



HAL
open science

ENSEIGNEMENT DE L'ALGORITHMIQUE EN CLASSE DE SECONDE : UNE INTRODUCTION CURRICULAIRE PROBLÉMATIQUE

Michèle Couderette

► **To cite this version:**

Michèle Couderette. ENSEIGNEMENT DE L'ALGORITHMIQUE EN CLASSE DE SECONDE : UNE INTRODUCTION CURRICULAIRE PROBLÉMATIQUE. *Annales de Didactiques et de Sciences Cognitives*, 2016, *Annales de Didactiques et de Sciences Cognitives*, 21, pp.267-296. hal-01926799

HAL Id: hal-01926799

<https://hal.science/hal-01926799>

Submitted on 19 Nov 2018

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.

Article paru dans les ANNALES de DIDACTIQUE et de SCIENCES COGNITIVES,
volume 21, p. 267-296.

Citer COUDERETTE, M. (2016) Enseignement de l'algorithmique en classe de seconde : une introduction curriculaire problématique, in *Annales de didactiques et de sciences cognitives*, 21, 267-296

MICHELE COUDERETTE

ENSEIGNEMENT DE L'ALGORITHMIQUE EN CLASSE DE SECONDE : UNE INTRODUCTION CURRICULAIRE PROBLEMATIQUE

Résumé. La réforme des lycées de 2009 a introduit l'enseignement de l'algorithmique en cours de mathématiques. Il s'agit d'un nouvel objet hybride, au carrefour des mathématiques et de l'informatique, pour lequel la plupart des enseignants ne sont pas formés. Dès lors, comment les enseignants abordent-ils cette partie du programme ? Pour répondre à cette question, nous avons observé une enseignante de mathématiques dans une classe de seconde, durant une année scolaire. Afin de mener une analyse, nous articulons deux cadres théoriques : la théorie anthropologique du didactique (TAD) et la théorie de l'action conjointe en didactique (TACD). La TAD fournit des outils pour analyser les tâches que l'enseignante propose aux élèves sous couvert du concept de praxéologies mathématiques. La TACD permet de mettre en regard les attentes de l'institution avec le savoir réellement enseigné dans la classe au fil de plusieurs séances d'enseignement de l'algorithmique. Les résultats montrent les difficultés d'enseignement d'un savoir dont les références appartiennent à deux disciplines. Les savoirs mis à l'étude sont référés soit à leur spécificité mathématique soit à leur spécificité informatique. Du coup, les élèves n'entrent pas dans l'étude des savoirs réellement algorithmiques.

Mots-clés. Algorithmique – Mathématiques – Analyse de pratiques enseignantes – TACD – TAD

Abstract: In 2009, the high school curricular reform introduced algorithms in mathematics teaching. This is a new hybrid object, at the crossroads of mathematics and computer science, for which most teachers are not trained. Therefore, how do teachers approach this part of the program? To answer this question, we observed a mathematics teacher in a second grade in high school, for one year school. In order to conduct an analysis, we articulate two theoretical frameworks: the anthropological theory of didactics (ATD) and the joint action theory in didactics (JATD). The ATD provides tools for analyzing the tasks proposed by the math teacher to the students against the background of the concept of mathematical praxeologies. The JATD allows to put in view the expectations of the institution with what is actually taught in the classroom over several sessions of algorithm teaching. The results show the difficulties of teaching a knowledge that has its references in two disciplines. The co-constructed knowledge developed in the class is referred either under their mathematical specificity or their computer specificity. As a result of that, students do not really learn algorithmic knowledge.

Keywords. Algorithms - Mathematics - Analysis of teaching practices - JATD – ATD

Introduction

Cet article traite de l'introduction de l'algorithmique en mathématiques en classe de seconde. Des recherches ont déjà été menées sur ce sujet (Modeste, 2012 ; Modeste, Gravier, Ouvrier-Bufferet, 2010 ; Haspekian, Nijimbéré, 2012) mais peu ont porté sur l'analyse de pratiques ordinaires d'enseignement. Nous rendons compte ici d'une recherche s'appuyant sur l'observation d'une classe de seconde lors d'un enseignement d'algorithmique en mathématiques au moment de son introduction dans les nouveaux programmes (MEN, 2009a)

L'enseignement de l'algorithmique, en tant qu'objet d'étude, a été introduit dans les universités françaises dans les années 1950 (Chi Thanh N'Guyen ; 2005). En 2009, cet enseignement est ensuite apparu dans les programmes scolaires de lycée dans le cours de mathématiques. Auparavant, outil au service des autres domaines enseignés dans la discipline mathématique, l'algorithmique est devenu un objet d'étude.

A la suite d'une enquête (Coudерette, 2013) auprès d'enseignants en classe seconde montrant leur difficultés à enseigner cet objet d'étude et compte tenu de la complexité et du caractère interdisciplinaire de ce dernier, nous avons pensé utile de documenter la manière dont un enseignant mettait en œuvre cette partie du programme¹. Pour effectuer cette analyse des pratiques ordinaires en classe, nous avons dans un premier temps fait une étude épistémologique à propos du savoir algorithmique. Ceci nous a permis de mener une analyse critique du curriculum tel qu'il a été proposé en 2009. C'est ce qui fera l'objet de la première partie de cet article. Dans un deuxième temps, fort de l'analyse épistémologique et curriculaire effectuée, nous sommes allés observer une enseignante participante à l'enquête ayant accepté de nous recevoir dans sa classe durant l'année scolaire 2010-2011. Cette étude de cas fera l'objet de la seconde partie de cet article.

1. De l'analyse épistémologique aux choix curriculaires actuels

Ainsi que le souligne Artigue (1988), une analyse d'ordre épistémologique est un préalable nécessaire à l'analyse de curriculums et de pratiques enseignantes. Les travaux de Knuth (1968), auteur faisant référence dans la communauté des mathématiciens informaticiens, permettent d'éclairer la manière dont cet objet est transposé dans les textes d'accompagnement en vigueur ainsi que dans les pratiques de classe sur le terrain.

1.1. Contexte d'introduction de l'algorithmique dans le curriculum 2009

Initialement objet mathématique, l'algorithme est maintenant souvent perçu comme un objet de l'informatique associé à la programmation. Dans son ouvrage, « Histoire d'algorithmes. Du caillou à la puce », Chabert (2010) montre l'évolution du concept d'algorithme jusqu'au développement de l'informatique, informatique qui a largement contribué à la formalisation des algorithmes et à l'élargissement du domaine de recherche des mathématiques. L'algorithmique est devenue à elle seule

¹ La recherche présentée dans cet article relève d'une approche descriptive à visée compréhensive afin de mettre en évidence les contraintes et les possibles du fonctionnement d'un système didactique aux prises avec l'enseignement d'un objet, pensé comme interdisciplinaire dans les textes du programme mathématique. Dans cette perspective, la posture du chercheur s'attache à suspendre autant que possible son jugement sur les pratiques observées.

un domaine de recherche, au carrefour de l'informatique et des mathématiques. Reconnu par la société civile, dans les différentes institutions politiques, économiques, industrielles, de recherche, ou d'enseignement, ce savoir algorithmique est devenu enjeu d'enseignement. Le groupe d'experts de l'Association ITIC-ASTI² milite pour un enseignement de l'informatique au secondaire ne se réduisant pas à une utilisation des outils informatiques : « *Un tel enseignement est de nature à créer les conditions d'une bonne utilisation des outils informatiques dans les autres disciplines de par la maîtrise qu'il contribuerait à donner aux élèves. De ce point de vue aussi, sa valeur ajoutée est donc indispensable. « Objet » et « outil » d'enseignement, loin de s'opposer, sont complémentaires et se renforcent mutuellement* » (Association ITIC-ASTI, 2007). Une proposition de programme est présentée par Dowek en 2008, proposition qui introduit l'informatique à tous les niveaux de scolarité : « *Placer le programme des lycées dans une vision pour l'ensemble du cursus en informatique de la maternelle à l'Université avec une phase en amont du lycée centrée autour de la maîtrise de logiciels [...], une phase au Lycée centrée autour de l'apprentissage des connaissances nécessaires à l'écriture d'un programme simple [...], et une phase en aval du lycée consacré aux diverses branches de l'informatique en tant que discipline scientifique* ». En 2009, suivant ainsi les conseils de la commission Kahane (2002) d'introduire « *une part d'informatique dans l'enseignement des sciences mathématiques et dans la formation des maîtres* », le ministère fait le choix d'inclure dans tous les programmes de mathématiques de lycée et à tous les niveaux, un enseignement d'algorithmique.

Le projet et les objectifs de l'enseignement de l'algorithmique dans le nouveau programme de mathématiques sont d'amener à « *une formalisation en langage naturel propre à donner lieu à traduction sur une calculatrice ou à l'aide d'un logiciel.* » et ainsi de « *de familiariser les élèves avec les grands principes d'organisation d'un algorithme : gestion des entrées-sorties, affectation d'une valeur et mise en forme d'un calcul* » (MEN, 2009a, p. 9). Le programme précise que cet enseignement ne doit pas être déconnecté de l'enseignement des mathématiques : « *au contraire, l'introduction de chaque nouvel élément (variable, boucle, itération, etc.) devrait apparaître lors de la résolution de problèmes pour lesquels les démarches habituelles sont malcommodes ou peu performantes* » (MEN, 2009b). Cette spécification qui amène donc à travailler à la fois les mathématiques et l'informatique ne nous semble pas aller de soi. Les discussions dans la communauté des professeurs de mathématiques ont montré un certain nombre de difficultés des enseignants à mettre en œuvre ce nouveau programme sur l'algorithme. L'APMEP³ signale dans son bulletin qu'un « *déficit de formation est déclaré de façon explicite lorsqu'il s'agit d'évoquer les nouveaux thèmes introduits dans le programme, notamment dans le cadre de l'algorithmique. Ce manque de formation peut être repéré de façon implicite cette fois à travers certaines difficultés que rencontrent les enseignants à interpréter le programme.* » (APMEP, 2014). Le rapport de la commission de suivi des programmes de 2014 montre que l'introduction de l'algorithmique est « *jugée positive par 65% des*

² ITIC-ASTI : Informatique et TIC - Association française des Sciences et Technologies de l'Information. Groupe animé par des informaticiens, Maurice Nivat, Gilles Dowek, Éric Bruillard et Monique Grandbastien notamment.

³ APMEP : Association des Professeurs de Mathématiques de l'Enseignement Public

répondants, [mais que] plus de 40% des répondants ne sont pas satisfaits de son écriture dans le programme officiel ; c'est la partie du programme qui est jugée la moins explicite. Les entretiens laissent penser que les relations entre algorithmique, mathématiques et programmation ne sont pas clairement perçues. » (MEN-DGESCO, 2014). Si les recherches de Haspékian et Nijimbéré confirment les observations du ministère, ils relèvent par ailleurs que « les textes officiels n'aident pas les enseignants à faire la part des choses précisément entre ces différents points de vue, à délimiter clairement ce qui relève de l'algorithmique au niveau des raisonnements de ce qui relève des nécessités de son implémentation technologique » (2012, p. 270). Il apparaît donc ici que l'introduction de ce nouvel objet d'enseignement paraît complexe. En quoi l'analyse épistémologique peut-elle éclairer cette complexité ?

1.2 Analyse épistémologique de ce domaine d'étude

Les travaux de Knuth, chercheur que l'on pourrait qualifier de « mathématicien-informaticien », sont reconnus à la fois par la communauté des mathématiciens et par celle des informaticiens. Professeur à l'université de Standford (États-Unis), Knuth a contribué autant aux mathématiques qu'à l'informatique : ses travaux de recherche ont porté sur les mathématiques discrètes et sur l'algorithmique. Son ouvrage *The Art Of Computer Programming (TAOCP)* date de 1968. Pour autant, malgré l'évolution de l'informatique, celui-ci reste toujours d'actualité et fait encore référence dans la communauté des chercheurs du domaine. Dans cet ouvrage, Knuth développe la vision qu'il a d'un algorithme. Celui-ci n'est *ni que* informatique, *ni que* mathématique : *il est des deux*. Dans cet ouvrage, tous les algorithmes présentés s'appuient sur des énoncés mathématiques. Pour autant, l'objectif n'est pas de faire un cours de mathématiques, mais de faire comprendre à un lecteur « novice » en informatique ce qu'est un algorithme. Aussi, donne-t-il à son propos une forte dimension pédagogique en décrivant des concepts algorithmiques au versant plutôt informatique et en particulier en pointant des concepts algorithmiques potentiellement difficiles à comprendre du fait de leur proximité avec des concepts informatiques (le stockage dans une variable par exemple). Nous cherchons à mettre en tension la vision hybride de l'algorithmique développée par Knuth avec celle que l'on peut deviner à la lecture des documents officiels (programme et document ressources) afin de faire émerger ce qui a été retenu pour constituer le savoir à enseigner.

Les quatre sous-parties ci-dessous rendent de concepts de base et de difficultés pointées par Knuth dans le premier volume « *Fundamental Algorithms* » du TAOCP (1968) : (1) définition et caractéristique, (2) texte de communication, (3) difficultés potentielles (4) Validation. Nous reprenons ces quatre points dans la section relative à l'analyse des textes officiels.

1.2.1. Une définition et cinq caractéristiques

S'appuyant sur plusieurs exemples, tous mathématiques, Knuth définit et caractérise ce qu'est un algorithme par « a finite set of rules which gives a sequence of operations for solving a specific type of problem » (Knuth, 1968 ; p. 4).

En utilisant les cinq caractéristiques d'un algorithme, finitude (finiteness), définition précise (definiteness), effectivité (effectiveness), entrée (input) et sortie (output),

Knuth (1968) nous met en alerte sur la présentation de processus de vie courante comme des algorithmes : une recette de cuisine, qui si elle respecte bien la propriété de séquentialité, les critères d'entrée, de sortie et de finitude, ne respecte pas les critères de « définition précise » et d'« effectivité ». Un algorithme doit fournir un même résultat quelque soit l'opérateur le traitant.

1.2.2. Un texte de communication

Knuth (1968) formalise d'une manière très précise l'écriture des algorithmes (cf. ci-dessous), montrant ainsi l'importance qu'il donne à l'aspect communicationnel entre celui qui écrit l'algorithme et celui qui le lit : nom de l'algorithme, début, fin, numérotation des différentes étapes, résumés et commentaires, sont autant d'éléments composant un algorithme permettant au lecteur d'en avoir une meilleure compréhension : « they are only for the reader's benefit as possible aids to comprehension » (Knuth, 1968, p. 3).

```

Algorithm NomAlgo : Given liste des variables
NomAlgo1 [résumé de l'action 1] action 1 (commentaire 1)
NomAlgo2 [résumé de l'action 2] action 2 (commentaire 2)
.....
NomAlgon [résumé de l'action 2] action 2 (commentaire 2) |

```

Figure 1 : Écriture formelle d'un algorithme selon Knuth (1968).

L'algorithme d'Euclide formalisé par Knuth se présente ainsi :

```

Algorithm E (Euclid's algorithm). Given two positives
integer  $m$  and  $n$ , find their greatest common divisor, i.e.,
the largest positive integer witch evenly divides both  $m$  and
 $n$ 
E1 [Find remainder] divide  $m$  by  $n$  and let  $r$  be the remainder
(we will have  $0 \leq r < n$ )
E2 [is it zero?] If  $r = 0$ , the algorithm terminates;  $n$  is
the answer.
E3 [Interchange] Set  $m \leftarrow n$ ,  $n \leftarrow r$ , and go back to step E1. |

```

Figure 2 : Écriture formelle de l'algorithme d'Euclide selon Knuth (1968)

1.2.3. Des difficultés potentielles lors de l'enseignement de l'algorithmique

- *L'affection et la variable, enjeux de savoir problématiques*

Knuth pointe plusieurs difficultés de compréhension de certains objets et concepts résultants de signifiants différents selon qu'ils sont utilisés dans un cadre exclusivement mathématique⁴ ou dans un cadre algorithmique ; il désigne par

⁴ Pour des commodités de langage, nous écrivons « mathématique » lorsque nous désignerons tout ce qui relève exclusivement des mathématiques (c'est à dire ne relevant

exemple le concept « d'affectation » comme l'un des plus difficiles à comprendre, celui-ci renvoyant à la notion de variable informatique : une variable informatique désigne un emplacement dans la mémoire de l'ordinateur et nous comprenons alors, que même si le vocabulaire est le même, le mot variable dans un contexte algorithmique relève ici de l'informatique et non des mathématiques. Le mot « variable » représente donc des concepts différents selon qu'il est utilisé en mathématique ou en informatique. Du point de vue informatique, une variable ne peut contenir qu'une seule valeur à un instant t , qui pourra ensuite évoluer selon les instructions de l'algorithme. En mathématique, une variable revêt une tout autre signification : par exemple, dans l'écriture $x \in N$, x représente n'importe quelle valeur de N . Knuth conseille à ses lecteurs d'étudier des algorithmes déjà écrits et de les dérouler à la main pour comprendre leur fonctionnement, en particulier lors des opérations d'affectation. Pour des algorithmes plus complexes (mettant en jeu une boucle par exemple), « suivre à la trace » des variables suppose que l'on ait une technique de traçage. Nous présentons ci-dessous à titre d'exemple la trace de l'algorithme d'Euclide, nécessaire pour la gestion des variables :

Déroulement		Etapes	m	n	r	
Entrées		E0	28	72	?	
Traitement	1 ^{er} passage dans la boucle	E1	28	72	28	
		E2	28	72	28	
		E3	72	28	28	
	2nd passage dans la boucle	E1	72	28	16	
		E2	72	28	16	
		E3	28	16	16	
	3 ^{ème} passage dans la boucle	E1	28	16	12	
		E2	28	16	12	
		E3	16	12	12	
	4 ^{ème} passage dans la boucle	E1	16	12	4	
		E2	16	12	4	
		E3	12	4	4	
	5 ^{ème} passage	E1	12	4	0	
	Sortie		E2	12	4	0

Figure 3 : Trace de l'algorithme d'Euclide pour les valeurs 28 et 72 selon Knuth (1968)

- L'égalité en mathématiques et en informatique

Un autre concept pointé par Knuth comme difficile à comprendre est le signe « = ». Selon qu'il est utilisé en mathématique ou en algorithmique, il n'a pas la même signification. Dans le domaine mathématique, il symbolise l'expression « est le résultat de... » ou décrit une relation d'équivalence entre deux objets de même nature, ou encore traduit une assertion logique ayant la valeur de vrai ou faux, alors que dans le domaine algorithmique le signe « = » est utilisé dans les tests d'égalité et donc revêt uniquement le statut d'assertion logique. Les travaux de Knuth rejoignent les nombreuses recherches en didactique de l'algèbre portant sur les différentes significations du signe « = » selon leur contexte d'utilisation. On citera,

pas de l'algorithmique) et par « informatique » tout ce qui relève exclusivement de l'informatique (et non de l'algorithmique).

en autres, Chevallard (1985), Kieran (1994), Gascon (1994), Artigue (2002) et plus récemment Knuth et al. (2006).

1.2.4. Validation d'un algorithme

Poursuivant sa volonté de montrer le caractère hybride des algorithmes, Knuth s'intéresse à leurs validations. Il montre qu'une démonstration de type expérimental est insuffisante et qu'une démonstration mathématique est nécessaire. Il présente alors deux méthodes : l'induction mathématique et la preuve de programme par assertions. Toutes deux reposent sur des domaines mathématiques : la démonstration par récurrence pour la première et la logique formelle pour la seconde (Hoare, 1969). Si la logique formelle n'est pas au programme de lycée, la démonstration par récurrence est possible dans les classes scientifiques.

Comment alors la transposition opère-t-elle dans les documents institutionnels ? Les documents institutionnels montrent-ils la double appartenance – l'hybridité – de l'algorithmique ou présentent-ils principalement son versant informatique ? Les difficultés pointées par Knuth sont-elles explicitement indiquées ? Quelles propositions sont alors avancées pour les contourner ?

1.3. Les choix curriculaires effectués dans le programme 2009

Tout d'abord, il nous semble important de signaler que contrairement aux autres savoirs enseignés en mathématiques, le texte du savoir à enseigner l'algorithmique n'a pas été découpé en fonction des niveaux seconde, première et terminale. Il est décrit dans le programme de seconde et est établi pour les trois années de lycée. (MEN, 2009a). Nous considérons que cette place « à part » dans les programmes lui donne un statut « à part ». D'autre part, il est clairement indiqué que l'algorithmique a une visée doublement transversale : elle doit s'insérer dans chacune des trois parties du programme de mathématiques et s'articuler avec les autres disciplines enseignées : « *l'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes posés doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante* » (MEN, 2009a, p. 9).

Dans cette section, nous rendons compte de quelques points saillants d'une analyse critique des documents institutionnels (programme de seconde, documents ressource) en reprenant point par point les éléments dont nous avons rendu compte dans la section précédente.

1.3.1. Une définition floue de l'algorithme

Knuth insiste sur les propriétés de *definiteness* et *effectiveness* d'un algorithme, au point de prendre pour exemple une recette de cuisine. Nous pouvons alors nous interroger sur la pertinence d'introduire les algorithmes à l'aide de situations de vie courante, qui de part leur nature, ne respecte pas les propriétés de *definiteness* et d'*effectiveness*.

Des exemples souvent repris pour illustrer ces différents aspects, comme le rendu de monnaie pour le volet numérique, ou encore les recettes de cuisine. On trouve encore des algorithmes dans des situations de la vie courante (s’habiller) ou professionnelle (ainsi, la conduite d’un train, la consultation d’un catalogue de bibliothèque, etc.).

...

Ainsi, dans la recette de la pâte à crêpes, on trouve la séquence d’instructions (mettre la farine dans le bol ; faire un puits ; mettre les œufs dans le puits)

Figure 5 : extrait du document ressources en algorithmique (MEN (2009b), pp. 6-8)

1.3.2. Aspect communication du texte algorithmique

C’est au travers des préconisations pour l’évaluation que nous trouvons trace des principes de Knuth. Dans le document-ressources, des critères d’évaluations relatifs à la lisibilité de l’algorithme sont définis : explicitation du but de l’algorithme, commentaires précisant le déroulement, noms des variables bien choisis font partis des critères permettant d’évaluer un algorithme non pas seulement sur son aspect fonctionnel mais aussi sur son aspect communicationnel.

Critère	Excellent	Bon	Moyen	Insuffisant
Respect des bons usages Le but visé par l’algorithme est explicité, et des commentaires précisent le déroulement. Les variables ont des noms bien choisis.	Aucune erreur	De petits détails sont négligés. Le but est difficile à déterminer	Des détails manquent, mais le programme tente quand même d’accomplir ses fonctions essentielles.	Ne répond pas au problème posé. Objectif impossible à déterminer

Figure 6 : Extrait du document ressources en algorithmique (MEN (2009b), p. 13)

Pour autant, l’aspect communication est-il réellement présent dans le texte de programme ? Dans la plupart des algorithmes présentés, seules les variables déclarées sont commentées et non les différentes actions. L’algorithme 8 du document ressource (MEN (2009b), (affichage automatique d’une fenêtre graphique), certes complexe, est pratiquement le seul à contenir des commentaires pour les différentes étapes du traitement. Le nombre d’algorithmes du document traduits en langage informatique (calculatrices, applications logicielles) peut implicitement laisser à penser que l’aspect fonctionnel du programme est privilégié. Cette perception, plus informatique qu’algorithmique, laisse en réalité peu de place à la compréhension même des objets algorithmiques.

1.3.3. Des difficultés potentielles dans l’enseignement de l’algorithmique

- Affectation et variable, enjeux de savoir problématiques

Si l’affectation et la notion de variable ne sont pas soulignées comme des difficultés dans le programme, on peut supposer qu’elles sont implicitement suggérées par la description imagée « *le rangement d’un objet dans un petit tiroir ne pouvant contenir qu’un objet à la fois* » (MEN, 2009b, page 7). L’activité de traçage signalée comme nécessaire par Knuth (1968) est complètement absente dans le programme. Il nous semble alors possible de dire que, du fait de cet

absence, la transposition didactique dans le programme actuel d'un concept purement algorithmique tel que l'affectation dans une variable, ne permet pas d'en traiter le sens.

Nous retrouvons toutefois une des préoccupations de Knuth relative à la compréhension du fonctionnement d'un algorithme dans les textes du savoir à enseigner : étudier des algorithmes afin d'en déterminer leurs fonctions ou de déceler des erreurs dans leurs conceptions. Mais nulle indication sur une technique d'étude d'un algorithme. Est-il implicitement convenu que l'implémentation d'un algorithme dans une calculatrice ou application logicielle pourra rendre possible la détection d'une erreur et ainsi de la corriger ? Des techniques sont envisageables (mettre des tests d'arrêt par exemple, suivre à la trace le contenu des variables) mais difficiles à mettre en œuvre sans une compréhension au préalable des concepts algorithmiques.

- L'égalité en algorithmique et en mathématique, écueil potentiel

Le concept d'égalité en algorithmique n'est pas non plus signalé comme possible difficulté. Cet écueil est complètement occulté dans les programmes. Jamais les enseignants n'ont leur attention attirée sur le fait que ces signifiants différents selon que le concept soit abordé sous couvert des mathématiques ou sous couvert de l'informatique, sont porteurs potentiels de difficultés. On peut donc anticiper que les enseignants seront confrontés à ces difficultés lorsqu'ils auront à enseigner ce concept.

1.3.4. La validation d'un algorithme : une absence dans les programmes

La preuve d'un algorithme n'est pas explicitement mentionnée dans le programme. On peut voir une explication dans le fait que les algorithmes écrits sont en lien direct avec les notions mathématiques étudiées dans le cours de mathématiques. Le risque est alors que pour les élèves la notion de preuve d'algorithme n'existe pas et que seule la machine ou l'application logicielle valide l'algorithme.

En conclusion de cette analyse, il ressort que les concepteurs des programmes d'algorithmique partagent formellement la même vision des algorithmes que Knuth : des algorithmes reposant sur des concepts mathématiques mais aussi sur des concepts informatiques. Néanmoins, les documents institutionnels ne permettent pas, par élision, de traiter du sens des concepts purement algorithmiques signalés par Knuth : certaines difficultés ou écueils signalés par cet auteur, bien que présents allusivement dans ce programme, semblent laisser les enseignants démunis face à l'organisation didactique et/ou mathématique de la séance, comme l'a mis en évidence l'enquête évoquée en introduction.

Au terme de cette analyse transpositive d'un domaine nouvellement objet d'étude, examinons les mises en œuvre possibles dans le cours de mathématiques.

2. Analyse des pratiques d'enseignement de l'algorithme en classe de seconde : une étude de cas

Nous nous intéressons à la manière dont une enseignante met en œuvre cet objet d'enseignement compte tenu de sa spécificité c'est à dire celui d'être un objet interdisciplinaire (mathématique et informatique) et qui par conséquent vise à travailler à la fois les mathématiques et l'informatique. L'analyse épistémologique

et curriculaire permet de préciser les questions de recherche : (i) quel est le savoir réellement mis à l'étude dans le cours d'algorithmique ? (ii) comment ce savoir évolue-t-il au sein du système didactique ? (iii) comment les enseignants jouent-ils avec les contraintes d'un enseignement sous couvert de deux disciplines ?

Cette problématique suppose deux niveaux d'analyse, celle des praxéologies ici de type interdisciplinaires et celle des pratiques en classe.

2.1. Cadres théoriques

L'étude de cas a été menée sous couvert de deux cadres : le premier, permettant de mettre en exergue le savoir réellement mis à l'étude, s'appuie sur le concept de praxéologie développé dans la théorie anthropologique du didactique – TAD – (Chevallard ; 1999), le second rendant compte de la co-construction de ce savoir au sein du système didactique, s'appuie sur les études de l'action conjointe (Sensevy ; 2007). La compatibilité de ces deux cadres théoriques a déjà été discutée dans plusieurs travaux dont ceux de Brière-Guenoun et Amade-Escot (2010), Athias (2014), Salone (2015).

Pour décrire et étudier les conditions de réalisation des pratiques institutionnelles, la TAD offre un outil d'analyse que nous utilisons pour étudier les savoirs effectivement mis à l'étude par une enseignante dans une classe de seconde : les organisations praxéologiques. Les outils praxéologiques permettent de décrire une séance en terme d'unité, chaque unité correspondant à une tâche dans laquelle l'élève est engagé. Ils permettent de déterminer les savoirs mis à l'étude, et ainsi de répondre aux questions : (i) quel est l'apprentissage visé dans la séance ? (ii) l'algorithmique étant un objet interdisciplinaire, à quelle discipline de référence se réfère l'enseignante quand elle propose une tâche aux élèves ?

La théorie de l'action conjointe en didactique – TACD – pose l'action didactique comme une résultante des interactions élèves-professeur. L'analyse de l'action conjointe permet de montrer comment, dans ces interactions, les praxéologies, telles que proposées par le professeur, vivent au sein de la classe. Dans notre analyse, nous utilisons les genèses {topogenèse, chronogenèse, mésogenèse} décrivant la dynamique évolutive du système didactique : (iii) quels sont les objets constituant le milieu, sachant que ceux-ci sont évolutifs dans la co-construction du savoir, (iv) à quel rythme ce savoir est-il conduit et (v) quelles places élèves et professeur occupent-ils vis à vis du savoir construit ?

2.2 Indications de méthode

Contexte et participants : La classe observée est une classe de seconde générale d'un lycée dans la banlieue toulousaine. D'un niveau moyen en mathématiques, elle est constituée de 33 élèves (23 garçons et 9 filles) âgés de 15 à 16 ans. L'enseignante a une ancienneté de 25 ans dans l'enseignement et a suivi une formation en algorithmique dans son cursus universitaire alors qu'elle était étudiante.

Recueil des données : Nous avons observé et filmé trois séquences d'enseignement chacune étant composée de trois séances de 55 minutes. Respectant le protocole de Leutenegger (2009), chaque séance a été précédée d'un entretien *ante* puis suivie d'un entretien *post*, tous deux enregistrés. Les entretiens

ante ont pour objectif de recueillir l'intention didactique de l'enseignante et les entretiens *post* de recueillir ses réactions 'à chaud' quant au déroulement du cours. Des notes écrites « prises au vol » ainsi que les documents distribués aux élèves ont complété ce recueil.

Analyse des données : les deux cadres théoriques précités sont mobilisés pour l'analyse des données.

- Nous appuyant sur les travaux de Chevallard (1998) et de Bosch et Gascon (2004), nous avons cherché à déterminer pour chacune des tâches proposées aux élèves si la référence était d'ordre mathématique, algorithmique ou informatique. Pour cela nous avons fait appel au modèle $[T/\tau/\theta/\Theta]$ de la TAD qui décrit l'organisation praxéologique des tâches proposées aux élèves : accomplir une tâche t relevant d'un type de tâche T nécessite la mise en œuvre d'une technique τ , justifiée par une technologie θ qui est elle-même justifiée par une théorie Θ .
- Suivant les préconisations de Leutenegger (2009), l'ensemble des données vidéos ont été transcrites puis condensées dans des synopsis nous permettant d'obtenir une vue d'ensemble des séances observées. Chaque synopsis présente sur une échelle temporelle un découpage de la séance en termes de tâches, de modalités de travail fixées par l'enseignante, de repérage d'épisodes remarquables ou significatifs pour l'analyse didactique. Les interactions élèves-professeur sont ensuite analysées au travers de trois genres {mésogénèse, chronogénèse, topogénèse}.

Dans le cadre de cet article, nous nous appuyons sur deux exemples prototypiques de ce que nous avons observé et que nous développons en deux temps : l'analyse des praxéologies algorithmiques puis l'analyse de la coconstruction des savoirs mis à l'étude.

2.3. Analyse des praxéologies algorithmiques en classe de mathématiques

L'analyse praxéologique du savoir mis à l'étude a été conduite afin de pouvoir déceler si des éléments relevés par Knuth comme problématiques dans leur enseignement (aspect hybride de l'algorithmique, difficultés concernant les notions d'affectation, de variable, d'égalités, de validation) étaient perceptibles dans les dispositifs proposés par l'enseignant.

2.3.1 Première séquence : des algorithmes pour consolider des savoirs mathématiques

L'enseignante poursuit « officiellement » dans cette séquence deux objectifs : écrire un algorithme en langage naturel puis l'implémenter sur une calculatrice Texas (TI 82) ou Casio (Graph35). C'est ce qu'elle indique dans l'entretien ayant eu lieu avant la séance : « je veux qu'ils écrivent les algorithmes en langage naturel [...], qu'ils les entrent dans la calculatrice ».

Trois types de tâches sont inscrits dans cette séquence : étudier des algorithmes déjà écrits, écrire des algorithmes, implémenter des algorithmes dans une calculatrice. Les tâches sont successivement :

- Etudier, en se situant dans un cadre géométrique, l'algorithme 1 permettant d'obtenir le milieu d'un segment à partir de deux points du plan.
- Etudier, en se situant dans un cadre numérique, l'algorithme 2 permettant d'obtenir $3x - 4$ à partir de x
- Ecrire puis implémenter dans la calculatrice un algorithme 3 permettant d'obtenir la moyenne pondérée de plusieurs notes ; l'enseignante choisit de faire écrire cet algorithme « *parce que l'on est à la fin du trimestre et les élèves sont motivés pour calculer leur moyenne générale* » (Entretien post-séance)
- Ecrire puis d'implémenter dans la calculatrice plusieurs algorithmes portant sur des calculs de prix HT, TTC, taux d'évolution.

Dans cette section, nous nous intéressons à la séance 2 portant sur l'écriture d'algorithmes relatifs à des calculs de prix. Il est demandé aux élèves d'écrire les algorithmes suivants :

Algorithme 4 : écrire un algorithme qui permet de calculer une remise de $t\%$ sur un prix p

Algorithme 5 : écrire un algorithme qui permet de calculer le prix final (après la remise), écrire ensuite un programme avec la calculatrice qui donne le prix final puis compléter le tableau

Algorithme 6 : écrire un algorithme qui permet de calculer le prix HT connaissant le prix TTC et le taux de TVA

Algorithme 7 : écrire un algorithme déterminant le taux de TVA (5,5% ou 7,5% ou 18,5%) connaissant le prix HT et le prix TTC.

Notons que les tâches proposées dans cette séance d'algorithmique correspondent à celles que l'on donnerait dans un cours de mathématiques. Le cours de mathématiques « classique » portant sur les pourcentages et l'utilisation du coefficient multiplicateur fait travailler principalement quatre types de tâches : le calcul de la remise (ou augmentation), le calcul du prix après réduction (ou augmentation), le calcul du prix initial, le taux d'évolution. Les tâches algorithmiques sont calquées sur des tâches mathématiques classiques.

L'écriture de l'algorithme 5 peut appeler plusieurs techniques :

Une technique τ_1 utilise le coefficient directeur, exprimant la relation entre les deux prix :

$$\text{prix}_{\text{après augmentation}} = \text{prix}_{\text{initial}} \times (1+t \%)$$

$$\text{prix}_{\text{après remise}} = \text{prix}_{\text{initial}} \times (1-t \%)$$

Une autre technique τ_2 que les élèves de collège connaissent bien fait appel au calcul intermédiaire de la remise ou de l'augmentation :

$$\text{prix}_{\text{après augmentation}} = \text{prix}_{\text{initial}} + \text{montant de l'augmentation}$$

$$\text{prix}_{\text{après remise}} = \text{prix}_{\text{initial}} - \text{montant de la remise}$$

L'appel à cette dernière technique est d'autant plus naturel pour les élèves qu'ils viennent d'écrire un premier algorithme qui permet d'obtenir le montant de

l'augmentation ou de la remise. Or, ainsi que nous le développerons plus loin dans l'analyse de l'action conjointe, l'enseignante écarte la deuxième technique au profit de la première technique qui vient juste d'être revue dans le cours de mathématiques.

La tâche que nous venons d'analyser est emblématique de la séquence :

- Toutes les tâches sont soumises à des organisations praxéologiques de différents ordres, chacune ayant un ancrage référentiel plus ou moins marqué dans une des deux disciplines (cf. schéma ci-après)

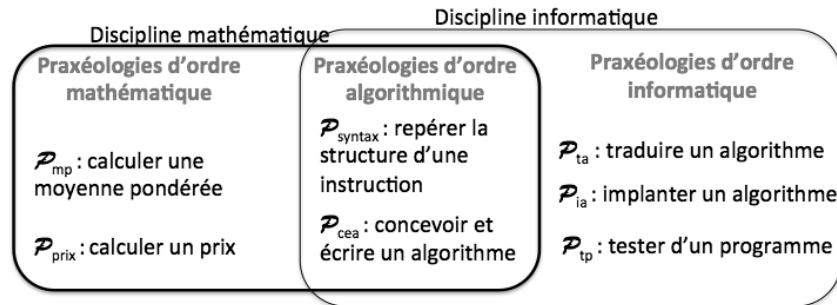


Figure 7. Praxéologies mises en œuvre dans la séquence 1

Nous observons que ces praxéologies sont dépendantes les unes des autres : la praxéologie \mathcal{P}_{cea} fait appel à des praxéologies « auxiliaires », \mathcal{P}_{mp} et \mathcal{P}_{prix} , nécessaires à l'écriture de la « phase de traitement » des algorithmes. L'implémentation des algorithmes nécessitera successivement la réalisation des praxéologies \mathcal{P}_{ta} et \mathcal{P}_{ia} .

Le schéma ci-après résume les dépendances entre ces praxéologies.

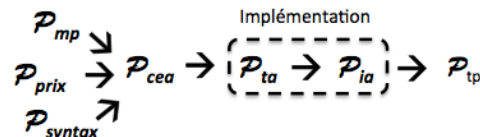


Figure 8. Dépendances entre les praxéologies de la première séquence

- La mise œuvre des algorithmes est fortement conditionnée par des praxéologies mathématiques : pour la praxéologie \mathcal{P}_{cea} , praxéologie algorithmique, la mise en œuvre de la technique mobilise des connaissances mathématiques qui sont des œuvres du milieu didactique. Parfois, ces œuvres peuvent être déjà connues de la classe (dans l'étude du premier algorithme, les élèves savent déterminer sans problème que le point I construit est le milieu du segment [AB]) mais parfois, elles peuvent nécessiter une étude préalable, de façon à rendre ces œuvres disponibles pour la classe (c'est le cas pour le calcul d'une moyenne pondérée ou le l'utilisation d'un coefficient multiplicateur pour le calcul de prix). L'écriture d'algorithmes devient alors, de par sa place dans l'organisation didactique, une tâche d'application ou de consolidation d'un savoir mathématiques sous un format particulier (celui d'un algorithme), un prétexte à consolider voire réinvestir une notion déjà étudiée dans le cours de mathématiques.

2.3.2. Deuxième séquence : des savoirs mathématiques pour écrire des algorithmes

L'enseignante appuie son projet d'enseignement sur des notions mathématiques stabilisées dans le milieu de l'élève. Elle pose d'emblée l'appareillage technologique qui sous-tend les activités algorithmiques proposées aux élèves en portant leur attention sur ce qui deviendra plus tard des éléments technologiques dans l'écriture des algorithmes :

Complétez les phrases suivantes :

P1 : Si $AB^2+AC^2=BC^2$ alors le triangle ABC est.....

P2 : SI Alors les vecteurs $\vec{u}(x,y)$ et $\vec{v}(x',y')$,dans un repère orthonormé (O,I,J) sont colinéaires

P3 : Si.....alors les points A,B et C sont alignés sinon ils ne sont pas alignés

P4 :Si mon âge est supérieur à ans alors je peux passer mon permis de conduire sinon je ne peux pas

P5

- Avec le forfait téléphonique « liberto », il n'y a pas d'abonnement fixe mais l'heure d'appel est facturée 12 euros
- Avec le forfait « freedom », le temps de communication est illimité mais le prix de l'abonnement est de 80 euros par mois.

Si je téléphone 3h par mois alors je dois choisir le forfait.....

Si je téléphone plus de.....h par mois alors je dois choisir le forfait.....

Figure 9. Appareillage praxéologique pour la séquence 2

Cet appareillage étudié soit dans les années précédentes (théorème de Pythagore), soit quelques semaines auparavant (colinéarité de vecteurs, fonction affine), permet à l'enseignante de se centrer sur l'objectif principal : « *comprendre le 'si...alors...sinon' et utiliser le langage machine 'if...then...else'* » (Entretien post-séance). Le contrat didactique est explicitement algorithmique et informatique : « *On a cinq algorithmes à faire. Sur votre feuille, vous faites deux colonnes : une colonne où vous écrirez le langage naturel. Le langage naturel, il doit être compris par n'importe quelle personne. N'importe qui doit comprendre votre algorithme en langage naturel. Par contre, le langage Texas ou le langage Casio, c'est pour vous. Mais surtout pour la machine, pour que le programme puisse tourner* ». La consigne donnée aux élèves est d'écrire 5 algorithmes correspondants aux cinq propositions étudiées précédemment (cf. Figure 9). Nous retrouvons les mêmes types de tâche qu'à la première séquence : étudier un algorithme déjà écrit, écrire des algorithmes et implémenter des algorithmes.

- Comme pour la première séquence, toutes les tâches sont soumises à des organisations praxéologiques d'ordre mathématique, algorithmique ou informatique (cf. schéma ci-dessous).

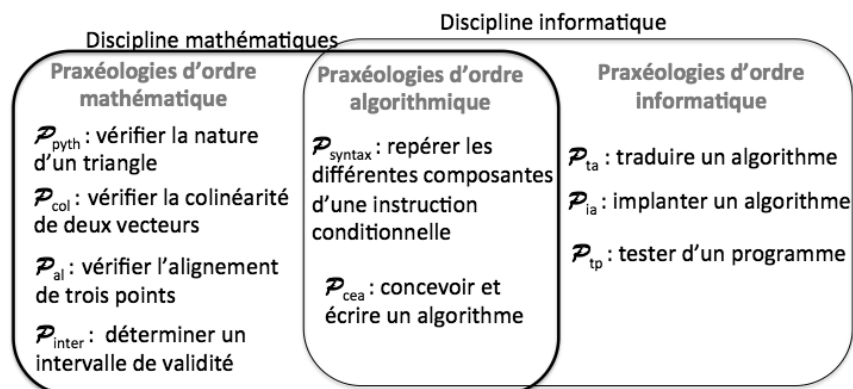


Figure 10. Praxéologies mises en œuvre dans la séquence 2

Par contre, les praxéologies mathématiques mobilisent cette fois-ci des techniques connues des élèves alors que la praxéologie \mathcal{P}_{syntax} , d'ordre algorithmique, mobilise des connaissances, non connues des élèves, nécessaires à l'écriture des algorithmes de la séquence. La conception et l'écriture d'algorithmes nécessite la réalisation des praxéologies mathématiques et de la praxéologie algorithmique \mathcal{P}_{syntax} . Nous schématisons ci-après la dépendance entre ces praxéologies.

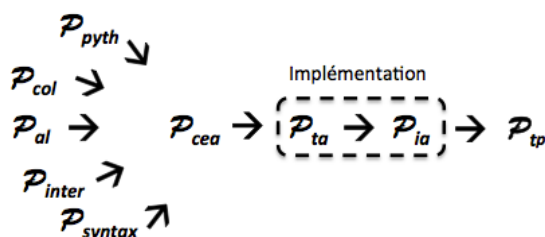


Figure 11. Dépendances entre les praxéologies de la seconde séquence.

- Le milieu didactique initial n'est composé que de matériaux considérés par l'enseignante comme connus des élèves : le théorème de Pythagore, la condition de colinéarité des vecteurs, les fonctions affines. Or, ainsi que nous le verrons dans la seconde partie, l'enseignante se heurte à une autre difficulté : les élèves restent dans un cadre géométrique, ne basculant pas dans un cadre algorithmique.

2.3.3. Pour conclure sur l'organisation praxéologique retenue par cette enseignante

Les organisation praxéologiques des deux séquences s'appuient essentiellement sur des praxéologies mathématiques. En première séquence, la construction d'algorithmes vient en fin de processus d'apprentissage de concepts mathématiques montrant qu'implicitement la visée était principalement de consolider des savoirs mathématiques. Dans la seconde séquence, le projet didactique est résolument algorithmique, mais les tâches proposées reposent sur des savoirs emblématiques des mathématiques nécessitant des passages dans différents cadres conceptuels pour pouvoir manipuler des concepts algorithmiques.

Comment la co-construction des savoirs algorithmiques a-t-elle alors lieu au sein de la classe ?

2.4. Analyse de la co-construction de ces savoirs en classe

Pour analyser la co-construction d'un savoir algorithmique, nous avons utilisé les outils d'analyse de l'action conjointe en didactique (ACD) tels que les avons brièvement décrits dans la section « cadre théorique ».

2.4.1. Première séquence : variable et affectation, deux concepts problématiques

Une désignation floue du concept de variable algorithmique

Rappelons que selon Knuth (1968), l'affectation, en lien avec la notion de variable, est une des opérations conceptuellement la plus difficile à comprendre. Nous avons donc observé l'enseignante aux prises avec cette difficulté, difficulté se traduisant d'abord par l'utilisation d'un vocabulaire imprécis : mettre, lettre, stockage, mémoire, flèche, Sto, égal, variable, affecter. Ce flou, se traduisant par un vocabulaire approximatif, entraîne un ensemble de questions de la part des élèves, questions qui révèlent une incompréhension quant à la nature même de cet objet :

E1 : « et le résultat de ça, je vais mettre un nombre, un chiffre... Un nombre ... Je vais mettre un nom ? » (E écrit par le geste un « X ») (séance 2)

Une séance plus tard, un autre élève, lors du codage en langage machine :

E2 : « peu importe la lettre ? »

Extrait 1 : interrogation sur la notion de variable

Cette incompréhension fait écho à une autre, à propos de la notion d'affectation : celle-ci est introduite par l'emploi des verbes « mettre » (en séance1) et « stocker » (en séance 2). Dans la séance 3, L'enseignante introduit le terme institutionnel « affecter » qui est ensuite utilisé de façon quasi-systématique par les élèves. Par la suite, dans la traduction des algorithmes en langage machine (Texas ou Casio), elle est amenée à utiliser indifféremment les deux termes : « flèche » et « STO » (séances 2 et 3).

L'affectation, concept problématique

Une analyse plus fine des interactions entre les élèves et l'enseignante à propos de la question de l'affectation, montre comment l'enseignante tente de faire accepter aux élèves l'utilisation d'un résultat de cours, mais qui du coup ne lui permet pas d'étudier pleinement le concept d'affectation. Tout d'abord, nous remarquons que les premiers algorithmes étudiés ne mettent en jeu qu'une seule variable, le traitement s'écrivant en une seule instruction et l'affectation du résultat précédant son affichage. Dans cette configuration, pour bon nombre d'élèves la fonction STO d'affectation est réservée au résultat du calcul, juste avant l'affichage avec la fonction DISP. On peut penser que les notions de mémoire et d'affectation ne sont pas comprises et que l'instruction d'affectation est associée à l'instruction d'affichage. Ceci n'est pas sans conséquences sur la chronogenèse du savoir relatif au concept d'affectation. Nous l'illustrons au travers de trois épisodes remarquables, issus de la première séquence.

Lors de l'écriture de l'algorithme 5, nous assistons à un jeu de persuasion à propos du calcul d'un prix réduit, entre des élèves ancrés dans des connaissances

antérieures (passage par le calcul de la remise) et l'enseignante qui veut leur faire utiliser une formule factorisée, formule établie lors d'un cours sur les fonctions affines.

Premier épisode : intrusion de la calculatrice dans l'écriture de l'algorithme

Les élèves ont écrit et implémenté avec succès un premier algorithme (calcul du montant de la remise à partir du prix et du taux de réduction). Pour écrire le second algorithme qui permet d'obtenir le prix remisé, plusieurs élèves souhaitent écrire les deux opérations qu'ils utilisent habituellement : calculer la remise, puis effectuer une soustraction pour obtenir le résultat escompté.

Ayant trouvé dans un des menus de la calculatrice le symbole « = », l'élève E3 pense pouvoir l'utiliser pour les calculs intermédiaires, comme il le ferait en mathématiques et stocker le résultat final dans une variable :

Tdp ⁵ 1	E3	comment on fait le « = » ?
Tdp 2	P	comment le ...
Tdp 3	E3	le « = »
Tdp 4	P	pourquoi tu as besoin du « = » ?
Tdp 5	E3	ben pour faire égal !
Tdp 6	P	pourquoi f... Pour où ? ...oui ?
Tdp 7	E3	Prompt p,
Tdp 8	P	Oui
Tdp 9	E3	Prompt t
Tdp 10	P	Oui
Tdp 11	E3	je vais faire $p*t/100$
Tdp 12	P	mmm
Tdp 13	E3	et le résultat de ça, je vais mettre un nombre, un chiffre... Un nombre ... Je vais mettre un nom ? (E12 écrit un « X » gestuellement)
Tdp 14	P	Oui
Tdp 15	E3	par exemple x, je ferai x moins p
Tdp 16	P	oui. Hé bé ... alors ?
Tdp 17	E3	comment je ferai le « égal » pour trouver ?
Tdp 18	P	on va toujours utiliser la flèche, en fait.
Tdp 19	E3	ah je fais sto...
Tdp 20	P	tu ne tu ne mets pas égal, voilà. Tu le stockes dans une mémoire, hein, et on va se resservir de cette lettre pour la mettre ensuite dans une autre mémoire
Tdp 21	E3	Ok
Tdp 22	P	ou la même. Comme tu veux.
Tdp 23	E3	(parlant tout seul) prompt p, prompt t ...
Tdp 24	P	(s'éloignant) ben, sinon demain, on verra qu'on peut... On est pas obligé de faire comme tu fais
Tdp 25	E3	ah ouai ? hé madame, c'est quoi là ? Il y avait « plus », « moins », « différent » ...
Tdp 26	P	oui, c'est dans test. Mais on verra plus tard...


Extrait 2 : épisode 1 (séquence 1 - séance 2)

⁵ Tdp : tour de parole

Cet épisode montre l'intrusion de la calculatrice dans l'activité d'écriture. Si E3 écrit bien le premier algorithme « à la main », il écrit le second directement dans la calculatrice. L'enseignante est alors happée par les questions de E3 (Tdp 7, 9 et 19) qui l'entraînent sur un terrain informatique : elle répond dans le langage de la calculatrice et non dans un langage algorithme. A la fin de cet épisode, l'enseignante s'éloigne sans apporter de réponse de type algorithmique quant à l'écriture d'une ou deux instructions et/ou l'utilisation d'une ou deux variables dans l'algorithme. En position topogénétique basse, elle ne rebondit pas sur les remarques de E3 pour agir sur l'avancée du savoir algorithmique.

Deuxième épisode :

En réalité, l'enseignante reste concentrée sur un objectif : écrire le traitement en une seule opération, opération qu'elle cherchera à conforter plus tard dans une phase de régulation :

Tdp 1	E12	ah moi je sais pas comment on fait le « égal », je n'ai pas compris
Tdp 2	P	Le « égal », je te dis, vous utilisez le stockage là.  (Le professeur montre le tableau ci-contre :)
Tdp 3	E12	ouais mais...[E12 se prend la tête dans les mains]
Tdp 4	P	vous mettez dans... [Le professeur écrit au tableau]. Si vous avez 20 % de remise, comment vous avez fait pour calculer ? Vous avez fait 32 - 20 % de 32. E12, toi tu veux faire en deux opérations. Si tu veux faire en une seule opération . Tu fais 32 - 20 % de 32. On est d'accord ?
		[E12 hoche la tête.]
		On peut mettre 32 en ...? En facteur ! Ça c'est ce qu'on va voir jeudi. Hein ? Quand on fait une remise de 20 %, on peut multiplier directement le prix initial 32 par $1 - t\%$. J'ai vu qu'il y en a qui avaient mis la formule. [...] Bon, E13, tu n'y as pas pensé ? À écrire $32 \left(1 - \frac{20}{100}\right)$, c'est-à-dire $32 \times 0,8$. Vous avez directement le prix remisé, comme ça. Et vous avez une seule opération si vous ne voulez pas faire l'étape « je calcule la remise et j'enlève la remise »... C'est ce qu'on va voir demain, pour terminer le cours. Ou plutôt jeudi... Pensez bien que plutôt que le faire en deux opérations , vous pouvez utiliser ça. (Le professeur montre le tableau). Diminuer, c'est faire, c'est multiplier par $1 - t\%$. Diminuer de $t\%$. Et donc on revient avec une seule opération .

Extrait 3 : épisode 2 (séquence 1 - séance 2)

A la fin de cette deuxième séance, nous constatons que plusieurs élèves ne cherchent pas à effectuer le traitement en une seule opération. Pour ces élèves, la confusion entre le signe « = » repéré dans un des menus de la calculatrice et l'opération d'affectation demeure. L'enseignante, quant à elle, cherche à convaincre de l'intérêt de l'utilisation du coefficient multiplicateur $(1 - t\%)$. Pour cela, elle rappelle la technologie justifiant cette technique, la factorisation, et use

d'arguments d'économie (en gras dans l'extrait 3). L'expression «*Bon, E13, tu n'y as pas pensé ?*» (tdp 4) suggère qu'en se plaçant en surplomb, elle espère remporter l'adhésion des élèves. La troisième séance permettra aux élèves d'affirmer leur point de vue.

Troisième épisode :

Ainsi que le montre l'échange ci-dessous, l'enseignante finit par accepter la proposition des élèves d'effectuer le traitement en deux étapes, et donc d'utiliser deux variables. Elle avance cette fois-ci un argument algorithmique : « ça va faire une ligne de calcul de plus ». On peut supposer qu'elle ne souhaite pas faire implémenter deux lignes de calculs, ce qui pourrait l'amener à discuter de la signification du « = » en informatique, mais du coup ne lui permet pas de saisir l'opportunité d'approfondir les concepts d'affectation et de variables. Notons toutefois qu'elle mentionne pour la première fois, sans s'y attarder, l'écrasement d'une variable.

Tdp1	E5	pourquoi on ne fait pas P - R ?
Tdp2	P	alors, si on fait P - R... On peut faire un algorithme où on fait d'abord calculer la remise, et puis après on va faire P - R. Ça va rajouter une ligne de calcul. Dans ton algorithme, ce serait un algorithme où on ferait calculer R, c'est-à-dire on ferait calculer $P \times \frac{t}{100}$, et après calculer P - R. Et après, tu le mettrais dans une autre lettre ou encore dans P et après voilà. Ça va marcher aussi. Si vous l'avez programmé avec la machine, tu as dû voir que ça marchait.

Extrait 4 : épisode 3 (séquence 1 - séance 3)

L'analyse de cette séquence nous a conduit à deux constats :

- Elèves et enseignant ne poursuivent pas la même démarche : les élèves cherchent à écrire un algorithme en réinvestissant des connaissances anciennes alors que l'enseignante cherche, tout en faisant travailler l'écriture d'un algorithme, à asseoir un savoir mathématique en cours d'apprentissage, vu précédemment en « papier-crayon » dans un cours traditionnel. Néanmoins, l'enseignante finit par accepter la démarche algorithmique proposée par les élèves. Dans cette séquence, ce sont principalement les élèves qui agissent sur la chronogenèse du savoir algorithmique.
- L'aspect informatique est prédominant : le langage utilisé, et par les élèves et par l'enseignante, appartient plus au domaine informatique (« stocker », « prompt ») qu'au domaine algorithmique. Cet aspect est majoré par le fait que élèves travaillent sur la calculatrice sans passer par l'activité d'écriture « papier-crayon ».

2.4.2 Deuxième séquence : difficultés dans les changements de cadres conceptuels

Nous présentons ici deux épisodes significatifs de la difficulté des élèves à passer d'un cadre géométrique à un cadre numérique pour ensuite travailler dans un cadre algorithmique. L'objectif de l'enseignante dans cette séquence est de faire travailler l'instruction conditionnelle « Si ... alors ... sinon ».

Premier épisode :

La tâche de l'élève est d'écrire un algorithme vérifiant la colinéarité de deux vecteurs. Les deux extraits ci-dessous révèlent que E12, E22 et E23 n'ont pas pris encore conscience de la nature des objets mathématiques en jeu dans l'écriture des algorithmes. Dans l'extrait ci-dessous, E12 inscrit en entrée les sommets du triangle, faisant une confusion entre A (tdp 3 et tdp 5), sommet du triangle et la mesure a d'un côté du triangle. C'est l'enseignante qui oriente les élèves sur des objets numériques, ici les coordonnées du point A (tdp 8).

Tdp 1	E 12	bein, quand on écrit, il faut marquer quoi...
Tdp 2	P	Voilà. On va rentrer les différentes longueurs des côtés du triangle et c'est l'algorithme du programme qui dira si le triangle est rectangle ou pas. Alors en entrée, qu'est ce qu'on met ? Qu'est ce qu'il va falloir écrire en entrée ?
Tdp 3	E 12	les points
Tdp 4	P	Les points, ça veut dire quoi les points ?
Tdp 5	E 12	Ben heu ... bein prompt a, déjà
Tdp 6	P	Prompt a, ça veut dire quoi pour toi ?
Tdp 7	E 12	Le point ...
Tdp 8	P	Tu voudrais rentrer les coordonnées du point A...

Extrait 5 : recherche des entrées de l'algorithme 1 (séquence 2 – séance 1)

Dans l'extrait 6, les deux élèves E22 et E23 prennent conscience du changement de cadres qu'elles doivent opérer (tdp 9). L'enseignante, usant de la technique du trilogue (Schubauer-Leoni, 1997), indique aux élèves les noms des variables, s'appuyant sur les contraintes de la calculatrice, mais éludant du même coup le travail sur la notion de variable en mathématique et en algorithmique.

Deux élèves filles au fond de la classe appellent le professeur		
Tdp 1	E23	ici c'est des nombres qu'il faut demander ? Alors que c'est des vecteurs ...
Tdp 2	P	oui. Alors il faut demander les coordonnées. Que tu écrives demander les coordonnées du vecteur U. Et entre parenthèses, c'est a et b par exemple. Tu vois c'est une bonne remarque ça.
Tdp 3	E22	mais on ne marque pas x, y ?
Tdp 4	P	il faut que tu marques... Il faut en demander quatre nombres. Il y a quatre entrées dans cet algorithme.
Tdp 5	E23	x et x'
Tdp 6	P	[élevant la voix] Si tu veux. Sur ta machine, tu n'as pas x' y'. C'est pour ça que je vous conseillais d'écrire a b c d
Tdp 7	E23	Oui mai si ... parce que là c'est c'est ... du coup, je ne marque pas entre parenthèses vecteur U et vecteur V
Tdp 8	P	Vaut mieux faire le contraire. Vaut mieux que tu marques demander les coordonnées du vecteur U, entre parenthèses tu marques a b. Et les coordonnées du vecteur V, entre parenthèses c d.
Tdp 9	E23	Le vecteur v, on ne peut pas le faire sur la calculatrice
Tdp 10	P	Non. Il faudra que tu demandes un nombre. Tu demanderas a et b. C'est comme si c'était xA et xB ou heu ...xU et xV

Extrait 6 : recherche des entrées de l'algorithme 2 (séquence 2 – séance 2)

L'enseignante pare les difficultés des élèves en prenant en charge l'avancée du savoir lors de corrections collectives pour chacun des algorithmes ce qui amènera

peu à peu les élèves à écrire seuls le dernier algorithme. La trace écrite (figure ci-après) montre la volonté de l'enseignante de dissocier ce qui relève de l'algorithmique ou de l'informatique.

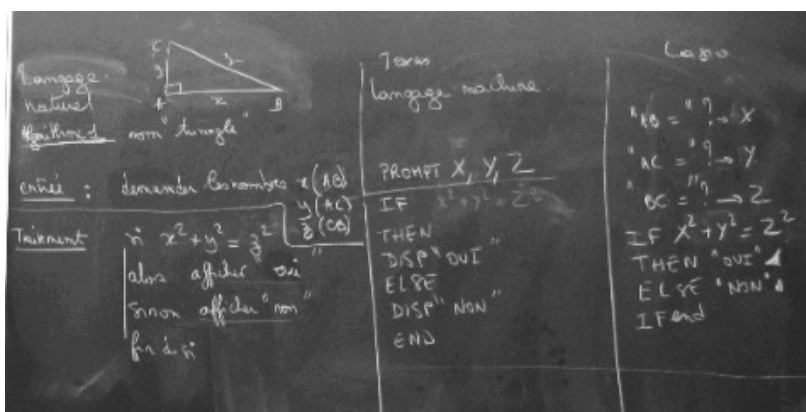


Figure 12 : correction de l'algorithme 1 (séquence 2 – séance1)

L'analyse de cette séquence nous a conduit à deux constats :

- L'enseignante, en constituant le milieu didactique d'objets mathématiques connus de longue date des élèves, fait le choix de centrer la séquence sur des apprentissages algorithmiques. Pour autant, les élèves ne réussissent pas à entrer dans la démarche algorithmique car ils n'ont pas conscience de la nature des objets entrant en jeu dans un algorithme.
- L'enseignante prend en charge l'avancée du savoir en se situant en surplomb : elle oriente les élèves en leur indiquant les variables en jeu abaissant ainsi l'enjeu didactique en algorithmique : le travail sur la notion de variable en mathématique et en informatique est évité.

3. Discussion : de la difficulté d'enseigner un savoir sous couvert de deux disciplines de référence

Si les textes officiels préconisent l'étude de l'algorithmique dans tous les domaines d'étude en mathématiques, ils ne donnent pas d'indications sur la ou les manières de l'introduire au sein du cours. L'observation d'un enseignement d'algorithmique dans une classe 'ordinaire' de seconde montre les difficultés d'enseigner un savoir se référant à deux disciplines. Les praxéologies algorithmiques mises à l'étude par cette enseignante au travers des tâches qu'elle propose mettent en évidence deux orientations. L'enseignante oscille entre deux positions : consolider des savoirs mathématiques en écrivant des algorithmes et/ou écrire des algorithmes pour construire des programmes utiles dans et hors le cours de mathématiques. Dans les deux cas, elle respecte bien les injonctions du programmes qui incitent à placer l'algorithmique « dans tous les champs des mathématiques » (MEN, 2009a).

(i) Toutes les tâches d'écritures d'algorithmes mettent en œuvre des concepts mathématiques et toutes ont donné lieu à une implémentation dans un langage informatique, celui de la calculatrice des élèves. Les activités d'écritures d'algorithmes mettent en œuvre des concepts étudiés peu de temps de temps auparavant dans le cours de mathématiques : « *En fait, j'aime bien les mettre en fin*

de cours [les algorithmes] » (entretien ante ; séquence 2-séance 2). Nous interprétons ce propos comme une manière de réinvestir, voire consolider, des apprentissages récents. Ce disant, l'enseignante dévoile que le projet didactique est, en filigrane, d'ordre mathématique. Cette approche est confortée par les préconisations institutionnelles : « *L'algorithmique a une place naturelle dans tous les champs mathématiques et les problèmes posés doivent être en relation avec les autres parties du programmes (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante* » (MEN, 2009a). On peut alors comprendre que la place dévolue à l'algorithmique soit comprise comme une « mise en textes algorithmiques » de concepts mathématiques étudiés dans le cours de mathématiques, ce qui nous autorise à poser la question de l'enjeu d'étude de l'algorithmique : l'enjeu est-il d'aborder des concepts mathématiques sous un autre angle, ou d'étudier des concepts algorithmiques ? Reprenant la formulation de Modeste (2012, p. 239), l'algorithme est-il un « *outil [didactique] d'enseignement ?* »

(ii) Par ailleurs, l'enseignante a une vision informatique de l'algorithmique : pour elle, il est important que les élèves « *fassent des programmes qui leur servent* » (entretien ante ; séquence 1- séance 1). Cette manière d'aborder l'algorithmique en favorisant son versant informatique est elle aussi confortée par les documents officiels qui présentent presque tous les algorithmes codés dans un langage informatique. La volonté de produire un programme 'opérationnel' a pour conséquence de donner le primat à la calculatrice ou au logiciel aux dépens d'activités purement algorithmiques telles que la validation ou le traçage de la variable, activités qui permettraient alors de se centrer sur l'étude de quelques concepts algorithmiques : variable, affectation par exemple. Notons aussi que dans cette approche, la calculatrice devient alors un outil de validation, ce qui est en contradiction avec ce qui se fait habituellement dans les autres parties du programme mathématique où la calculatrice, tout comme le tableur ou la figure géométrique, sont considérés comme des éléments initiateurs d'une conjecture qui sera ensuite prouvée avec des outils mathématiques. Ceci nous autorise à poser à nouveau la question de l'enjeu d'étude : est-il programmatique ou algorithmique ?

(iii) La deuxième séquence montre les difficultés de changements de cadres conceptuels que les élèves ont à effectuer lors de l'écriture des algorithmes. Ces difficultés peuvent s'expliquer par une absence de réflexion sur la nature des objets travaillés sur le plan mathématique, algorithmique ou informatique. Ainsi, comme l'écrit Briant dans sa thèse, « *l'élève, devant la tâche de concevoir un algorithme et un programme dans un environnement informatique, se voit dans une situation de décomposer chacune de ses actions en actions élémentaires. Cette décomposition passe par une réflexion nécessaire sur les objets mathématiques en jeu dans la tâche à effectuer et oblige les élèves à revenir sur leurs conceptions de ces objets.* » (Briant, 2013, p. 553). Les résultats produits dans cette étude de cas contribuent à la suite de cet auteur à questionner d'une autre manière l'enjeu d'étude de l'algorithmique dans les programmes actuels : en quoi l'approche algorithmique des concepts peut-elle alimenter une réflexion sur les objets mathématiques, et réciproquement sur les objets informatiques ?

4. Conclusion

Il ressort de cette analyse que la vision de Knuth (1973) quant à l'hybridité de l'algorithmique est bien perceptible tant dans les textes institutionnels que dans les pratiques observées. Le concept de praxéologie emprunté à la TAD et les descripteurs de l'action conjointe (TACD) ont permis de faire émerger les savoirs mis à l'étude en algorithmique dans une classe ordinaire en montrant combien les préconisations du programme rendent difficiles autant les actions du professeur que celles des élèves. Dans notre analyse de cas, nous montrons qu'enseignant et élèves travaillent séparément les aspects de l'algorithmique, ne tenant pas ainsi compte de son hybridité : l'enseignante le met à l'étude sous couvert des mathématiques *puis* sous couvert de l'informatique ; les élèves quant eux, dans l'action conjointe, tendent à valoriser le versant informatique dans ses dimensions techniques. Nous rejoignons ainsi des résultats de recherches en didactique montrant que les objets d'enseignement déclarés à l'interface de deux disciplines, sont difficilement transposables dans leur double référence (Devos-Prieur & Grandaty, 2011 ; Leutenegger, 2008 ; Schubauer, Leutenegger & Forget, 2007). Dans cette étude, et comme l'ont montré ces auteurs, nous avons mis en évidence que l'objet algorithmique, de nature hybride (Knuth, 1968) avance au fil du temps didactique soit sous couvert d'une discipline (les mathématiques), soit sous couvert d'une autre (l'informatique).

Pour autant, dans la perspective de résolution de cette difficulté didactique, les récents travaux de Modeste (2012) et de Briant (2013) posent d'une autre manière la question de l'intégration du domaine de l'algorithmique dans l'enseignement des mathématiques au lycée. Modeste montre dans sa thèse que l'algorithme tel qu'il est présenté dans les textes officiels est « *une transposition partielle du concept principalement orientée vers la programmation et l'usage de l'algorithme comme un outil* » et propose des situations didactiques « *mettant en jeu des activités de preuve ou de recherche dans un contexte de résolution de problèmes* ». Briant, à propos de l'enseignement de l'algèbre, soutient que algorithmique et programmation obligent « *à revenir sur les conceptions des objets* » et ouvre « *un nouveau registre de représentation pour les objets de l'algèbre* ». Ces propositions donneraient alors pleinement vie à l'enseignement des concepts algorithmiques dans l'enseignement des mathématiques sous réserve d'un « *équipement praxéologique de la profession, équipement comportant à la fois des savoirs savants et des savoirs pour enseigner* » (Briant, 2013, p. 545) qui s'avère indispensable.

Remerciements

Je tiens à remercier Chantal Amade-Escot, Gisèle Cirade ainsi que les relecteurs de la revue pour leur aide précieuse. Sans leur soutien et leurs remarques avisées, cet article n'aurait peut-être pas vu le jour ...

Bibliographie

ARTIGUE M. (2002), Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work, *International Journal of Computers for Mathematical Learning*, vol 7, n°3, 245-274.

- APMEP (2014). Réforme du lycée : un premier bilan pour la classe de seconde.
<http://www.apmep.fr/Reforme-du-lycee-un-premier-bilan>
- ASTI-ITIC (2007). Pour un enseignement de l'Informatique et des Technologies de l'Information et de la Communication au lycée.
<http://www.epi.asso.fr/revue/editic/asti-itic-lycee-prog.htm>
- ATHIAS F. (2014), « *La géométrie dynamique comme moyen de changement curriculaire* » Thèse de doctorat. Université Aix-Marseille.
- BRIANT N. (2013), *Étude didactique de la reprise de l'algèbre par l'introduction de l'algorithmique au niveau de la classe de seconde du lycée français*. Thèse de doctorat. Université Montpellier 2.
- BRIÈRE-GUENOUN F. & AMADE-ESCOT C. (2010), Analyse in situ des savoirs mobilisés par un professeur d'éducation physique et sportive dans l'interaction didactique. *Revue Suisse des sciences de l'éducation*, **32(2)**, 595-614.
- BOSCH M. & GASCON J. (2002), Organiser l'étude 2. Théories et empiries dans Dorier, J.-L. et al. (Eds). Actes de la 11e école d'été de didactique des mathématiques, Grenoble : La Pensée Sauvage, 21-30 Août 2001, 23-40.
- CHEVALLARD Y. (1999), L'analyse des pratiques enseignantes en théorie anthropologique du didactique. *Recherches en Didactique des Mathématiques*, **19.2**, 221-265.
- CHEVALLARD Y. (1985), Le passage de l'arithmétique à l'algèbre dans l'enseignement des mathématiques au collège - Première partie. L'évolution de la transposition didactique. *Petit x*, **5**, 51-94.
- COUDERETTE M. (2013), *Enseignement de l'algorithmique en cours de mathématiques en classe de seconde*. Communication orale au 3^{ème} colloque international de l'ARCD : "Savoirs, compétences, Approches comparatives de l'organisation des contenus, et des formes, de l'étude ; variations et constantes disciplinaires, institutionnelles, culturelles", Marseille.
- DEVOS-PRIEUR O. & GRANDATY M (2011), *Les retombées de la formation continue : articulation des contenus enseignés en EPS et Maîtrise de la Langue chez un maître formateur et un Professeur d'école*. Communication orale au 2^{ème} colloque international de l'ARCD : "Les contenus disciplinaires", Lille. In CD rom des actes.
- DOWEK G. (2008), Les rencontres de l'Ormes 2008. Séminaire de l'ASTI
<http://www.epi.asso.fr/revue/docu/d0805a.htm>
- GASCON J. (1994), Un nouveau modèle de l'algèbre élémentaire comme alternative à "l'arithmétique généralisée", *Petit x* **37**, 43-63. Grenoble: Université Joseph Fourier.
- HASPEKIAN M. & NIJIMBÉRE C. (2012), Les enseignants face à l'entrée de l'algorithmique dans l'enseignement des mathématiques au lycée, dans M. Gandit & B. Grugeon (dir.) Actes du 19e colloques de la CORFEM. IUFM Franche-Comté, juin 2012, 265-285.

- HOARE C. (1969), An Axiomatic Basis for Computer Programming. *Communications of the ACM*, 12, 576-583.
- KAHANE J.-P. (2002), Enseignement des sciences mathématiques : Commission de réflexion sur l'enseignement des mathématiques : Rapport au ministre de l'éducation nationale (O. Jacob, Ed.). Paris : CNDP. Disponible sur <http://smf4.emath.fr/Enseignement/CommissionKahane/>
- KIERAN C. (1981), Concepts associated with the equality symbol, *Educational Studies in Mathematics* **12(3)**, 317-326.
- KNUTH E. et al. (2006), Does Understanding the Equal Sign Matter? Evidence from Solving Equations. *Journal for Research in Mathematics Education*, **37(4)**, 297-312. Retrieved from <http://www.jstor.org/stable/30034852>
- KNUTH D. E. (1968), *The Art of Computer Programming*, Vol 1 : Fundamental Algorithms (2e éd.). Addison-Wesley.
- LEUTENEGGER F. (2009), *Le temps d'instruire. Approche clinique et expérimentale du didactique ordinaire en mathématique*. Berne, Peter Lang.
- LEUTENEGGER F. (2008), L'entrée dans une code écrit à l'école enfantine et l'articulation entre le collectif et l'individuel : comparaison de deux études de cas. *Éducation & Didactique*, **2(2)**, 7-42.
- Ministère de l'Éducation Nationale (2009a). Programme de mathématiques de la classe de seconde générale et technologique. Bulletin officiel n° 30 du 23 juillet 2009.
http://media.education.gouv.fr/file/30/52/3/programme_mathematiques_seconde_65523.pdf
- Ministère de l'Éducation Nationale (2009b). Ressources pour la classe de seconde, Algorithmique ; EduSCOL, DGESCO.
<http://eduscol.education.fr/cid45766/ressources-pour-faire-la-classe-au-college-et-au-lycee.html>
- Ministère de l'Éducation Nationale (2014). Ministère de l'Éducation nationale. Commission de suivi de la mise en œuvre des programmes de mathématiques. DGESCO.
http://cache.media.eduscol.education.fr/file/Mathematiques/01/6/CSM-projet-rapport2013_293016.pdf
- MODESTE S. (2012), *Enseigner l'algorithme pour quoi ? Quelles nouvelles questions pour les mathématiques ? Quels apports pour l'apprentissage de la preuve ?* Thèse de doctorat, Université Joseph Fourier – Grenoble.
- MODESTE, S., GRAVIER, S., OUVRIER-BUFFET, C. (2010). Algorithmique et apprentissage de la preuve. *Repères-Irem*, **79**.
- NGUYEN C-T. (2005), *Étude didactique de l'introduction d'éléments d'algorithmique et de programmation dans l'enseignement mathématique secondaire à l'aide de la calculatrice*. Thèse de doctorat, Université Joseph Fourier – Grenoble.

SALONE J.J. (2015) « *Les Références Praxéologiques dans les Systèmes Didactiques* » Thèse de doctorat, Université Aix-Marseille.

SCHUBAUER-LEONI M.-L. (1997), « Interactions didactiques et interactions sociales : quels phénomènes et quelles constructions conceptuelles ? ». *Skholê*, 7, 103-134.

SCHUBAUER-LEONI, M.L. ; LEUTENEGGER, F. ; FORGET, A. (2007). L'accès aux pratiques de fabrication de traces scripturales convenues aux commencements de la forme scolaire : interrogations théoriques et épistémologiques. *Éducation & didactique*, 1(2), 7-35

SENSEVY, G. (2007). Des catégories pour décrire et comprendre l'action didactique. In G. Sensevy & A. Mercier (Éds.) (2007), *Agir ensemble. Éléments de théorisation de l'action conjointe du professeur et des élèves*, 13-49. Rennes : Presses Universitaires de Rennes.

MICHELE COUDERETTE

ESPE Midi Pyrénées

michele.couderette@univ-tlse2.fr