

LA PENSÉE ALGORITHMIQUE : APPORTS D'UN POINT DE VUE EXTERIEUR AUX MATHEMATIQUES

Simon MODESTE*

Résumé – Nous abordons ici la question de la *pensée algorithmique*. Nous adoptons deux points de vue. Le premier consiste à regarder la pensée algorithmique comme *une* pensée des mathématiques, l'algorithme étant depuis des siècles, outil et objet des mathématiques. Le second consiste à la voir comme *la* pensée de la science algorithmique (ou informatique). L'algorithme est l'objet central de l'informatique, et c'est en informatique que son étude a été largement développée, avec un impact fort sur les mathématiques. Nous montrons que cette deuxième approche enrichit la première, dans l'objectif d'une caractérisation de la pensée algorithmique, mais aussi pour des perspectives d'enseignement.

Mots-clefs : algorithme, pensée algorithmique, pensée mathématique, activité, enseignement

Abstract – We deal here with the question of *algorithmic thinking*. We study it from two points of view. First, algorithmic thinking can be seen as *a particular* mathematical thinking, since algorithm has been a tool and an object of mathematics for centuries. Second, algorithmic thinking can be seen as *the* thinking of algorithmic science (or computer science). Algorithm is the most important concept in computer science, and computer science made it deeply evolve with a strong impact on mathematics. We show that the second point of view can enrich the first one, both for epistemological perspectives on algorithmic thinking and for educational issues.

Keywords: algorithm, algorithmic thinking, mathematical thinking, activity, teaching

Depuis plusieurs années, l'outil informatique et les questions qui lui sont liées font leur apparition dans l'enseignement secondaire. Le cours de mathématiques n'échappe pas à cela. Cependant, il est indéniable qu'un lien particulier unit mathématique et informatique et qu'en mathématique, la science informatique apporte bien plus que des outils. L'algorithme, objet central de l'informatique, mais aussi objet des mathématiques depuis des siècles, prend maintenant une nouvelle place dans les mathématiques et nécessairement dans leur enseignement. Les mathématiques s'en trouvent interrogées et leur enseignement aussi. Par exemple, le *National Council of Teachers of Mathematics* a consacré en 1998, son « Yearbook » à l'enseignement et l'apprentissage des algorithmes dans la classe de mathématiques (NCTM 1998). Depuis 2009, en France, une part d'algorithmique a été introduite dans les programmes du lycée.

Il n'est pas impossible que l'algorithmique accède dans les années à venir au même statut que l'algèbre, la géométrie, l'analyse ou les probabilités et statistiques dans les programmes et manuels du secondaire.

Un tel changement demande un approfondissement des liens qu'entretiennent mathématiques et algorithmique. Quelles relations et différences existe-t-il entre pensée mathématique et pensée algorithmique ? En quoi la pensée algorithmique diffère-t-elle des autres pensées mathématiques ?

I. CADRE DU QUESTIONNEMENT

1. *Algorithme et algorithmique*

Pour commencer, il nous semble indispensable de décrire ce qu'est l'algorithmique, dans son sens le plus répandu. Pour cela il nous faut préciser ce que constitue le concept d'algorithme,

* Institut Fourier, Équipe combinatoire et didactique, Grenoble – France – simon.modeste@ujf-grenoble.fr

au sens mathématique (et informatique) du terme, afin de lever toute ambiguïté, et de le distinguer de termes plus vagues (auquel il est souvent associé, voire substitué¹) tels que procédé, méthode, technique, heuristique ou même programme. C'est toujours dans le sens de cette définition que nous entendrons le terme algorithme, notamment nous éviterons de parler d'algorithmes concernant les techniques ou méthodes (créées ou apprises) mises en œuvre pour résoudre un problème.

Définition² : Un algorithme est une procédure systématique permettant de résoudre une classe de problèmes. À partir d'une entrée (représentant une instance du problème), un algorithme suit un ensemble déterminé de règles et, en un nombre fini d'étapes, produit une sortie (représentant une réponse à l'instance donnée).

L'algorithmique est donc l'étude des algorithmes, de leur conception et de leurs propriétés. Knuth (1985) quant à lui, perçoit l'algorithmique comme étant l'informatique toute entière.

For many years I have been convinced that computer science is primarily the study of algorithms. My colleagues don't all agree with me, but it turns out that the source of our disagreement is simply that my definition of algorithms is much broader than theirs: I tend to think of algorithms as encompassing the whole range of concepts dealing with well-defined processes, including the structure of data that is being acted upon as well as the structure of the sequence of operations being performed [...] In the U.S.A., the sorts of things my colleagues and I do is called Computer Science, emphasizing the fact that algorithms are performed by machines. But if I lived in Germany or emphasizing the stuff that France, the field I work in would be called Informatik or Informatique, algorithms work on more than the processes themselves. In the Soviet Union, the same field is now known as either Kibernetika (Cybernetics), emphasizing the control of a process, or Prikladnaia Matematika (Applied Mathematics), emphasizing the utility of the subject and its ties to mathematics in general. (Knuth 1985, p. 170)

Selon lui, l'algorithmique doit être vue comme l'étude de ce qui est automatisable. Une telle vision de l'algorithmique s'étend au-delà des mathématiques.

Du point de vue de l'enseignement, l'idée de concevoir l'algorithmique dans un sens large semble importante. Comme le souligne Maurer (1998) concernant l'enseignement de l'algorithmique :

In any event, algorithmics does not mean performing algorithms over and over by hand. Algorithms will be carried out more and more by machines or by person-machine combinations. Algorithmics is thinking about algorithms, not thinking like algorithms (Op. cité, p. 24)

2. Exemples d'interactions entre pensées mathématique et algorithmique

Dans l'ouvrage *Thinking mathematically* (Mason et al. 1985), les auteurs proposent une initiation au raisonnement mathématique grâce à une heuristique, développée à travers une série de problèmes mathématiques et de réflexions. Parmi les problèmes proposés, nous avons relevé trois exemples permettant d'illustrer et d'initier un questionnement sur la nature de la pensée algorithmique.

Patchwork

Take a square and draw a straight line right across it. Draw several more lines in any arrangement so that the lines all cross the square, and the square is divided into several regions. The task is to colour the regions in such a way that adjacent regions are never coloured the same. (Regions having only one point in common are not considered adjacent.) How few different colours are needed to colour any such arrangement?

¹ Pouvant induire, selon nous, beaucoup de conceptions erronées concernant algorithme et algorithmique.

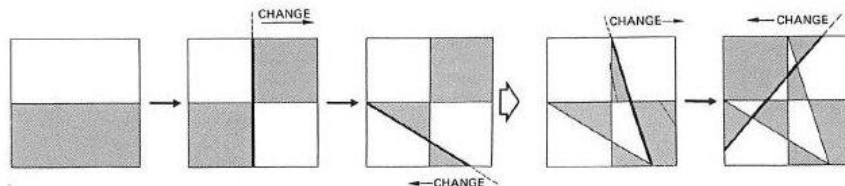
² Cette définition s'inspire de (Knuth, 1985) et (Maurer, 1998), les termes en gras mettent en avant les caractéristiques importantes de l'algorithme.

Figure 1 – Exemple 1 (Mason et al. 1985, p. 11)

Exemple 1 : La solution à ce problème est que l'on peut toujours colorier les différentes zones avec seulement deux couleurs (ici blanc et noir). La méthode proposée pour réaliser une telle coloration et pour montrer que cela est toujours possible est algorithmique :

On part du carré vide colorié en blanc.
 On ajoute les droites de l'arrangement une par une et à chaque fois :
 On inverse les couleurs des zones se situant d'un côté de la droite.

Cette méthode est illustrée sur un cas précis :

**Figure 2 – Illustration de l'algorithme de résolution de l'exemple 1 (Mason et al. 1985, pp. 15-16)**

Une preuve de cette méthode est avancée :

Each side of the new line is properly coloured because adjacent regions were coloured differently by the old colouring; regions which are adjacent along the new line are also coloured differently. (Op. Cité, p.16)

Ce problème amène à une résolution via un algorithme et à la preuve de cet algorithme. Il semble donc qu'une pensée algorithmique soit en jeu avec ce problème.

<i>Diagonals of a Rectangle</i>	
<p>On squared paper, draw a rectangle three squares by five squares, and draw in a diagonal. How many grid squares are touched by the diagonal?</p>	<p>Entry</p> <ul style="list-style-type: none"> ○ What is meant by touched? You decide! ○ Specialize. Be systematic. <p>Attack</p> <ul style="list-style-type: none"> ○ Obviously you can do it by counting, so generalize! ○ Focus on just the horizontal grid lines. Make a table. ○ Look for a pattern. Check it! <p>Extend</p> <ul style="list-style-type: none"> ○ What if the diagonal is replaced by a lawnmower cutting a diagonal swatch of grass? ○ What if the rectangle does not have integral sides? ○ What about three dimensions? ○ What if the grid lines are not equally spaced?

Figure 3 – Exemple 2 (Mason et al. 1985, p. 166)

Exemple 2 : Ce problème et les extensions proposées évoquent l'étude des droites discrètes, question algorithmique encore très vive en informatique et en mathématiques.

<i>Not Cricket</i>	
<p>Amongst nine apparently identical cricket balls, one is lighter than the rest which all have the same weight. How quickly can you guarantee to find the light ball using only a makeshift balance?</p>	<p>Attack</p> <ul style="list-style-type: none"> ○ Do not assume that a particular ball is the light one! ○ What is the worst that can happen? ○ Are you convinced that it cannot be done in fewer weighings? <p>Extend</p> <ul style="list-style-type: none"> ○ What if there are more than nine balls? ○ What if you know only that one ball has a wayward weight? ○ What if there are two kinds of balls, heavy and light, but unknown numbers of each? ○ What if the balls are all different weights, and I wish to line them up in order of weight?

Entry

- Specialize to fewer balls.
- What sort of thing do you WANT?

Figure 4 – Exemple 3 (Mason et al., 1985, p. 183)

Exemple 3 : Nous avons étudié ce problème dans un contexte didactique (Modeste et al. 2010) et nous avons montré son intérêt pour aborder les questions d'algorithmique.

Ces trois problèmes, utilisés par Mason et al. pour développer le raisonnement mathématique, sont fortement imprégnés d'algorithmique. Le deuxième est même en lien direct avec de réelles problématiques d'informatique. Il semble donc qu'une activité

algorithmique soit présente dans les mathématiques. Cependant, certains problèmes algorithmiques, comme l'exemple 2, montrent que des questions mathématiques proviennent de l'informatique. Il pourrait donc être enrichissant de s'intéresser à la pensée algorithmique en elle-même, sans nécessairement la plonger dans l'activité mathématique.

3. *L'algorithmique dans les programmes du secondaire en France*

Dans les programmes du lycée, l'algorithmique fait depuis 2009 l'objet d'un paragraphe à part (figure 5).

Les quelques lignes que constitue ce paragraphe attribuent à l'algorithmique un rôle important dans l'activité mathématique et transversal aux différents champs des mathématiques. Cependant, l'accent est mis essentiellement sur la programmation, la mise en œuvre d'algorithmes et la description d'algorithmes, les différents objectifs attendus des élèves le montrent bien. À aucun moment le lien entre algorithme et résolution de problèmes n'est abordé et il semble que l'objectif principal d'un enseignement d'algorithmique soit l'apprentissage de la rigueur.

Dans une perspective d'enseignement, il est fondamental de ne pas réduire les questions d'algorithmiques à des questions de programmation, de rédaction, et d'exécution d'algorithmes.

Les algorithmes et leur étude jouent un rôle important dans la pensée mathématique, et la pensée algorithmique demande d'être analysée de manière détaillée afin de lui donner sa place dans l'activité mathématique en classe. Dans cette optique, il est important de savoir quels sont les problèmes qui mettent réellement en jeu une activité algorithmique.

Algorithmique (objectifs pour le lycée)

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, les élèves ont rencontré des algorithmes (algorithmes opératoires, algorithme des différences, algorithme d'Euclide, algorithmes de construction en géométrie). Ce qui est proposé dans le programme est une formalisation en langage naturel propre à donner lieu à traduction sur une calculatrice ou à l'aide d'un logiciel. Il s'agit 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.

Dans le cadre de cette activité algorithmique, les élèves sont entraînés :

- à décrire certains algorithmes en langage naturel ou dans un langage symbolique ;
- à en réaliser quelques uns à l'aide d'un tableur ou d'un petit programme réalisé sur une calculatrice ou avec un logiciel adapté ;
- à interpréter des algorithmes plus complexes.

Aucun langage, aucun logiciel n'est imposé.

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.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de donner aux élèves de bonnes habitudes de rigueur et de les entraîner aux pratiques systématiques de vérification et de contrôle.

<p>Instructions élémentaires (affectation, calcul, entrée, sortie).</p> <p>Les élèves, dans le cadre d'une résolution de problèmes, doivent être capables :</p> <ul style="list-style-type: none"> • d'écrire une formule permettant un calcul ; • d'écrire un programme calculant et donnant la valeur d'une fonction ; <p>ainsi que les instructions d'entrées et sorties nécessaires au traitement.</p>
<p>Boucle et itérateur, instruction conditionnelle</p> <p>Les élèves, dans le cadre d'une résolution de problèmes, doivent être capables :</p> <ul style="list-style-type: none"> • de programmer un calcul itératif, le nombre d'itérations étant donné ; • de programmer une instruction conditionnelle, un calcul itératif, avec une fin de boucle conditionnelle.

Figure 5 – Extrait des programmes du lycée en France, passage concernant l'algorithmique

4. *Problématique et organisation de l'article*

Il nous semble important de se questionner sur la place et le rôle de l'algorithme en mathématiques.

Q1 : Qu'est-ce que la pensée algorithmique et quelles sont ses caractéristiques ?

Q2 : Quelles relations entretiennent la pensée mathématique et la pensée algorithmique ?
La pensée algorithmique et les diverses pensées mathématiques ?

Q3 : Quels intérêts didactiques à distinguer la pensée algorithmique de/dans la pensée mathématique ?

Nous nous focaliserons essentiellement sur la relation entre pensée mathématique et algorithmique d'un point de vue épistémologique et nous soulèverons certains aspects importants pour la mise en œuvre et l'apprentissage de ces pensées. Nous articulerons ces questions suivant deux visions de l'algorithmique : la première comme branche des mathématiques, la seconde dans un cadre plus général comme science à part, l'informatique.

5. *La « Pensée » ?*

Commençons par préciser ce que nous entendons par pensée (mathématique ou algorithmique). Le terme provient des recherches anglo-saxonnes sur le *mathematical thinking*. Concernant le sens des termes *thinking* ou pensée nous nous référons notamment à Tall (1991) et Rasmussen et al. (2005) :

[We] take a critical look at advanced mathematical thinking as part of the living process of human thought rather than the immutable final product of logical deduction. (Tall 1991, p.21)

Nous faisons donc référence à la pensée (mathématique, algorithmique, etc.) comme activité de l'esprit humain et non comme son produit. D'autre part, nous utiliserons, comme le font Rasmussen et al. (2005), le terme d'activité (mathématique, algorithmique...) de manière équivalente au terme pensée, afin d'insister sur le fait que nous n'opposons pas ici la pensée à l'action, mais que nous parlons de pensée en tant qu'activité globale, à la fois intellectuelle et pratique :

We also use the term activity, rather than thinking. This shift in language reflects our characterization of progression in mathematical thinking as acts of participation in a variety of different socially or culturally situated mathematical practices. [...] We view the relationship between doing and thinking to be reflexive in nature, not dichotomous. (Rasmussen et al. 2005, p. 51)

On retrouve aussi souvent le terme de « advanced mathematical thinking », qu'il faut comprendre au sens de la pensée (ou l'activité) du chercheur en mathématiques. Nous nous placerons par rapport à cela dans la même optique que Harel et Sowder (2005) :

Advanced mathematical thinking, usually conceived as thinking in advanced mathematics, might profitably be viewed as advanced thinking in mathematics (advanced mathematical-thinking). (Op. cité, p.1)

Nous concevrons l'*advanced mathematical thinking* de cette façon, comme la pensée avancée en mathématique. Nous donnerons un sens équivalent à un « advanced algorithmic thinking ».

II. LA PENSÉE ALGORITHMIQUE EN TANT QUE PENSÉE MATHÉMATIQUE

1. Considérations épistémologiques

La pensée algorithmique, contrairement à la pensée mathématique, n'a été que très peu étudiée. Dans les travaux où on la rencontre, elle est souvent abordée comme une pensée mathématique en particulier. C'est ce point de vue que nous allons adopter dans un premier temps.

Mingus et Grassl (1998), étudient la vision d'enseignants de mathématiques en activité et en formation concernant la pensée algorithmique, la pensée récursive et le *problem-solving*. Ils se placent implicitement à l'intérieur de la pensée mathématique et définissent l'*algorithmic thinking* de la manière suivante :

Algorithmic thinking is a method of thinking and guiding thought processes that uses step-by-step procedures, requires inputs and produces outputs, requires decisions about the quality and appropriateness of information coming and information going out, and monitors the thought processes as a means of controlling and directing the thinking process. In essence, algorithmic thinking is simultaneously a method of thinking and a means for thinking about one's thinking. (Op. Cité, p. 34)

Les auteurs donnent comme modèle (au sens de modélisation) de la pensée algorithmique, l'heuristique de résolution de problèmes proposée par Pólya (1945). Ceci pose la question de la différence entre *problem-solving* et pensée algorithmique, ainsi que la différence avec l'activité mathématique en générale (l'heuristique de Pólya concernant la résolution de n'importe quel problème mathématique). Il nous semble que les auteurs font plutôt référence ici à ce que l'on pourrait traduire par une pensée « algorithmisée », c'est-à-dire un effort pour forcer sa pensée, en mathématique par exemple, à suivre un cheminement établi, systématique, avec un contrôle sur son produit.

Nous nous distinguons d'une telle approche. Nous nous intéressons ici, non pas à rendre la pensée algorithmique (à l'« algorithmiser »), mais bien à l'étude de la pensée (ou l'activité) **en** algorithmique³. Dans cette optique, la définition précédente nous semble trop ambiguë et non effective pour notre questionnement.

Hart (1998) parle, quant à lui, d'*algorithm problem solving*, un thème central dans toutes les branches des mathématiques discrètes, qu'il définit comme l'étude de la question : une solution peut-elle être construite de manière effective ? On trouve ici encore le *problem-solving* et l'algorithmique mêlés. La pensée algorithmique serait alors une façon d'aborder un problème en essayant de systématiser sa résolution, de se questionner sur la façon dont des algorithmes pourraient ou non le résoudre. On peut rapprocher cette définition de la vision constructiviste des mathématiques.

Notre vision de la pensée algorithmique se rapproche de celle de Hart (1998). L'activité mathématique étant centrée sur la résolution de problèmes, la pensée algorithmique, en tant que pensée mathématique parmi d'autres, serait une approche particulière des problèmes mathématiques (de certains en tout cas).

Ces problèmes sont généralement issus d'une branche particulière des mathématiques, les mathématiques discrètes ou bien d'une discrétisation d'objets du continu (problèmes d'approximation en analyse, etc.).

³ De même que les pensées algébriques ou statistiques ne sont pas la conduite de sa pensée de manière algébrique ou statistique mais les modes de pensée en jeu dans les branches concernées.

2. Perspectives pour l'enseignement

Rasmussen et al. (2005) s'intéressent à l'*advanced mathematical thinking* dans une perspective éducative. Dans une description de ce qu'ils nomment *advancing mathematical activity*, ils se focalisent sur trois grandes pratiques dans l'activité mathématique :

Students' symbolizing, algorithmatizing, and defining activities are three examples of such social or cultural practices. These three mathematical practices are not meant to be exhaustive, but represent a useful set of core practices that cut across all mathematical domains. Another significant mathematical practice, one that we leave to later analysis, is justifying. (Op. cité, p. 52)

Attardons-nous sur ce qu'ils nomment l'*algorithmatizing*. Bien qu'ils utilisent le terme algorithme dans un sens plus large que le nôtre, « as a reference for a generalized procedure that is effective across a wide range of tasks » (Op. Cité, p. 63), leur approche nous paraît enrichissante :

Keeping this activity perspective of algorithms in the forefront suggests that instead of focusing on the acquisition of these algorithms, we can characterize learning to use and understand algorithms as participating in the practice of algorithmatizing. By examining the activity that leads to the creation and use of artifacts, as opposed to the acquisition of the artifacts, we view mathematical learning of and in reference to algorithms through a different lens. (Op. cité, p.63)

L'activité algorithmique ne se résume donc pas à l'apprentissage et la mise en œuvre d'algorithmes mais englobe aussi leur production, leur compréhension et leur étude. Rasmussen et al. (2005) montrent ensuite qu'il est possible et pertinent de mettre des étudiants en situation d'*algorithmatizing*.

Dans de précédents travaux (Modeste et al. 2010), nous avons montré qu'aborder l'algorithme et l'algorithmique en tant qu'objet et pas uniquement en tant qu'outil était possible en classe :

On peut questionner les différents aspects de ce concept [l'algorithme] dans certaines situations, et tenter de faire apparaître l'algorithme comme un véritable objet mathématique. (Op. cité, p. 71)

En mathématiques, l'activité algorithmique peut se définir comme une étape dans la résolution d'une certaine catégorie de problèmes dans laquelle on recherche un procédé systématique et effectif de résolution. Il est possible d'aborder de manière utile l'objet algorithme et la pensée algorithmique en classe dans le cadre d'une réelle activité mathématique (résolution de problèmes, preuve de solutions, etc).

Cependant, il est peut-être restrictif de voir la pensée algorithmique comme une pensée des mathématiques et il nous semble que l'étude de la pensée algorithmique comme la pensée en jeu dans la science informatique (l'*algorithmics* de Knuth) peut enrichir notre approche.

III. PENSÉE ALGORITHMIQUE VERSUS PENSÉE MATHÉMATIQUE

1. Considérations épistémologiques

Donald Knuth, mathématicien de formation, devenu l'un des pionniers de l'informatique et auteur des ouvrages de références *The Art of Computer Programming*, s'est beaucoup intéressé aux liens qu'entretiennent mathématiques et informatique. Dans l'article *Algorithmic thinking and mathematical thinking* (Knuth 1985), il aborde la question du rôle de l'algorithmique dans les sciences mathématiques et se demande ce qui différencie *algorithmic*

thinking et *mathematical thinking*⁴. Il formule « Do most mathematicians have an essentially different thinking process from that of most computer scientists? »

Knuth s'intéresse donc à la pensée algorithmique comme étant celle de l'informatique. Il s'appuie sur des ouvrages de mathématiques et sur des preuves qui y sont présentées pour analyser la pensée mathématique et la comparer à la pensée algorithmique. Il divise la pensée mathématique en neuf « modes de pensée » qu'il résume dans le tableau de la figure 6.

Précisons que Knuth ne considère pas ce découpage comme parfait ou immuable : « Thus, I am not all certain of the categories; they are simply put forward as a basis for discussion » (Knuth 1985, p.180).

En s'appuyant sur ce découpage, il analyse ce qu'il considère comme la pensée algorithmique, laquelle englobe parmi les manières de raisonner précédentes : la représentation de la réalité, la réduction à des problèmes plus simples, le raisonnement abstrait, les structures d'information, le recours à des algorithmes et, dans une moindre mesure, la manipulation de formules.

(Two *'s means a strong use of some reasoning mode, while one * indicates a mild connection).

	Formula manipulation	Representation of reality	Behavior of function values	Reduction to simpler problems	Dealing with infinity	Generalization	Abstract reasoning	Information structures	Algorithms
1 (Thomas)	**	**	**						
2 (Lavrent'ev/Nikol'skiï)	**		*		**				
3 (Kelley)	*					**	**		
4 (Euler)	**		**	*		**			*
5 (Zariski/Samuel)	*			*	**	*	**	**	
6 (Kleene)	*					**	**		*
7 (Knuth)	**	*		*					
8 (Pólya/Szegő)	**		**	**	**				
9 (Bishop)	**		**	**		*	**	**	*
"Algorithmic thinking"	*	**		**			**	**	**

Figure 6 – Les différentes pensées mathématiques selon Knuth (1985)

On pourrait reprocher à Knuth de mélanger pensée (algorithmique ou mathématique) et les questions auxquelles cette pensée s'applique ou les objets dont elle traite. Cependant, son analyse peut enrichir notre questionnement : est-ce que ce ne sont pas plus le type de questionnement et les objets étudiés qui distinguent mathématique et algorithmique ?

Wilf (1982) semble aller dans ce sens : dans un article intitulé *What is an Answer*, il analyse ce qui est une réponse acceptable dans un problème de dénombrement et montre notamment qu'une formule de dénombrement peut être parfois plus complexe⁵ à évaluer que l'ensemble étudié à énumérer. Dans un tel cas, peut-on considérer la réponse produite comme une réponse acceptable du point de vue mathématique ? Du point de vue algorithmique ?

De tels exemples ne sont pas uniquement liés aux mathématiques discrètes. Par exemple, une formule exacte donnant la valeur d'une intégrale est-elle toujours plus simple à évaluer que l'intégrale elle-même ? Un tel exemple est développé par Maurer (1998).

Il semblerait que ce qui caractérise la pensée algorithmique soit plutôt une préoccupation quant à l'effectivité des résultats recherchés qu'une réelle différence dans l'activité.

⁴ Comme nous l'avons dit plus haut, pour Knuth, *algorithmics* prend le sens d'informatique (*computer science*).

⁵ Au sens de la complexité algorithmique.

Cependant, Knuth (1985) note aussi que deux aspects fondamentaux de la pensée algorithmique ne sont pas présents dans sa liste des modes de pensée mathématiques : la notion de complexité et l'opération d'assignation qu'il symbolise par « := ».

Selon lui, ces deux aspects pourraient être ce qui différencie les pensées mathématiques et algorithmiques.

La notion de **complexité** est relative à l'efficacité d'un algorithme ou d'une procédure effective. Elle englobe toutes les questions se rapportant au nombre d'étape de calcul, au temps, ou à l'espace mémoire, nécessaires à l'exécution d'un algorithme, dans le pire des cas ou en moyenne. La notion de complexité est clairement liée à des problématiques de mathématiques constructives ou effectives, cependant elle précise ces notions et permet une hiérarchisation des algorithmes selon certains critères d'effectivité réelle, ou d'efficacité.

La notion d'**assignation** évoque ici le concept de variable informatique, par opposition à variable mathématique. Une variable informatique désigne un emplacement dans la mémoire et son contenu peut changer. L'opération d'affectation d'une valeur à une variable est une opération non-symétrique, contrairement à l'égalité. Cependant, la notion d'affectation (ainsi que le concept de variable informatique) est en lien étroit avec la notion de complexité. En effet, c'est l'utilisation de variables informatiques qui permet la production d'algorithmes efficaces en termes de complexité en espace.

L'absence de ces deux aspects dans la pensée mathématique peut se résumer par les remarques de Knuth concernant Bishop :

Bishop's mathematics is constructive, but it does not have all the ingredients of an algorithm because it ignores the "cost" of the constructions. [...] The assignment operation in Bishop's constructions aren't really assignments, they are simply definitions of quantities, and those definitions won't be changed. (Knuth 1985, p.181)

En d'autres termes, la pensée des mathématiques constructives diffère de la pensée algorithmique par le fait qu'elle ne questionne pas l'efficacité des algorithmes qu'elle produit, et qu'elle ne cherche pas toujours à décrire ces algorithmes.

Dans un autre article, traitant des mêmes questions, Knuth (1974) aborde les interactions entre informatique et mathématiques et évoque les diverses façons dont l'informatique influence les mathématiques. Il note que l'informatique permet d'effectuer certains calculs inaccessibles et de valider ou d'invalider certaines conjectures, que l'informatique a entraîné une certaine insistance sur les constructions en mathématiques (notamment par le remplacement de preuves constructives par des algorithmes), que l'informatique permet la construction de certaines bijections et que l'informatique a complètement étendu et enrichi le champ des mathématiques discrètes. Cependant, l'impact le plus notable selon Knuth (1974), est que l'informatique apporte de nouveaux problèmes.

The most significant thing is that the study of algorithms themselves has opened up a fertile vein of interesting new mathematical problems; it provides a breath of life for many areas of mathematics that had been suffering from a lack of new ideas. (Op. Cité)

Le mathématicien Herbert S. Wilf, dans l'article *Mathematics: An Experimental Science* (Wilf 2005), évoque lui aussi les changements apportés par l'informatique aux mathématiques. Il se concentre principalement sur les interactions possibles lors des phases d'expérimentation et de conjecture dans l'activité mathématique. La pensée mathématique se trouve donc changée par l'informatique, et son aspect expérimental s'en trouve renforcé :

A mathematician can act in concert with a computer to explore a world within mathematics. From such explorations there can grow understanding, and conjectures, and roads to proofs, and phenomena that would not have been imaginable in the pre-computer era. (Wilf 2005)

Finalement, l'informatique est source de questions d'ordre mathématiques mais aussi de questions sur l'essence des mathématiques. Ces questionnements doivent aussi atteindre l'enseignement des mathématiques, dans ses pratiques mais aussi dans ses contenus.

The most surprising thing to me, in my own experiences with applications of mathematics to computer science, has been the fact that so much of the mathematics has been of a particular discrete type. [...] Such mathematics was almost entirely absent from my own training, although I had a reasonably good undergraduate and graduate education mathematics. Nearly all of my encounters with such techniques during my student days occurred when working problems from the *American Mathematical Monthly*. I have naturally been wondering whether or not the traditional curriculum – the calculus courses, etc. – should be revised in order to include more of these discrete mathematical manipulations, or whether computer science is exceptional in its frequent application of them. (Knuth 1974, p.329)

2. Perspectives pour l'enseignement

Une vision de la pensée algorithmique, non seulement comme *une* pensée mathématique mais aussi comme *la* pensée informatique semble élargir les perspectives pour son enseignement.

Les aspects que Knuth présente comme absents de la pensée mathématique - description effective d'algorithmes et étude de leur complexité - nous paraissent être propices à une véritable mise en œuvre de la pensée algorithmique. De plus, ces aspects sont de plus en plus pris en compte par les mathématiciens qui travaillent avec des algorithmes, et la frontière entre pensée algorithmique et pensée mathématique n'en devient que plus floue.

Selon le mathématicien Lovász (1988, 2007), l'activité mathématique a subi des changements importants durant les cinquante dernières années et le développement de l'informatique en est la source la plus importante. Il résume son questionnement ainsi :

Is algorithmic mathematics of higher value than classical, structure-oriented, theorem-proof mathematics, or does it just hide the essence of things by making them more complicated than necessary? Does teaching of an algorithm lead to a better understanding of the underlying structure, or is it a more abstract, more elegant setting that does so? (Lovász 1988, p.1)

Il insiste sur le fait que l'algorithmique peut apporter un regard nouveau sur des problèmes anciens et se base sur divers exemples. Il s'interroge alors sur les implications de tels changements sur l'enseignement des mathématiques, et propose des pistes pour faire entrer l'algorithmique dans l'enseignement des mathématiques. Lovász insiste sur le fait que la création d'algorithmes devrait être abordée avant l'apprentissage de leur exécution. Dans un deuxième temps, il considère que l'étude d'algorithmes et de leur analyse devrait être abordée au même titre que celle de théorèmes et de leurs preuves. Il met aussi en garde contre une utilisation trop rapide de l'ordinateur dans l'étude des algorithmes.

The route from the mathematical idea of an algorithm to a computer program is long. It takes the careful design of the algorithm; analysis and improvements of running time and space requirements; selection of (sometimes mathematically very involved) data structures; and programming. In college, to follow this route is very instructive for the students. But even in secondary school mathematics, at least the mathematics and implementation of an algorithm should be distinguished. (Lovász, 1988, p.10)

D'autre part, il insiste sur le fait que l'activité mathématique est en plein changement pour plusieurs autres raisons et insiste sur le fait que l'enseignement des mathématiques doit suivre cette évolution :

In addition, as we will see, we should put more emphasis on (which also means giving more teaching time to) some non-traditional mathematical activities like algorithm design, modeling, experimentation and exposition. I also have to emphasize the necessity of preserving problem solving as a major feature of teaching mathematics. (Lovász 2007, p.3)

Il semble donc important que la pensée algorithmique prenne une place dans l'enseignement des mathématiques. Certaines branches des mathématiques paraissent plus favorables que

d'autres au développement de cette pensée. Les mathématiques discrètes, qui se situent à la frontière entre mathématiques et informatique, peuvent être un champ fertile pour développer la pensée algorithmique et sont un domaine favorable à l'expérimentation, à la modélisation, au problem-solving et la preuve (Ouvrier-Bufferet 2009).

Des expérimentations ont été menées en classe avec des résultats convaincants, concernant l'algorithmique en théorie des graphes, optimisation combinatoire ou dans d'autres branches des mathématiques discrètes (Schuster 2004 ; Hart 1998 ; Modeste et al. 2010).

IV. CONCLUSIONS ET PERSPECTIVES

La pensée algorithmique peut être vue comme faisant partie de la pensée mathématique et nous avons évoqué certaines de ses caractéristiques. Cependant, voir la pensée algorithmique comme pensée majeure de l'informatique permet un réel enrichissement de son analyse. Nous avons montré en quoi cette pensée algorithmique influe sur la pensée mathématique. Il nous semble donc que, pour aborder la pensée algorithmique, il soit indispensable d'appréhender simultanément les deux points de vue, intra-mathématique mais aussi extra-mathématique.

Dans ce contexte, une étude détaillée des différences et similarités entre pensée mathématique et algorithmique reste à développer et nous l'avons vu, ce sont plus les objets et les questions en jeu que l'activité elle-même qui diffèrent. Notamment, il nous semblerait approprié d'étudier en quoi les modèles pour la pensée mathématique et son enseignement (par exemple ceux de Rasmussen et al. (2005) ou de Dreyfus (1991)) peuvent se différencier de modèles pour la pensée algorithmique.

L'apparition dans les curricula de l'algorithmique, n'est pas liée à un simple effet de mode et doit être accompagnée d'une réelle réflexion sur les liens entre pensée algorithmique et pensée mathématique. En effet, une certaine méconnaissance de l'objet algorithme par les enseignants est à envisager :

We can assume that this lack of knowledge about algorithm is shared by teachers of mathematics (at least in France) and their training curriculum has to be questioned. (Modeste et Ouvrier-Bufferet 2011)

L'étude des conceptions des enseignants et de leur rapport à la pensée algorithmique nous semble être d'une grande importance pour l'avenir, afin d'adapter les formations initiale et continue⁶ aux évolutions des mathématiques et de leur enseignement évoquées ici.

Dans cette optique, l'étude de problèmes mettant en jeu la pensée algorithmique, dans un premier temps, permettrait la caractérisation de cette pensée. Dans un second temps, elle permettrait le développement de ressources pour les enseignants.

Avant cela, une analyse détaillée de ce qu'est l'activité algorithmique est nécessaire et doit tenir compte de la relation des mathématiques à l'informatique. Une caractérisation des problèmes et des champs dans lesquelles la pensée algorithmique est pleinement à l'œuvre est alors indispensable. Les analyses présentées ici amorcent une telle étude et constituent selon nous, une base théorique nécessaire pour construire un ensemble de situations pour l'algorithmique. Le champ des mathématiques discrètes représente une bonne piste pour cela.

Ajoutons qu'un enseignement d'informatique est sur le point d'être introduit dans l'enseignement secondaire en France⁷. Il est alors intéressant de retourner la question que nous nous posions au départ : celle de la pensée mathématique dans la pensée informatique. En

⁶ La majorité des enseignants de mathématiques n'ont reçu aucune formation en algorithmique dans leur cursus.

⁷ Un enseignement de spécialité intitulé « Informatique et sciences du numérique » fera son apparition en terminale scientifique dès la rentrée 2012.

effet, la proximité entre ces deux disciplines laisse à penser qu'un enseignement d'informatique peut être l'occasion de développer une certaine pensée mathématique. Comme le note Knuth :

However, I believe that a similar argument can be made for the proposition that mathematics is a part of computer science! (Knuth 1974, p.325)

REFERENCES

- Dreyfus T. (1991) Advanced Mathematical Thinking Processes. In Tall D. O. (Ed.) (pp. 25-41) *Advanced Mathematical Thinking*. Kluwer: Holland.
- Harel G., Sowder L. (2005) Advanced Mathematical-Thinking at Any Age: Its Nature and Its Development *Mathematical Thinking and Learning* 7, 27–50.
- Hart E. W. (1998) Algorithmic Problem Solving in Discrete Mathematics. In Morrow L. J., Kenney M. J. (Eds.) (pp.251–267). *The teaching and learning of Algorithm in school mathematics*. NCTM 1998 Yearbook. Reston, VA : National Council of Teacher of Mathematics.
- Knuth D. E. (1974) Computer Science and its Relation to Mathematics. *The American Mathematical Monthly* 81(4), 323-343. Réed. avec corrections (1996) In *Selected Papers on Computer Science*, Stanford : Center for the Study of Language and Information, 5-29.
- Knuth D. E. (1985) Algorithmic thinking and mathematical thinking. *The American Mathematical Monthly* 92(1), 170-181. Réed. avec corrections (1996) Algorithms in Modern Mathematics and Computer Science. In *Selected Papers on Computer Science*, Stanford : Center for the Study of Language and Information, 87-114.
- Lovász L. (1988) Algorithmic mathematics: an old aspect with a new emphasis. In *Proc. of the Sixth International Congress on Math. Education, Budapest, 1988*, J. Bolyai Math. Soc. (pp. 67-78).
- Lovász L. (2007) Trends in Mathematics: How they could Change Education ? Conférence Européenne « The Future of Mathematics Education in Europe Lisbonne ». <http://www.cs.elte.hu/~%20lovasz/lisbon.pdf>
- Mason J., Burton L., Stacey K. (1985) *Thinking Mathematically*. Workingham : Addison-Wesley publications.
- Maurer S. B. (1998) What is an algorithm? What is an answer? In Morrow L. J., Kenney M. J. (Eds.) (pp.21-31) *The teaching and learning of Algorithm in school mathematics*. NCTM 1998 Yearbook. Reston,VA: National Council of Teacher of Mathematics.
- Mingus T. T. Y., Grassl R. M. (1998) Algorithmic and Recursive Thinking - Current Beliefs and Their Implications for the Future. In Morrow L. J., Kenney M. J. (Eds.) (pp.32-43). *The teaching and learning of Algorithm in school mathematics*. NCTM 1998 Yearbook. Reston, VA: National Council of Teacher of Mathematics.
- Modeste S., Gravier S., Ouvrier-Bufferet C. (2010) Algorithmique et apprentissage de la preuve. *Repères IREM* (79), 51-72.
- Modeste S., Ouvrier-Bufferet C. (2011) The appearance of algorithms in curricula, a new opportunity to deal with proof? A paraître dans *Proc. of the sixth conference of the European Society for Research in Mathematics Education*. <http://www.cerme7.univ.rzeszow.pl/index.php?id=wgl>

- NCTM (1998) *The Teaching and Learning of Algorithms in School Mathematics*. In Morow L. J., Kenney M. J. (Eds) *The Teaching and Learning of Algorithms in School Mathematics*. Reston, VA: National Council of Teachers of Mathematics.
- Ouvrier-Buffet C. (2009) Mathématiques discrètes : un champ d'expérimentation mais aussi un champ des mathématiques. *Actes du séminaire national de didactique des mathématiques* (pp. 31-45). Université Paris 7. Édité par l'ARDM.
- Pólya G. (1945) *How to Solve It: A New Aspect of Mathematical Method*. Princeton, NJ : Princeton University Press.
- Rasmussen C., Zandieh M., King K., Teppo, A. (2005) Advancing mathematical activity: A view of advanced mathematical thinking. *Mathematical Thinking and Learning* 7, 51-73.
- Schuster A. (2004) About traveling salesmen and telephone networks - combinatorial optimization problems at high school, *ZDM* 36(2), 77-81.
- Tall D. O. (1991) The Psychology of Advanced Mathematical Thinking. In Tall D. O. (Ed.) (pp. 3-21) *Advanced Mathematical Thinking*. Kluwer: Holland.
- Wilf H. S. (1982) What Is An Answer ? *The American Mathematical Monthly* 89(5), 289-292.
- Wilf H. S. (2005) Mathematics: an experimental science. A paraître in *Princeton Companion to Mathematics*, W.T. Gowers Ed., Princeton University Press.
<http://www.math.upenn.edu/~wilf/reprints.html>