon l'activité que l'enseignant veut réaliser ; pour organiser le travail en classe (utilisation de listes d'exercices produites par l'enseignant, dont des exercices de modélisation donnés en langue naturelle, écriture par l'enseignant et de scénarios² d'utilisation de l'EIAH) ; pour enregistrer, relire, analyser automatiquement la production des apprenants. En particulier, afin de convenir à un maximum d'enseignants, l'EIAH doit pouvoir être configuré en ce qui concerne les points où il n'y a pas d'accord entre les enseignants (par ex. : faut-il autoriser/obliger la rédaction de commentaires lors de la résolution d'un problème) ou vis-à-vis des points innovants ou apportant des fonctionnalités que tel enseignant pourrait ne pas vouloir, de telle sorte qu'au minimum l'EIAH soit proche de la référence : l'environnement papier-crayon.

15.4 Définition des micromondes algébriques

Dans [LAB 89, BAL 94], la notion de micromonde est définie comme une construction informatique réifiant des objets et des relations existant entre ces objets. Un micromonde permet la construction de ces objets, leur modification,

² Il s'agit de scénarios informatiques. Les scénarios envisagés sont des listes de phases organisées chronologiquement. Une phase est l'association, entre autre, d'une liste d'exercices, d'une liste de paramètres pour l'EIAH, d'un temps d'exécution, et de différents messages.

6 Environnements Informatiques pour l'Apprentissage Humain

l'exploration des propriétés de ces objets, comme l'explicitera plus longuement le chapitre 16. C'est un éditeur et un visualisateur évolué d'objets complexes. Appliquée à APLUSIX, cette définition délimite deux entités constituant APLUSIX : l'éditeur ergonomique d'expressions (décrit section 15.4.1) et l'éditeur simple de raisonnements algébriques (décrit section 15.4.2). Chacun de ces deux éditeurs peut être vu comme un micromonde, chacun définissant un univers mathématique propre et comportant ses rétroactions.

L'univers des expressions et celui des raisonnements partagent comme objets de base les expressions algébriques. Deux types de relations existent pour ces objets : une relation structurelle dans l'univers des expressions, et une relation d'équivalence entre expressions dans l'univers des raisonnements. Tout d'abord, la relation structurelle : ce sont les opérateurs (entiers, variables, opérateurs arithmétiques + - * /) qui permettent de construire des expressions algébriques. Ces expressions algébriques peuvent ensuite être associées avec des opérateurs relationnels (=, ≠, >, ≥, ≤, <) pour produire des expressions booléennes (qui elles-mêmes peuvent être composées à l'aide des opérateurs booléens (et, ou) pour fournir des expressions booléennes complexes, par abus de langage nous appellerons encore algébrique ces expressions). Ensuite, ces expressions algébriques sont reliées par une relation d'équivalence. Par exemple, $(x+2)^2 = 0 \Leftrightarrow x+2 = 0$. Les objets ainsi construits sont appelés des raisonnements algébriques.

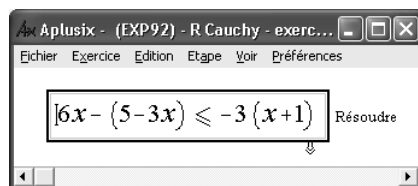


Figure 1 : Début de session avec APLUSIX : dans une fenêtre presque vide (réduite ici pour gagner de la place), l'utilisateur trouve une expression algébrique et une consigne.

Pour une vue d'ensemble de l'agencement entre ces éléments : la construction de nouveaux objets se fait au niveau des expressions via l'éditeur d'expressions algébriques, simplement en frappant un opérateur (ou en effaçant une partie de l'expression préexistante) après avoir positionné un curseur d'insertion dans une étape existante. La construction de raisonnements algébriques se fait en demandant une nouvelle étape de raisonnement à partir d'une étape existante (par menu, par un clic sur la petite flèche à la base de l'étape courante, ou en frappant la touche entrée). Les modifications apportées aux expressions et aux raisonnements sont visualisées immédiatement. À l'état initial, le micromonde se présente habituellement : soit avec une expression algébrique vide et modifiable formant l'étape initial d'un raisonnement à produire, soit avec une expression algébrique initiale non modifiable

et une consigne d'exercice formant la première étape d'une résolution d'exercice, comme figure 1. Après création de plusieurs expressions et étapes, l'utilisateur peut se trouver comme figure 2.

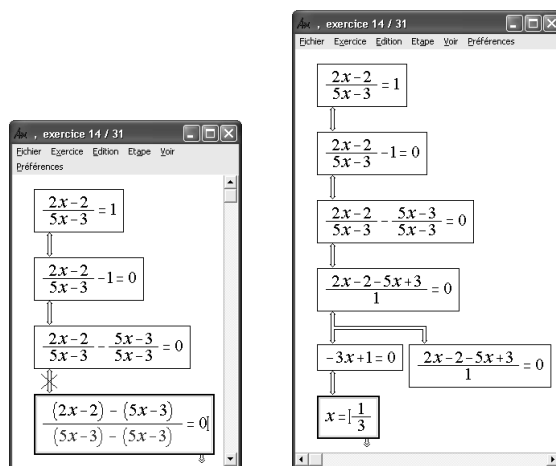


Figure 2 : exemple de raisonnements avec APLUSIX :
à gauche avec une dernière étape incorrecte, à droite avec deux branches parallèles.

Approfondissons maintenant la description de chacun des deux micromondes.

15.4.1 Micromonde des expressions algébriques

L'éditeur d'expressions algébriques est un élément clé d'APLUSIX. Il est le fruit d'un travail important qui, d'un côté a porté sur l'ergonomie et, de l'autre, sur l'algèbre des expressions définies dans le micromonde des expressions algébriques. Pour respecter les aspects micromonde algébrique, les objets manipulés dans l'éditeur sont des objets algébriques ; cela concerne non seulement les expressions apparaissant à l'écran, mais aussi les sélections permises à la souris ou au clavier. En particulier, les sélections sont des expressions algébriques bien formées, mais aussi des sous-expressions de l'expression qui les englobe et lors des insertions de copie d'expression, l'expression obtenue doit contenir parmi ses sous-expressions l'expression copiée. Pour ce qui concerne l'ergonomie, si l'obtention d'une fidélité de la représentation des expressions à l'écran par rapport aux habitudes sur le papier n'entre pas en conflit avec les objectifs de définition d'un micromonde algébrique, le fait de vouloir respecter les habitudes acquises avec les traitements de texte ou les calculatrices et l'intention ne pas trop alourdir l'écriture des expressions par quelques surcoûts dus à une écriture trop algébrique ont introduit des difficultés. Le résultat est la conclusion d'un processus de recherche de consensus entre les deux

8 Environnements Informatiques pour l'Apprentissage Humain

objectifs que sont la définition d'un micromonde algébrique et l'obtention de la meilleure ergonomie. Les conflits entre ces deux objectifs n'ont pas été rares. Au final, le résultat est assez loin des éditeurs scientifiques ou des éditeurs de textes que l'on trouve sur le marché et qui n'ont souvent qu'un objectif d'ergonomie au détriment des sciences.

Prenons un premier exemple pour illustrer notre propos. Lors de l'édition d'une expression, il faut pouvoir concilier la notion d'expression en cours d'édition, incomplète, mais utile pour des raisons ergonomiques évidentes, et le respect des règles des expressions bien formées en algèbre. Par exemple, au cours de l'écriture de l'équation ' $3x+2=5$ ', avant la fin, l'utilisateur arrive à l'expression ' $3x+2=$ '. Celle-ci est incomplète, mal formée, mais nécessaire. Ce genre d'expression apparaît à chaque fois qu'un utilisateur frappe un opérateur, avant de pouvoir frapper l'opérande. Aussi pour compléter des expressions intermédiaires d'une édition des opérandes spéciaux '?' sont ajoutées automatiquement, qui tiennent lieu autant d'opérande que d'emplacement de curseur. Dans l'exemple précédent, l'utilisateur obtient ' $3x+2=?$ ' au lieu de ' $3x+2=$ '. D'un point de vue algébrique, ces '?' sont considérés comme des opérandes et permettent d'avoir une expression bien formée. D'un point de vue ergonomique, ce sont des positions de curseur symbolisées par un '?', alors que les autres positions de curseurs ne sont pas visibles explicitement. Quand le curseur se trouve à cette position, le '?' est représenté légèrement différemment (sur fond sombre). Ainsi, dans la frappe de ' $3x+2=5$ ' après la frappe du '=', l'utilisateur obtient ' $3x+2=?$ ' et le curseur se trouve après le '=' à la position symbolisé par '?'. La frappe du 5 ensuite, fait disparaître le '?', le remplace par un 5 et donne l'expression recherchée. Dans la suite nous appellerons 'expression en cours d'édition' une expression comportant un '?'.

Citons, rapidement, deux autres conflits entre ergonomie et micromonde. L'opérateur parenthèse nécessite la frappe simultanée de deux éléments (la parenthèse ouvrante et la parenthèse fermante). Cela ne peut se faire d'un point de vue pratique. Alors comment pouvoir dissocier ces deux frappes et conserver des expressions toujours bien formées? Dans une expression comme ' $3x^2$ ', il y a plusieurs significations possibles pour la position du curseur entre le x et le 2 : derrière le 'x', devant le '2', derrière le produit ' $3x$ '. Comment distinguer ces différentes positions et permettre un déplacement fluide du curseur au clavier avec les flèches gauche et droite, sans qu'il y ait de *piétinement* ?

Voici quelques exemples de règles de mises en œuvre pour réaliser quelques-unes des actions d'un éditeur algébrique. Une description plus complète se trouve dans [NIC 04].

- Lors d'une frappe sur une sélection :
 - o si la frappe est un entier ou une variable : on efface la sélection,

- si la frappe est un opérateur unaire, on prend la sélection comme opérande,
 - si c'est le signe $-$ on change le signe de la sélection (si la sélection avait déjà un signe, il est supprimé),
 - pour le reste : on ne fait rien.
- Lors de la sélection d'expression, il est possible de faire des sélections complexes : des sélections de sous expressions éloignées (non connexes) pour les opérateurs n-aires associatifs, en revanche il n'y a pas de sélection d'opérateur seul.
 - Lors d'un Glisser-Déposer, sur les opérateurs n-aires associatifs, l'expression obtenue est sémantiquement équivalente à l'expression de départ.



figure 3 : clavier virtuel d'APLUSIX.

Par ailleurs, l'éditeur d'expressions algébriques fournit quelques rétroactions en utilisant des couleurs pour certaines sous expressions d'une expression éditée. Le bleu signifie qu'une sous expression est incomplète ou hors domaine. Le rouge signifie qu'une sous expression est indéfinie (d'un point de vue mathématique) comme $1/0$ ou incorrecte du point de vue des types comme $x+x$ ou 1 . Enfin, entre autre à la demande des enseignants utilisant APLUSIX, un clavier virtuel à été ajouté pour faciliter l'appropriation des apprenants qui ont du mal avec le clavier physique et qui préfèrent la souris. Ce clavier virtuel est montré figure 3

15.4.2. Micromonde des raisonnements algébriques par équivalence

Un raisonnement algébrique est représenté dans APLUSIX par un arbre d'étapes, chaque étape comportant une expression algébrique appartenant au micromonde des expressions algébriques. Deux opérateurs existent et peuvent s'appliquer à chaque étape, l'opérateur de création d'une étape fille et l'opérateur de suppression d'une étape. Lors de la création d'une étape, selon un paramètre de l'application, l'étape fille comporte la même expression algébrique que l'étape mère, on parle alors de duplication, ou l'étape fille comporte l'expression vide ' $?$ '. Entre chaque étape mère et étape fille, se trouve le dessin d'un lien avec deux types d'extrémités possibles :

10 Environnements Informatiques pour l'Apprentissage Humain

avec ou sans flèches. Le type de l'expression de l'étape mère fixe le dessin des extrémités du lien. Pour une expression strictement algébrique (polynôme multivarié) : pas de flèche. Pour une expression booléenne (équations, inéquations, systèmes d'équations) : une flèche est présente à chaque extrémité. Pour chaque genre de lien, trois dessins et couleurs sont possibles. Couleur noir pour deux expressions équivalentes, bleu pour des expressions hors domaine ou en cours d'édition, rouge pour deux expressions qui ne sont pas équivalentes. Le dessin du lien, quant à lui, est composé soit d'un trait simple (et alors il est de couleur noire exclusivement, indépendamment de l'équivalence entre expressions) quand un paramètre supprime les rétroactions sur le calcul d'équivalence. Sans ce paramètre, le dessin est composé d'un double trait, barré d'une croix quand la couleur est bleu (expression en cours d'édition), ou rouge (expressions non équivalentes).

La notion d'équivalence est au centre de ce micromonde, lire [NIC 01] pour une définition plus complète de la notion d'équivalence définie dans APLUSIX. Le lien entre étapes, décrit au paragraphe précédent peut prendre cinq formes différentes selon qu'il s'agit d'équivalence entre équations, ou égalité entre polynômes (avec ou sans flèche), selon que l'équivalence est réalisée ou pas (couleur et croix) et selon la valeur du paramètre qui annule tous ces affichages. Le calcul de l'équivalence entre expressions est un problème indécidable dans le cas général des expressions avec sin, exp, valeur absolue, x, [RIC 68]. Il est réputé très difficile dans le cas limité aux expressions algébriques élémentaires, décidable mais exponentiel [TAR 51]. Il est souvent mal géré par les logiciels de calcul formel. L'approche adoptée dans le projet APLUSIX a donc été pragmatique, en limitant le degré des équations à 4, en limitant les systèmes à des équations linéaires, en définissant aussi la notion d'expressions 'hors domaine' pour ce qui nous échappe. L'équivalence proposée est alors bien définie. En effet, pour le domaine défini, la résolution des équations, inéquations et systèmes d'équations est quasi immédiate et la comparaison des ensembles de solutions trouvés permet de conclure.

15.5 Ergonomie et paramétrisation : éléments de l'interface

L'ergonomie d'APLUSIX a été pensée pour qu'un minimum d'apprentissage du logiciel soit nécessaire à l'utilisateur apprenant pour pouvoir l'utiliser. La liberté d'utilisation et la proximité avec l'environnement papier-crayon ont été recherchées.

Une session d'APLUSIX commence avec une fenêtre essentiellement blanche comportant seulement un énoncé composé d'une expression algébrique et d'une consigne (cf. figure 1), comme lors d'un usage en classe sans ordinateur, dans l'environnement papier-crayon. L'élève peut alors produire ses propres pas de calculs, faire plusieurs essais, les gommer, les prolonger, développer son propre arbre de raisonnement, (cf. figure 2)

Il y a bien quelques entorses à la métaphore de l'environnement papier-crayon, l'écriture ne se fait pas avec un crayon (même électronique) mais avec le clavier. L'organisation des pas de calculs dans la fenêtre n'est pas libre comme sur une feuille de papier, mais prédéfinie par l'ordinateur. Cependant, la liberté du papier-crayon est respectée. Le cadre dans lequel se développent les calculs n'est pas contraignant. Ce que l'élève pouvait écrire sur le papier, il peut l'écrire dans la fenêtre.

La recherche d'ergonomie s'exprime aussi dans toutes les actions utilisateur (frappe de touche, sélection, copier-coller) en relation avec le micromonde défini. Ainsi, la syntaxe des expressions reconnues par APLUSIX est une syntaxe 'naturelle' (signe 'multiplié' optionnel, barre de fraction horizontale, vraie racine carrée, ces constructions spatiales définissant implicitement des sous-expressions sans avoir à ajouter de parenthèses supplémentaires etc.), la saisie et la représentation des expressions est spatiale et non linéaire, elle est en WYSIWYG (What You See Is What You Get). Sa gestion n'est pas immédiate, elle repose sur une structure graphique en lien avec la structure algébrique sous-jacente.

Pour améliorer l'ergonomie, adapter le logiciel aux utilisations des enseignants, des paramètres ont été ajoutés. Ces paramètres sont déjà apparus à plusieurs reprises dans les sections précédentes. C'est un point délicat du développement. Les raisons de la présence de ces paramètres sont claires : il s'agit de fournir aux enseignants les moyens de régler les niveaux de guidage et les aides fournies par le logiciel, en fonction des activités qu'ils comptent réaliser ou du niveau de leurs élèves.

Un exemple est donné par les commandes de calcul algébrique disponibles. Elles sont en nombre réduit : calculer (selon les paramètres, cette commande effectue un calcul sur les entiers, ou un calcul sur les entiers, les fractions et les racines carrées), développer-réduire, factoriser, résoudre (selon les paramètres, cette commande effectue la résolution des équations linéaires, des systèmes d'équations linéaires, ou des équations du second degré). L'implémentation de ces commandes repose sur une base de règles de productions algébriques et des méthodes de calcul exact³ définies en fonction des paramètres évoqués. Ces commandes font également parties des paramètres du logiciel. Ainsi, c'est l'enseignant qui fixe leur présence et leur domaine d'application. Coté ergonomie, une fois fixée, il est à noter que la présence

³ Le projet APLUSIX a choisi de travailler avec un module de calcul formel interne développé pour et par le projet, reposant sur l'arithmétique entière et l'arithmétique des réels en précision finie disponible dans l'environnement de base de programmation. Comme pour tout EIAH scientifique effectuant des calculs, il aurait été possible de s'appuyer sur un logiciel de calcul formel existant. Cette option a été rejetée, d'une part, pour garantir l'indépendance du développement et faciliter la diffusion et, d'autre part, pour des raisons pratiques. L'utilisation des logiciels de calcul formel n'est pas immédiate et demande souvent beaucoup de travail, leur utilisation comme expert du domaine n'est pas évidente, cf. [PRA 01].

12 Environnements Informatiques pour l'Apprentissage Humain

de ces commandes est indépendante du problème ou de l'expression sélectionnée, contrairement à ce qui se passe pour [BEE 96].

L'introduction de paramètres (ou leur retrait) est souvent le résultat d'un travail pluridisciplinaire au long court avec les enseignants et les didacticiens. Il faut aussi considérer deux autres points de vue du côté de la conception informatique des EIAH. Tout d'abord, pour les aspects positifs, l'introduction de nombreux paramètres permet le développement de multiples solutions logicielles répondant à diverses préoccupations et choix possibles laissés à l'utilisateur (enseignant ou apprenant), à des moments où la 'bonne' solution n'est pas connue. Rappelons que le développement d'APLUSIX s'effectue en même temps qu'un travail de recherche en EIAH et que, pour beaucoup de points touchant à l'ergonomie ou à la nature algébrique du micromonde défini, il n'y a pas d'évidences et de solutions toutes prêtes. Les paramètres sont des outils d'une démarche exploratoire de définition des points clés d'APLUSIX. Après utilisation du logiciel avec les différentes valeurs possibles des paramètres et observation des usages, certains paramètres sont supprimés. La suppression de certains paramètres, résulte par exemple, de fonctionnalités finalement peu adaptées aux objectifs visés. D'autres paramètres sont supprimés pour diminuer la complexité du travail de configuration pour l'enseignant.

Il est difficile d'évaluer le surcoût dû à cette forte paramétrisation. Le surcroît de travail de développement n'est pas le seul point. Avec plusieurs milliards de possibilités de configurations différentes, nombre rapidement atteint avec plus de vingt paramètres dans les versions intermédiaires, apparaissent des difficultés de mise au point, de recherche de bug et de correction, des tests, de l'évaluation d'un paramètre (indépendamment des autres ?).

15.6. Rétroactions épistémiques

Parmi les trois guidages, syntaxique, sémantique et stratégique, fournis par APLUSIX, les deux premiers s'accordent assez bien de la notion de micromonde. Ils peuvent s'effectuer sans limiter la liberté de l'utilisateur. Le troisième guidage, le guidage stratégique, est beaucoup moins naturel et beaucoup moins évident à mettre en place dans le cas général des micromondes. Le cas d'APLUSIX est différent, car le logiciel comporte une connaissance du 'type de problèmes' envisagé par l'utilisateur. Ainsi sont définis les quatre types de problèmes suivants : calculer, résoudre une équation, factoriser et développer. A chacun est associé une panoplie d'indicateurs qui permettent d'introduire un guidage stratégique. Indicateur de réduction pour le problème 'Calculer' ; indicateurs de factorisation ou de degré de développement pour les deux problèmes 'Factoriser' et 'Développer' ; indicateur de mise sous forme résolue pour le problème 'Résoudre une équation'. Ces indicateurs sont calculés soit à partir d'algorithmes directs (nombre de parenthèses pour 'Développer'), soit à

partir de bases de règles ('Calculer') et affichés sous forme de jauges en bas de la fenêtre, mis à jour en permanence, instantanément.

Selon la configuration souhaitée par le professeur, soit tous les indicateurs sont présents, soit une partie d'entre eux, soit aucun. Ces indicateurs ont été pensés comme des informations objectives relatives à une expression algébrique donnée, indépendamment du contexte d'enseignement où elles apparaissent. Les paramètres qui déterminent leur présence n'influencent pas leur puissance. Contrairement aux commandes qui sont paramétrées par un domaine d'application, ces indicateurs donnent une information absolue, indépendante des compétences acquises par l'apprenant.

Ces indicateurs servent également pour l'évaluation par le logiciel d'un raisonnement proposé par un apprenant. Un problème est dit bien résolu par le logiciel s'il respecte deux conditions. La première condition est sémantique : il faut que la forme finale du problème soit équivalente au problème initial. La seconde condition est donnée par l'indicateur associé au type de problème du raisonnement (ou un ensemble d'indicateurs associé au problème) : il faut que cet indicateur (ou ces indicateurs) soit satisfait.

Mais revenons aux rétroactions les plus évidentes. La plus évidente des rétroactions fournies par le système est d'offrir à l'écran une représentation visuelle, fidèle, immédiate et continue de l'objet algébrique (expression ou raisonnement) sur lequel l'utilisateur travaille.

Au niveau syntaxique, l'introduction du '?' révèle les expressions incomplètes, qui sont soulignées par la couleur bleue au niveau de l'expression et des liens d'équivalence entre étapes d'un raisonnement. La sélection et l'action de coller une expression révèlent la syntaxe des expressions. La forme du lien entre étapes du raisonnement (flèche ou pas flèche) exprime la nature de l'expression. Enfin, un indicateur au bas de la fenêtre de l'application répète le statut de l'expression courante (bien formée, incomplète, hors domaine, incorrecte [bien que cela soit une notion sémantique])

Pour le niveau sémantique, tout tient dans le dessin d'un lien noir ou rouge, barré d'une croix ou non. Insistons sur le fait que l'affichage de cette rétroaction est immédiat et s'effectue en continu, tout au long de l'édition d'une expression. Ajoutons de plus qu'au cours du développement d'APLUSIX un certain nombre de paramètres ont existé pour faire varier les modalités d'affichage de cette information. Entre l'affichage en continu et l'absence d'affichage, il y a un paramètre pour un affichage 'à la demande'. Pour cet affichage à la demande, la durée de l'affichage et la zone où il s'applique (toute la feuille, ou juste l'expression courante et sa mère) ont été réglées par des paramètres. Mais au final, il ne reste qu'un paramètre ayant trois valeurs possibles : affichage en permanence, à la demande, ou jamais.

Opérons une transition vers la dernière section en analysant quelques usages d'APLUSIX. Avec autant de rétroactions, et en particulier avec la rétroaction

sémantique très forte, est-il possible d'avoir des comportements gagnants de devinette ? La réponse est négative dans la plupart des cas. Au vu des enregistrements d'étudiants effectuant une recherche de solution avec des comportements de devinette, la démarche est inopérante pour la plupart des recherches aléatoires de la solution. En revanche, quand il y a un doute sur un signe, ou sur le sens d'une inégalité (mais pas les deux à la fois), une recherche aléatoire du signe ou du sens d'une inégalité peut être gagnante. Mais les cas sont bien plus rares. Au contraire, ce que l'on observe, c'est la correction immédiate des erreurs d'étourderie, des erreurs de calcul quand l'équivalence recherchée n'est pas obtenue. En revanche, l'information d'une équivalence barrée ne permet pas aux apprenants de corriger dans l'instant les erreurs importantes (erreurs algébriques). Face à un constat d'erreur, l'apprenant a besoin de temps pour arriver à trouver son erreur et la corriger, à moins qu'il n'abandonne son raisonnement courant pour partir dans une nouvelle direction.

15.7. Expérimentations

Le développement d'APLUSIX est récent puisqu'il a commencé au second semestre de l'an 2000. Dès que cela a été possible, il s'est accompagné d'expérimentations en classes, dès le début de 2001, avec un enseignant très motivé appartenant au projet, ensuite, avec des enseignants indépendants. Chaque début d'année scolaire a été l'occasion d'une grande expérimentation comprenant plusieurs classes sur un thème particulier. Au cours de chaque année, ont été effectués des suivis d'utilisation d'APLUSIX par des enseignants appartenant au projet. Par ailleurs, depuis le début de l'année 2002, des versions fonctionnelles du logiciel en cours de développement sont disponibles sur internet (<http://aplusix.imag.fr>) et plus de 5000 téléchargements des différentes versions de l'application ont été effectués. Il n'y a pas eu de suivi des usages produits par ces utilisations d'APLUSIX. Tests ou expérimentations et développements vont de pair certains développements ont précédé certaines expérimentations, ils ont été faits pour pouvoir réaliser ces expérimentations. Suite à ces expérimentations, ou suite à des retours sur l'utilisation d'APLUSIX, des choix ont pu être faits, des paramètres ont pu être figés, des fonctionnalités et des développements supplémentaires ont été mis en chantier.

Ainsi, à l'automne 2002, une grande expérimentation a été effectuée pour analyser les compétences générales des étudiants en algèbre. L'utilisation d'APLUSIX fut pensée comme celle d'un cahier de brouillon informatique en algèbre, la bonne ergonomie a permis ces expérimentations sans ajouter de biais liés à l'utilisation d'un logiciel particulier. Près de 300 élèves de fin de collège et début de lycée y ont participé. L'expérimentation se déroulait sur trois séances d'une heure réparties dans les premières semaines de septembre. La première et la troisième séance s'effectuaient avec un affichage de l'équivalence en permanence. Pour la seconde

séance, l'affichage de l'équivalence se faisait à la demande. Les enseignants avaient pour consigne de ne pas aider les élèves pour ce qui concernait les mathématiques. Ils pouvaient aider les élèves pour l'utilisation du logiciel, mais eurent, de leur point de vue, peu de travail, l'utilisation du logiciel étant suffisamment intuitive. Le travail des élèves fut enregistré sous forme de logs des micro-actions effectuées et analysé. L'analyse automatique de ces logs, a montré une progression du pourcentage de succès entre les trois situations, en particulier entre la seconde séance et la troisième. L'analyse a fourni aussi des résultats plus fins sur les succès en fonction du type d'exercice proposé, comme ceux donnés figure 4. Enfin, l'analyse manuelle, avec un magnétoscope des logs redonnant le travail des élèves micro-action par micro-action, a permis la construction d'un ensemble de règles de transformations algébriques erronées utilisées par les élèves, comme la règle $A+AB \rightarrow A(A+B)$ repérée dans des transformations (par exemple un élève qui transforme $(4x-8) + (4x-8) (6x-9)$ en $(4x-8) ((4x+8)+ (6x-9))$).

Type d'exercice	1 ^{re} S(a)	1 ^{re} S(b)
Calculer	86	82
Factorisation	83	81
Développement	66	73
Résolution d'équation	60	60
Résolution d'inéquation	45	48

Figure 4 : Réussite de deux classes en fonction du type d'exercice, lors de l'expérimentation de Septembre 2002.

Deux expérimentations, à l'automne 2003 et en Avril 2004, ont porté plus particulièrement sur l'évaluation des apprentissages avec APLUSIX. Plus d'un millier d'élèves était concerné. Pour chaque expérimentation, il y avait une séquence comportant 3 séances. La première et la dernière étaient des périodes dites de test (sans rétroaction sur l'équivalence), la seconde séance était une période dite d'apprentissage, avec rétroaction. La première expérimentation comportait une séance d'apprentissage d'une heure. Le taux de réussite -calculé comme nombre d'exercices correctement résolus par l'ensemble des élèves divisé par le nombre d'élèves- a plus que doublé entre le pré-test et le post-test. La séance d'apprentissage de la seconde expérimentation ne durait que 30 minutes. Le nombre moyen d'exercices bien résolus par heure est passé de 5,33 au pré-test à 7,02 au post-test, soit une augmentation de 31,7%.

D'autres résultats plus qualitatifs ont été obtenus après ces expérimentations, ou suite à l'utilisation d'APLUSIX en cours d'année. Les enseignants soulignent un gain en autonomie des élèves, associé à un engouement permettant de rattraper certains élèves en décrochage scolaire. L'utilisation d'APLUSIX est particulièrement appréciée pour les cas de remédiation et de correction des erreurs isolées. Les enseignants

observent aussi un changement de leur rôle avec APLUSIX, étant donné qu'APLUSIX dit ce qui est juste et faux. L'ordinateur semble jouer le rôle du *méchant* juge de paix, implacable, ne pouvant pas se tromper. L'enseignant se voit déplacé vers des activités d'explication de l'algèbre et de description des méthodes algébriques de résolution de problème. Bien que plébiscitée pour son ergonomie, sa simplicité d'utilisation, l'interface a été construite sur des choix qui ont pu introduire de nouveaux types d'erreur. Nous avons repéré celui-ci. Lors de la suppression d'une parenthèse, améliorant l'ergonomie et parce que cela a du sens du point de vue algébrique, APLUSIX fait disparaître automatiquement la parenthèse associée. Aussi, dans la situation particulière où l'expression entre parenthèses était précédée du signe moins, il y a l'effet suivant : $A-(B+C)$ devient $A-B+C$ et il n'est pas rare de constater que l'élève ne rétablit pas le signe de C. Mais faut-il gérer ce problème de distribution des signes lors de la disparition des parenthèses ? La question du transfert sur papier crayon se fait bien : des progrès notables sont faits sur la syntaxe (respect d'une écriture systématique, complète et plus juste des équations et des systèmes d'équations), ce qui entraîne la disparition de certaines erreurs. Nous notons cependant quelques bémols avec les 'bons' élèves qui pensent perdre leur temps en devant tout taper.

15.8. Conclusion

Ce chapitre a eu pour parti pris de se concentrer sur le cœur de l'environnement APLUSIX, les micromondes algébriques, leur ergonomie et les rétroactions épistémiques. Cette description a ainsi essayé de privilégier ce qui est stable et rodé, ce sur quoi nous avons le plus de recul et ce sur quoi nous avons eu le plus de temps pour travailler. Ces éléments forment le noyau d'APLUSIX et leur réalisation a commencé avec les premiers développements du projet.

Parmi les parties importantes peu décrites dans ce chapitre, il y a tous les à-côtés de l'EIAH destinés aux enseignants, à l'intégration de l'outil dans le déroulement d'une séquence didactique et le reste ... que ce soient les fonctionnalités d'administration des enseignants, des classes et des élèves ; que ce soient les éditeurs d'exercices et particulier les éditeurs d'exercices de modélisation ; que ce soit l'éditeur de paramètres, que ce soit l'éditeur de scénarios ; que ce soit la fenêtre de login ou d'inscription ; que ce soit le magnétoscope ; que ce soit le logiciel d'analyse automatique des logs ; que ce soient les fonctionnalités de notation ; que ce soient les fonctionnalités de génération automatique d'exercices ; que ce soit les conception d'un compagnon et de micro-tuteurs.

Sous forme de perspective, nous noterons, dans de nombreuses directions, un glissement progressif du micromonde vers le tuteur. En particulier, cela s'observe à

travers la définition d'un compagnon⁴ et d'un micro-tuteur qui sont respectivement destinés à proposer une résolution type d'un exercice et à corriger des erreurs typiques observées dans un raisonnement proposé. Cette évolution du projet APLUSIX est liée à l'introduction de la notion de type de problème. Ajoutée initialement pour fournir un guidage stratégique, cette notion a servi ensuite de support pour l'adjonction des fonctionnalités de type exerciceur. Progressivement, nous nous acheminons vers la conception d'un EIAH de l'algèbre de type tuteur placée par-dessus le micromonde et d'utilisation facultative.

Pour conclure, le pari de la réorientation vers une approche micromonde à l'ergonomie soignée pour aller vraiment en classe est aujourd'hui gagné, en témoignent le nombre des téléchargements et les opinions d'utilisateurs (enseignants et élèves) que nous recevons depuis deux ans, dont certaines peuvent être lues à l'adresse <http://aplusix.imag.fr>

Références

[ACT 02] The ActiveMath group : MELIS E., BÜDENBENDER J., FRISCHAUF A., GOGUADZE G., LIBBRECHT P., ULLRICH C., *Using Computer Algebra Systems as Cognitive Tools*, Proceedings of ITS'2002, Intelligent Tutoring Systems, Biarritz, France, LNCS 2363, p. 802-810, ISBN 3-540-43750-9, Ed. Springer-Verlag, 2002.

[BAL 94] BALACHEFF, N., SUTHERLAND, R., *Epistemological domain of validity of micro-worlds, the case of Logo and Cabri-géomètre*, in R. Lewis and P. Mendelshon (eds.), Proceedings of the IFIP TC3/WG3.3, Lessons from learning, p. 137-150, North-Holland, 1994.

[BEE 96] BEESON M., *Design principles of mathpert: Software to support education in algebra and calculus*, in N. Kajler, (ed.), Human Interfaces to Symbolic Computation, ISBN 3-211-82843-5, Ed. Springer-Verlag, 1996.

⁴ Il s'agit d'un réinvestissement du travail effectué sur le projet APLUSIX entre 1980 et 2000.

[BUC 02] BUCHBERGER B., *Teaching Without Teachers?* in J Böhm (ed.), Proceedings of the Vienna International Symposium on Integrating Technology in Math Education, VISIT-ME'2002, ISBN 3-901769-49, Ed. BK Teachware, Vienna, Austria, 2002.

[ERI 02] ERICKSON T., *Learning Math with Graphing Calculator*, <http://www.PacificT.com>, 2002.

[JEA 97] JEAN S., DELOZANNE E., JACOBINI P., GRUGEON B., *Conception, réalisation et évaluation d'interfaces en EIAO : l'exemple de PÉPITE*, Actes des 5èmes journées EIAO de Cachan, p. 37-48, ISBN 2-86601-612-2, Ed. Hermès, Paris, 1997.

[KOE 97] KOEDINGER K.R., ANDERSON J.R., HADLEY W.H., MARK M.A., *Intelligent tutoring goes to school in the big city*, International Journal of Artificial Intelligence in Education Vol. 8, p. 30-43, ISSN 1560-4292, 1997.

[LAB 89] LABORDE J.M., *Designing intelligent tutorial systems: The case of geometry and cabri-géomètre*, IFIP WG 3.1 Working Conference on Educational Software at the Secondary Education Level, Reykjavik, Islande, 1989.

[LAG 02] LAGRANGE J.B., *Casyopée, a computer algebra environment for secondary students*, in J Böhm (ed.), Proceedings of the Vienna International Symposium on Integrating Technology in Math Education., VISIT-ME'2002, ISBN 3-901769-49, Ed. BK Teachware, Vienna, Austria, 2002.

[NIC 90] NICAUD J.F., AUBERTIN C., NGUYEN-XUAN A., SA M., WACH P., APLUSIX: A learning environment for student acquisition of strategic knowledge in algebra, Proceedings of PRICAI'90, Nagoya, Japon, 1990.

[NGU 02] NGUYEN-XUAN A., NICAUD J.F., BASTIDE A., SANDER E., *Les expérimentations du projet APLUSIX*, Sciences et Techniques Educatives, N1-2 Vol 9, p. 63-91, ISBN 2.7462.0438.X, Ed. Hermès, Paris, 2002.

[NIC 01] NICAUD J.F., BOUHINEAU D., *Syntax and semantics in algebra*, Proceedings of the 12th ICMI Study Conference, University of Melbourne, Australia, 2001.

[NIC 04] NICAUD J.F., BOUHINEAU D., CHAACHOUA H., *Mixing Microworld and CAS Features for Building Computer Systems that Help Students to Learn Algebra*, International Journal of Computers for Mathematical Learning, Vol. 9, p. 169-211, Ed. Kluwer Academic Publisher, 2004.

[PRA 01] PRANK R., TÖNISSON E., *Is the Domain Expert Module for Expression Manipulation Exercises Ready?* Proceedings of the Tenth International PEG Conference, p. 51-56, Tampere, Finlande, 2001.

[RIC 68] RICHARDSON D., *Some Unsolvable Problems Involving Elementary Functions of a Real Variable*, Journal of Symbolic Logic, Vol 33, pp. 514-520, ISSN 00224812, 1968.

[TAR 51] TARSKI A., *A decision method for elementary algebra and geometry*, Ed. University of California Press, Berkeley, Californie, 1951.

[TRO 02] sous la direction de TROUCHE L., GUIN D., *Les calculatrices symboliques, transformer un outil en un instrument du travail informatique : un problème didactique*, ISBN 2.85919.180.1, Ed. La pensée sauvage, Grenoble, 2002.