

Enseignement des mathématiques,  
informatique et algorithmique

**La quadrature binaire du cercle ?**

## **En guise d'introduction...**

Lorsque j'ai accepté de venir à faire cette série d'interventions dans l'île de la Réunion, j'ai immédiatement perçu l'intérêt scientifique de ce déplacement. Et accessoirement, il faut bien le dire, celui des délices des tropiques, des paysages enchanteurs, des fonds sous-marins magnifiques et des diverses boissons fermentées locales. Oui, tout cela était extrêmement séduisant. Ce que je ne savais pas encore, c'est que j'allais tomber dans un traquenard.

Traquenard en effet, car au milieu des cours d'algorithmique proprement dits, domaine dans lequel je sentais un sol ferme sous mes pieds, s'était subrepticement glissée une conférence que je devais prononcer sur les rapports entre algorithmique, mathématiques et enseignement secondaire. Dominique Tournès et Nathalie Carrié, qui m'ont très gentiment invité, souhaitent, je les cite de mémoire mais sans les trahir, avoir sur ces questions « un œil extérieur ». Eh bien, autant vous le dire tout de suite, vous n'allez pas être déçus. Sur ces problèmes, mon œil est tellement à l'extérieur qu'on peut se demander s'il peut encore distinguer quoi que ce soit à l'intérieur.

Les rapports entre mathématique, enseignement secondaire et algorithmique, disions-nous.

Je ne suis pas mathématicien ; mes études en mathématiques se sont arrêtées à peu de chose près au baccalauréat - un baccalauréat scientifique il est vrai, mais dont les souvenirs se font de plus en plus lointains avec les années. Et je n'ose imaginer la note que j'obtiendrais si je devais le repasser à présent. En ce qui concerne l'enseignement secondaire, je suis tout aussi peu qualifié puisque toute ma carrière d'enseignant, je l'ai intégralement accomplie devant des adultes ou dans

le supérieur. Quant à l'algorithmique, histoire de couronner le tout et de parachever le tableau, je vous avouerai avec un brin de provocation que si je possède une certaine expérience en tant que pédagogue, je ne suis absolument pas un spécialiste de la question. A la base, mes études ont porté sur l'économie, et nullement sur l'informatique, domaine dans lequel je ne suis qu'un autodidacte. Comme vous le constatez, le désastre est donc total ; et me voilà devant vous, censé disserter sur les rapports entre trois domaines dans lesquels je suis parfaitement incompetent. Je m'apprête donc ce soir à illustrer un vieux proverbe, selon lequel : « *si tu sais faire quelque chose, fais-le ; si tu ne sais pas le faire, enseigne-le ; et si tu ne sais pas l'enseigner, forme les enseignants* ». C'est très précisément ce que je vais faire.

Il va de soi que pour les besoins de ma petite provocation, je force un peu le trait ; mais pas tant que cela. Je tenais vraiment à ce que vous sachiez que vous devez prendre ce que je vais vous raconter avec tout le recul - je n'ose dire la méfiance - nécessaires. Sur bien des aspects, vous connaissez la réalité beaucoup mieux que moi, et ce n'est pas difficile. Et moins que toute autre chose, je voudrais apparaître comme le donneur de leçons qui, du haut de son piédestal, connaît mieux que les autres le travail qu'ils accomplissent eux-mêmes quotidiennement. De piédestal, je n'en ai pas. Et tout ce que je m'apprête vous dire est réellement le point de vue d'un oeil totalement étranger à vos problèmes, à votre expérience et à vos compétences. Si mes quelques réflexions parviennent malgré tout cela à vous servir, à nous servir, de base de discussion, ma foi, je considérerais que j'ai honorablement rempli mon contrat.

Mais si j'ai parlé de traquenard, c'est aussi pour une autre raison. C'est que derrière le premier piège évident, celui de mon incompetence

totale ou relative, il s'en cachait un second, plus sournois mais tout aussi dangereux.

En m'intéressant à la question sur laquelle je devais intervenir, je me suis aperçu qu'au risque de me les prendre dedans, je mettais les pieds dans un long, un très long tapis. Un tapis - en fait, un débat, pour ne pas dire une polémique, qui agite les enseignants du secondaire depuis pour ainsi dire 25 ans. Des centaines de pages ont été écrites, des dizaines de colloques organisés, de multiples réunions ministérielles, des commissions officielles et officieuses, qui ont formulé des consignes et des contre-consignes, sur cette question de la place de l'informatique dans l'enseignement secondaire. Une association d'enseignants s'est même constituée, qui milite depuis plusieurs années sur ce seul problème. On n'a donc pas attendu, et heureusement, ma brillante intervention de ce jour sur le sujet pour discuter de ces problèmes, et pour les examiner sous tous les angles et sous toutes les coutures. De part et d'autre, des dizaines d'arguments ont été échangés, des plus pondérés aux plus polémiques.

Alors, je n'aurai pas aujourd'hui la prétention d'apporter sur le sujet un point de vue nouveau, qui n'ait pas déjà été exprimé ou défendu au cours de ce débat. Mes arguments, mes réflexions, je les ai certes tirés pour partie de ma propre expérience, une expérience comme je l'ai dit bien limitée, d'enseignant d'université en algorithmique et d'ancien élève de terminale scientifique. Mais sur le fond, il me semble que les termes de la discussion ont déjà été largement explorés par de nombreuses contributions. Sur ce chapitre également, je plaide donc la modestie et j'implore l'indulgence du jury, car mes propos risquent de ne pas bouleverser la science.

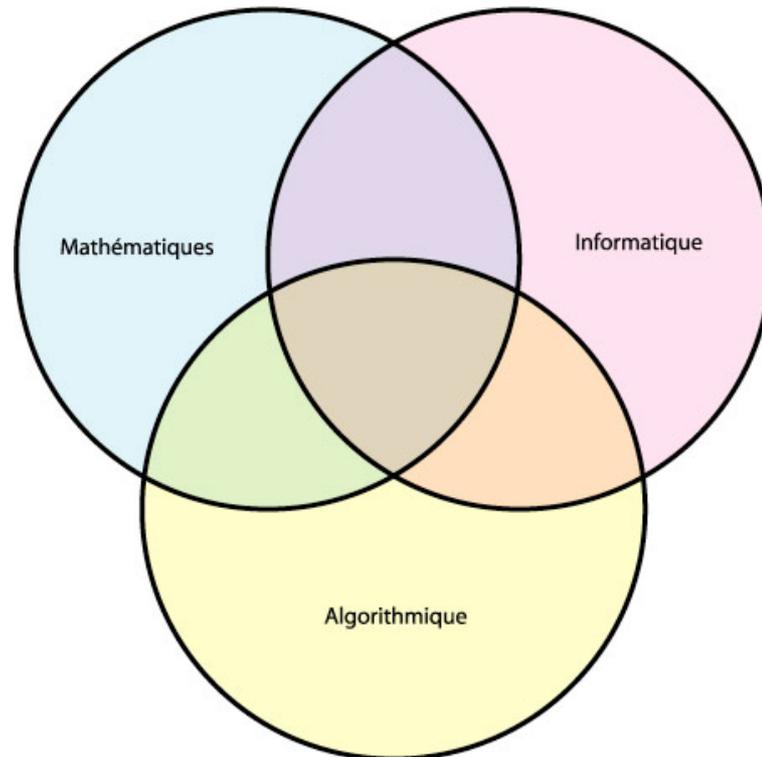
Mais là encore, s'ils éclaircissent quelques points et qu'ils nous permettent de réfléchir ensemble, nous n'aurons déjà pas perdu notre temps.

## **Mathématiques, Informatique, Algorithmique : quelques définitions.**

Après cette longue introduction en forme d'excuses anticipées, je vous propose d'en venir - enfin ! diront certains - au vif du sujet.

Avant toute chose, il me semble que la première chose à faire c'est de clarifier un peu ce dont on parle. En particulier, ce qu'on entend exactement par « informatique » et par « algorithmique ». Il me semble, mais ce n'est qu'un avis personnel, que les mathématiques, l'informatique et l'algorithmique sont trois domaines qui s'ils se recoupent partiellement, ne se confondent absolument pas.

C'est bien connu, un petit dessin valant mieux qu'un long discours, je propose de représenter schématiquement la situation par trois cercles ainsi disposés (même si cela ne vous épargnera pas le grand discours) :



On peut ainsi essayer de préciser ce qui relève des mathématiques mais qui n'est ni informatique, ni algorithmique, ce qui est informatique mais qui n'est ni mathématique ni algorithmique, et ce qui est algorithmique en étant ni mathématique ni informatique. Toujours plus fort et plus compliqué, pour le niveau 2 de la complexité, il y a des choses qui relèvent des mathématiques et de l'informatique mais pas de l'algorithmique, d'autres qui relèvent de l'informatique et de l'algorithmique mais pas des mathématiques, et, vous l'aurez deviné tout seuls, certaines qui sont à la croisée des mathématiques et de l'algorithmique mais qui se situent en dehors de l'informatique. Enfin histoire de terminer en apothéose, il y a le saint Graal, qui est en réalité la situation la moins intéressante pour nous : ce qui concerne les trois domaines à la fois.

Reprenons tout cela en essayant à chaque fois de savoir de quoi l'on parle.

## **Mathématiques**

Je ne vous ferai pas l'injure de définir ce qui relève des mathématiques, même s'il y aurait certainement matière à réfléchir sur cette définition. Mais d'abord vous le savez déjà, ensuite, et peut-être surtout, parce que vous le savez beaucoup mieux que moi. Vous seriez donc probablement consternés par ce que je pourrais en dire. Préférant donc garder un silence pudique sur ce point, je passe au plus vite à un domaine où je me sens un peu plus à mon aise.

## **Informatique**

En revanche, je peux dire quelques mots de l'informatique. Et là, tout de suite, les ennuis arrivent.

D'une manière triviale, en utilisant le bon sens populaire - vous savez, celui qui permet de conclure que la Terre est plate - on pourrait dire que l'informatique concerne les activités qui mettent en oeuvre des ordinateurs.

Néanmoins, cette définition provoquerait immédiatement la réaction courroucée d'un certain nombre de dignes intellectuels. Comment ? Définir l'informatique par le fait d'utiliser des ordinateurs ? Mais c'est une hérésie ! Définirait-on, en effet, l'astronomie comme les activités qui mettent en oeuvre des télescopes, ou la thermique comme les activités qui mettent en oeuvre des thermomètres ? Non, évidemment. Alors, si on ne l'a fait pas pour les autres domaines scientifiques, pourquoi réduire l'informatique à l'utilisation de son principal outil ?

Battant en retraite sous les foudres justifiées des gardiens du temple de la science informatique, il nous faudrait donc

immédiatement retirer de notre définition de l'informatique toute mention aux ordinateurs, pour proposer la seule et vraie maxime : l'informatique, c'est une science, celle du traitement formel des informations.

Pourquoi « formel » ? Parce qu'une machine, lorsqu'elle traite une information, n'a aucune idée de ce que cette information représente, de ce qu'elle signifie. Elle ne peut traiter que sa forme, jamais son contenu. Voilà pourquoi les machines sont capables d'effectuer de manière extrêmement rapide des traitements idiots comme des tris, des recherches ou des calculs, mais qu'on n'a pas pu encore écrire par exemple un seul programme capable de résumer correctement un texte - ni même de le traduire.

Nous voilà donc armés d'une définition infiniment plus rigoureuse, plus fine, pourrait-on dire. Le problème, c'est qu'en l'occurrence, elle est tellement fine qu'elle casse dès que l'on essaye de s'en servir. Parce que les activités qui mettent en œuvre des ordinateurs, que ce soit pour notre usage personnel ou dans le cadre de l'enseignement, tout le monde sait ce que c'est. Et tout le monde, pour avoir passé une nuit blanche sur une machine rétive, sait que cela réclame un type de compétences bien particulier. On peut donc vouloir apprendre à se servir rationnellement d'un ordinateur, d'un traitement de textes et d'un tableur, comme on apprend à se servir d'une automobile, d'une truelle ou d'une perceuse. Peut-on dire pour autant qu'en faisant cela, on acquiert des compétences dans la science du traitement formel de l'information ? Ce serait, je le crois, un tantinet ridicule.

Bien sûr, dans une certaine mesure, on ne peut utiliser rationnellement son ordinateur, même en simple quidam, qu'en possédant un minimum de connaissances, de représentations, sur ce qui

se passe à l'intérieur de la machine, pour éviter d'en avoir une approche « magique ». Autrement dit, on ne peut utiliser rationnellement son ordinateur qu'en possédant quelques notions de cette fameuse et mystérieuse « science du traitement formel de l'information ».

Alors ? Qui a raison ? Les tenants de la définition puriste, ou ceux de la définition pratique ? Les savants, ou les techniciens ?

Manifestement, les deux à la fois, mon capitaine. Toute l'ambiguïté, et le dialogue de sourds qui s'ensuit généralement, vient du fait que le terme d'informatique désigne à la fois une science et une technique. La définition de l'une n'est pas celle de l'autre, et les compétences requises par l'une ne sont pas celles requises par l'autre.

En tant que technique, l'informatique ne peut bien entendu pas se passer d'une certaine connaissance de la « science » de l'information. Tout comme la maîtrise de toute technique demande une connaissance scientifique minimum dans le domaine où elle s'applique. Mais il faut relativiser cette affirmation, et garder le sens de la mesure. On peut devenir un excellent praticien de l'informatique sans pour autant avoir fouillé très loin les concepts et les problèmes abstraits du traitement formel de l'information. Quitte à choquer certains, j'irais jusqu'à affirmer en forçant à peine le trait qu'une connaissance honnête des principes de codage, des tableaux, des types de fichiers et de quelques autres choses est un bagage suffisant pour les neuf dixièmes des informaticiens - je dis bien des informaticiens, et pas seulement des utilisateurs de l'informatique.

Inversement, il faut aussi comprendre qu'on peut être extrêmement savant dans la science du traitement formel des

informations, jongler avec les concepts abstraits avec une facilité désarmante, et être dans le même temps incapable de produire quoi que ce soit de concret avec un ordinateur. Je ne voudrais pas avoir l'air de dire du mal de mes collègues quand ils ne sont pas là, mais je ne dévoilerai pas un secret d'Etat en disant que bien des informaticiens universitaires, incollables sur des raisonnements formels extrêmement complexes, sont totalement démunis face à une machine qui refuse de fonctionner ou face à un logiciel inconnu.

Le savoir scientifique et le savoir technique se recourent partiellement. Mais ils sont loin, très loin, de coïncider.

Et je voudrais tout de suite enfoncer le clou en insistant sur le fait que cette distinction ne recouvre pas du tout, comme on le lit parfois, la distinction entre un « utilisateur » de l'informatique et un « informaticien », c'est-à-dire, pour simplifier, un programmeur. Un programmeur n'est pas davantage - ou guère davantage - censé maîtriser les arcanes de la « science » informatique qu'un mécanicien automobile n'est censé posséder un doctorat de physique et un de chimie. Encore une fois, bien entendu, il ne s'agit pas de faire l'éloge de l'ignorance ; un mécanicien qui croirait que le moteur d'une voiture se met en marche grâce à une énergie cosmique, et que s'il est en panne, c'est parce qu'un sorcier lui a jeté un sort de magie noire, ferait un bien piètre réparateur. Mais si la mécanique demande dans une certaine mesure des connaissances théoriques, précisément, ce sont les mots « dans une certaine mesure » qui sont importants.

Inversement, on peut être un excellent physicien, prix Nobel, et ne pas savoir quoi faire devant un moteur en panne. Tout simplement parce que la mécanique demande, en plus de certains éléments de

physique, tout un ensemble de connaissances et de savoir-faire qui lui sont propres.

En programmation informatique, les professionnels savent très bien que les gens qui sont les plus pointus sur le plan des constructions théoriques ne sont pas forcément - et même, en réalité, ils le sont rarement - mieux armés que des praticiens pour traiter de problèmes concrets. Pour illustrer mon propos, je voudrais juste donner un exemple. En effet, qu'est-ce qu'un bon programme ? Qu'est-ce qui distingue un programme bien fait d'un programme mal fait - à supposer que nous ayons affaire à deux programmes qui fonctionnent sans erreurs ?

Pour le scientifique du traitement formel de l'information, un bon programme est un programme optimisé, c'est-à-dire qui économise des ressources et du temps à la machine. Le scientifique n'hésitera donc pas à passer plusieurs heures à écrire un algorithme plus astucieux, qui permettra de gagner quelques nanosecondes de traitement. Mais un informaticien ordinaire, lui, ne voit pas du tout les choses de cette façon. Car pour lui, un programme optimisé, c'est avant tout un programme simple et lisible, qui sera facilement écrit, facilement repris par un autre programmeur, et dont on assurera ensuite facilement la maintenance. Et qu'importe quelques nanosecondes gagnées par la machine, si l'on doit payer ces nanosecondes par des heures de travail supplémentaire en amont, afin de pénétrer les arcanes d'un programme certes astucieux, mais beaucoup trop compliqué à élaborer et à comprendre. Les logiques du « scientifique » et du « technicien » sont donc deux choses distinctes, tout comme le sont leurs domaines de compétences.

Résumons-nous. Lorsqu'on parle d'informatique, il faut absolument préciser de quoi il s'agit : de la science du traitement formel de l'information, ou de l'ensemble des techniques impliquées par l'emploi des machines à traiter l'information, c'est-à-dire les ordinateurs. Ces deux savoirs, je le répète, sans être totalement disjoints, sont très loin de coïncider. Quant à la division plus traditionnelle entre simple utilisation des ordinateurs et programmation, elle ne recoupe pas la précédente. Je vais avoir l'occasion d'aborder ce point plus précisément en traitant du troisième cercle : l'algorithmique.

### **Algorithmique**

On a vu à propos de l'informatique les problèmes que soulève l'opposition entre une définition savante, scientifique, et une définition pratique, ou technique. Eh bien, nous allons rencontrer peu ou prou les mêmes problèmes en ce qui concerne l'algorithmique.

Au premier abord, l'algorithmique est une activité que nous pratiquons tous quotidiennement, sans aucun apprentissage particulier, un peu comme monsieur Jourdain et sa prose. Si je voulais définir d'une manière simple et, tant qu'à faire, juste, ce qu'est l'algorithmique, je dirais ceci : c'est l'art de donner des instructions à un exécutant afin de lui faire accomplir une tâche donnée.

Bien sûr, c'est avec la programmation des ordinateurs que l'algorithmique a connu un développement est une reconnaissance qu'elle n'avait jamais acquis jusque-là. Mais on identifie trop souvent, et à tort, l'algorithmique en général et la programmation informatique en particulier. À titre d'exemple d'algorithmes de la vie courante qui n'impliquent pas les ordinateurs, je pourrais citer tout simplement le fait d'indiquer son chemin à quelqu'un. On va bel et bien donner à ce

quelqu'un une suite d'instructions (« *Allez tout droit pendant 200 mètres, puis tournez à droite, etc.* ») qui vont l'amener à accomplir une tâche donnée. La rédaction d'un mode d'emploi - ou d'une demande de rançon - sont aussi des exemples courants d'algorithmes qui n'ont rien à voir avec les ordinateurs.

Pourtant, si tout le monde est capable, sans un grand apprentissage, d'indiquer son chemin à quelqu'un, en revanche, dès que l'on essaye d'apprendre à quelqu'un l'algorithmique spécifique à l'informatique, celle qui s'applique aux ordinateurs, le chemin s'avère généralement semé d'embûches. Cela prouve bien que l'algorithmique peut être définie d'une manière plus stricte que celle j'ai proposée à l'instant.

Pour obtenir cette nouvelle définition, plus restrictive, il me suffit en réalité d'introduire un petit mot, qui change tout : le mot « formalisé ». Reprenons donc : l'algorithmique, c'est l'art de donner des instructions formalisées à un exécutant afin de lui faire accomplir une tâche donnée.

Cette nuance fait évidemment écho à ce que nous venons de voir à propos de l'informatique, à savoir qu'il s'agit de la science du traitement formel de l'information.

« Formalisées », qu'est-ce que cela signifie ? D'une part, que ces instructions devront respecter une syntaxe rigide, syntaxe à laquelle bien des gens auront du mal à se plier. Ensuite, que ces instructions seront en nombre limité, et que les données qu'elles pourront manipuler répondront à quelques formes précises. Tout cela peut paraître bien abstrait. Mais dès qu'on essaye d'enseigner l'algorithmique ainsi définie, on se rend bel et bien compte que c'est là

que réside la source des difficultés pour ceux qui veulent l'apprendre. Ramener un problème, aussi compliqué soit-il, à un ensemble d'informations d'un certain type, et à une série de traitements élémentaires sur ces informations, c'est cela la vraie difficulté, le vrai savoir-faire algorithmique.

Histoire de couper définitivement les cheveux en quatre, je tiens à préciser que cette définition stricte, formaliste, de l'algorithmique, inclut l'algorithmique pour les ordinateurs, mais qu'elle ne se limite pas à cela. Il existe d'autres domaines (rares, il est vrai) qui sont couverts par cette définition. Je pense par exemple au bon vieux livre de recettes de cuisine, que tout le monde connaît. Personne ne pense a priori à faire la relation entre un livre de recettes et un programme informatique. Ce sont pourtant deux choses qui sont assez comparables. Dans un livre de recettes, on guide la ménagère étape par étape pour la réalisation du plat ; et ce guide, bien qu'il soit rédigé en langage courant, n'en obéit pas moins à des règles très strictes. Les instructions (couper, bouillir, sauter, faire revenir, etc.) sont en nombre limité. Les ingrédients aussi. La ressemblance avec les programmes informatiques va même plus loin, puisque si chaque recette commence par l'énumération des ingrédients nécessaires, tout programme informatique commence lui aussi par l'énumération des variables, c'est-à-dire des emplacements d'information, nécessaires au bon déroulement des instructions qui suivent.

Cela dit, le vocabulaire de la cuisinière est bien plus étendu que celui des ordinateurs, la cuisinière saura beaucoup mieux qu'une machine corriger d'elle-même les éventuelles imprécisions de la recette, et les tâches qu'on se propose de lui faire faire sont beaucoup

moins variées. Voilà pourquoi il est bien moins difficile de rédiger une recette de cuisine que d'écrire un programme informatique.

Ayant ainsi débroussaillé - ou embrouillé, c'est selon ! - le contenu des différents domaines de base, des trois cercles de mon schéma, nous pouvons à présent partir gaiement à l'assaut de leurs zones de recoupement. Rassurez-vous, cela va aller beaucoup plus vite.

### **Mathématique et informatique**

Il s'agit d'un domaine qui vous intéresse tout particulièrement, dans la mesure où les instructions officielles insistent abondamment sur ce point, et ce dès le collège. En l'occurrence, il est évident que la définition de l'informatique que l'on doit retenir est celle du technicien, et non celle du scientifique. Le but avoué des programmes scolaires, du moins tel que je l'ai compris, c'est d'utiliser les ordinateurs comme une aide à la compréhension des concepts mathématiques, de permettre aux élèves de mieux se représenter ces concepts et leurs rapports avec les situations réelles. De ce point de vue, l'accent est mis sur l'utilisation de certains logiciels, en particulier des capacités de calcul et de grapheur des tableurs ou des logiciels plus spécifiques aux mathématiques.

Il faut noter que les programmes scolaires, à ma connaissance, ne font aucune allusion à la programmation proprement dite. De prime abord, cela ne doit pas nous étonner : la programmation ne permet a priori pas de mieux comprendre les mathématiques, et on ne voit pas très bien ce qu'elle viendrait faire ici. Mais on va y revenir.

### **Informatique et algorithmique**

On se trouve là, en réalité, au cœur de ladite programmation.

Si j'en exclus, par définition, les mathématiques, qui se trouvent à l'intersection de mes trois cercles, alors il s'agira de programmes ne faisant pas particulièrement appel à des connaissances mathématiques, ou ne traitant pas spécifiquement de questions mathématiques. Bien entendu, une telle définition a toujours quelque chose d'un peu arbitraire, dans la mesure où je ne connais aucun programme digne de ce nom qui n'emploie pas ne serait-ce que les opérations mathématiques élémentaires. Mais il y a tout de même une énorme différence entre un programme qui va par exemple compiler les mouvements des comptes en banque pour mettre des comptes clients à jour, en faisant au passage quelques additions, et un programme chargé de calculer la  $n$ ème décimale du nombre Pi, qui mettra en jeu des algorithmes mathématiques spécifiques.

En tout cas, quelqu'un comme moi perçoit très bien la différence entre de l'informatique non mathématique et de l'informatique mathématique : c'est que dans le second cas, je n'ai pas la moindre idée de ce que je dois faire faire à la machine pour qu'elle parvienne au résultat, et que si l'on me montre le programme, je mets plusieurs heures à y comprendre quelque chose. Mais en général, j'abandonne avant.

Au passage, j'en profite pour signaler que la quasi-totalité des cours de programmation, et plus encore, d'algorithmique, s'adressent à des mathématiciens ou du moins à des gens supposés posséder des compétences en mathématiques. Ces cours n'emploient comme exemples que des traitements utilisant des connaissances mathématiques ou s'appliquant à elles. Pourtant, des gens très faibles en mathématiques peuvent apprendre la programmation et y devenir très compétents et performants (tant qu'on ne leur demande pas, bien

sûr, de programmer des problèmes spécifiquement mathématiques). Les non-mathématiciens constituent un vivier considérable de programmeurs potentiels. Mais le résultat infaillible de la méthode pédagogique voulant qu'on ne prenne, pour les programmeurs débutants, que des exemples mathématiques, a été de décourager des générations d'apprentis programmeurs qui avaient peut-être d'excellentes dispositions pour l'informatique, mais qui n'avaient pas le bonheur d'être titulaires de la médaille Fields.

Dans le même élan, j'en profite donc pour régler son compte à la partie centrale de mon schéma : les programmes informatiques spécifiques aux mathématiques. Pour notre propos du jour, ce domaine ne pose pas de problème particulier ; et encore une fois, à ma connaissance, les consignes officielles restent muettes sur l'opportunité de l'enseignement de ces compétences.

### **Mathématiques et algorithmique**

Nous en arrivons enfin au cœur de notre problème : les choses qui relèvent de l'algorithmique et des mathématiques, sans pour autant toucher à l'informatique - bien que la frontière ne soit pas exactement étanche, je vais y revenir.

Commençons par une brève digression culturelle : c'est précisément dans cette zone que se trouve l'origine même du mot algorithmique. Celui-ci provient en effet du nom d'un spécialiste arabe du Moyen Âge, Al Khwarizmi (780-850) qui avait rédigé un traité spécifiant une méthode de résolution pour certaines équations.

Alors, nous retombons, à propos des rapports entre mathématiques et algorithmique, sur le problème des deux définitions de l'algorithmique dont j'ai parlé tout à l'heure.

Si l'on prend la définition la plus large de l'algorithmique, celle qui ne précise pas que les informations et les instructions qui les manipulent doivent être formalisées, il me semble que l'enseignement des mathématiques comporte nécessairement, de fait, une dimension algorithmique, même quand celle-ci n'est pas formulée explicitement. Lorsqu'on explique aux élèves de l'école primaire une technique opératoire, par exemple celle de la multiplication, il s'agit d'un algorithme. À un niveau supérieur, lorsqu'on indique une méthode pour résoudre un système d'équations par le déterminant, ou que l'on donne les règles et les cas pour rechercher les racines d'un polynôme du second degré, il s'agit également d'algorithmes de la plus belle eau, même si le mot n'est pas prononcé.

J'enfoncerai cette porte déjà largement ouverte en généralisant cette proposition à l'ensemble des matières scolaires. Dans toutes les disciplines, on donne aux élèves des techniques, des règles, des démarches à suivre, qu'ils sont censés appliquer dans certaines situations. Ce sont autant d'algorithmes qui ne disent pas leur nom. Et bien des interrogations écrites, des devoirs, consistent à demander aux élèves soit de restituer ces règles, c'est-à-dire de réécrire des algorithmes, soit de les appliquer, c'est-à-dire de les exécuter.

Je ne vous dresserai pas une liste exhaustive des situations où l'enseignement se ramène, en réalité, à un algorithme. Pour faire vite, et ne donner que quelques cas qui me passent par la tête, pour la physique, toutes les formules de calcul sont des algorithmes. En sciences naturelles, toute méthode de classement d'un organisme ou d'un minéral, supposant une suite de questions du genre « a-t-il des poumons ou des branchies, a-t-il des poils des plumes ou des écailles, etc. ? », suite de questions dont on doit tirer une conclusion, est un

algorithme. Mais il n'y a pas que les sciences qui soient concernées. Les matières littéraires elles aussi, comportent ce type de connaissances « algorithmiques ». En français, les règles de grammaire, les règles d'orthographe, les règles de conjugaison, sont autant d'algorithmes. La règle de l'accord du participe passé avec l'auxiliaire avoir, selon qu'il s'agisse ou non d'un complément d'objet direct, que celui-ci soit placé ou non avant le verbe, que le verbe soit transitif ou non, est un merveilleux exemple d'algorithme. Même chose en ce qui concerne les langues étrangères. Il n'y a guère qu'en histoire ou en philosophie pour lesquels, spontanément je ne voie guère de règles algorithmiques évidentes.

Bien sûr, si l'on disait à un enseignant de français pris au hasard qu'en croyant enseigner le français, il fait en réalité de l'algorithmique, je pense qu'il ouvrirait de grands yeux ébahis et qu'il s'interrogerait sur la santé mentale de son interlocuteur. Mais c'est pourtant bel et bien le cas.

Donc, si l'on en reste à la définition large de l'algorithmique, celle-ci se trouve donc à peu près partout dans l'enseignement et j'ajouterai, dans la vie en général.

Cependant, si l'on restreint la définition de l'algorithmique au maniement des traitements formalisés, alors là, le champ d'intervention actuel de l'algorithmique dans l'enseignement se réduit comme peau de chagrin.

Je laisse provisoirement de côté la question de savoir de quelle manière les enseignants de mathématiques traduisent aujourd'hui dans la pratique les consignes des programmes en ce qui concerne l'enseignement de l'algorithmique dans le cadre de leur cours de

mathématiques. J'ai bien ma petite idée là-dessus, mais je fais durer encore un peu le suspense avant de vous en parler. En revanche, ce que je peux dire tout de suite, sans que cela constitue une brûlante révélation, c'est qu'aucun professeur de français, d'anglais, ou d'allemand ne s'amuse à formuler les règles de conjugaison, de déclinaisons, ou de grammaire, en passant par les fourches caudines du formalisme algorithmique. Le bon vieux si... Alors... Sinon ceci... Alors... Constituent le degré suprême de formalisation des règles que l'on inculque aux élèves. Doit-on le regretter ? C'est là une autre question sur laquelle je m'empresserai de revenir dans un instant.

Pour terminer sur ce point, je voudrais ajouter que « faire de l'algorithmique », dit tel quel, cela ne veut rien dire. Ou plus exactement, cela peut vouloir dire trois choses très différentes.

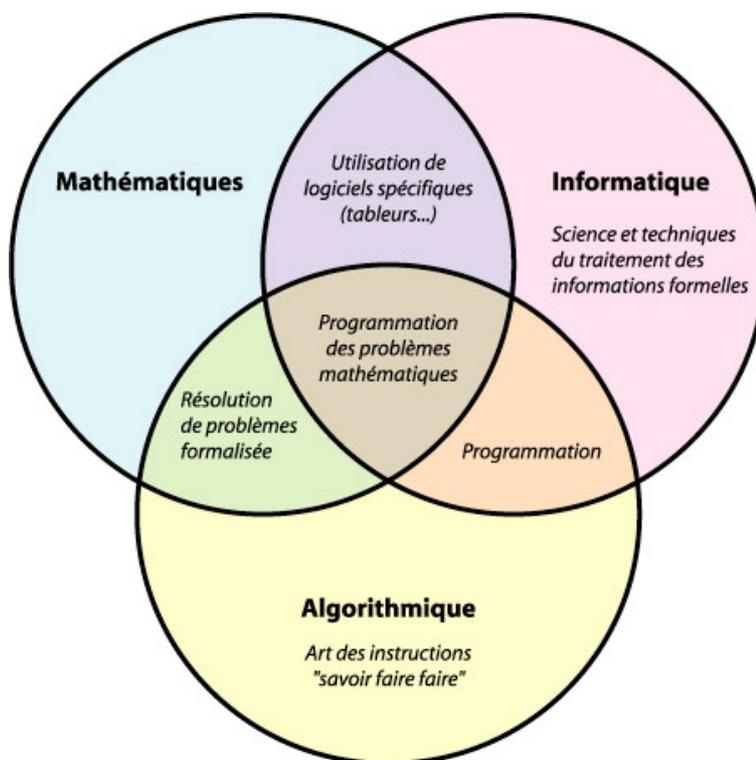
« Faire de l'algorithmique », cela peut signifier appliquer des algorithmes déjà construits. A ce compte là, tout le monde fait toute sa vie de l'algorithmique. Le cuisinier qui lit son livre de recette, l'étudiant en langues qui apprend ses conjugaisons, l'écolier qui apprend des techniques opératoires, tous ces gens appliquent des algorithmes.

« Faire de l'algorithmique », cela peut vouloir dire formaliser un algorithme existant. C'est-à-dire prendre un problème, de maths ou d'autre chose, dont on connaît déjà la méthode de résolution, et exprimer ce problème et cette méthode sous des formes rigides, celles que comprennent les ordinateurs. C'est déjà une toute autre compétence.

Enfin, « faire de l'algorithmique, c'est aussi - et peut-être surtout - imaginer des traitements, les concevoir.

Concevoir des traitements, puis les traduire sous une forme acceptable par les ordinateurs, c'est cela, le travail de l'informaticien. Est-ce de cela dont parlent les programmes actuels de mathématiques ? On peut en douter. Il me semble plutôt que ceux-ci font uniquement allusion au deuxième point : on ne se propose pas d'apprendre aux élèves à concevoir des traitements, mais uniquement à traduire des traitements existants sous une forme informatique.

J'insiste sur ce point, car la difficulté, mais aussi l'intérêt de cet apprentissage, ne sont alors plus du tout les mêmes. C'est toute la différence qu'il y aurait, dans un autre domaine, entre former des romanciers et former des traducteurs. Là encore, on y reviendra.



En conclusion provisoire de ce premier tour d'horizon, on peut donc dire que dans l'enseignement secondaire, si l'algorithmique au sens large est partout, au sens strict, elle n'est pour ainsi dire nulle part. Tout dépend finalement de ce que l'on entend par le terme d'algorithmique.

Mais alors, pourquoi les programmes officiels de mathématiques ont-ils récemment témoigné de la volonté de faire apparaître explicitement l'algorithmique ? Quelle est la raison de cette innovation ? Et quelle signification ce terme d'algorithmique est-il censé recouvrir en l'occurrence (car, comme on l'a vu, c'est un des aspects fondamentaux de la question) ?

Il me semble que pour répondre à tout cela, on ne peut se dispenser d'un petit retour en arrière sur l'histoire de la place de l'informatique dans l'enseignement secondaire. Car je pense pouvoir affirmer que ce flash-back fournit un éclairage indispensable sur la réalité actuelle.

## **L'informatique dans l'enseignement secondaire**

Je le disais en introduction, et je le maintiens, en venant ici, je suis tombé dans un traquenard. Lorsque j'ai accepté de venir parler de tout cela devant vous, je n'avais pas la moindre idée de la quantité de littérature qui avait pu être écrite sur cette question. La place que doit, ou que devrait prendre, l'enseignement de l'informatique dans le cursus des lycéens, et les modalités de cette éventuelle insertion, ont donné lieu depuis pour ainsi dire trente ans à des débats extrêmement fournis et, pour utiliser un euphémisme, passionnés.

## Les premiers « plans informatiques »

Commençons par une révélation d'importance : vous avez devant vous, en chair et en os, une victime, ou plutôt, soyons justes, un heureux bénéficiaire du premier programme de développement de l'informatique dans l'enseignement secondaire. C'était au début des années 1970, et le ministère avait décidé d'équiper en ordinateurs quelques dizaines de lycées en France, 58 très exactement, et de faire développer par des informaticiens un langage spécifique pour l'apprentissage de la programmation, langage en français qui s'appelait le L.S.E.

Et vous êtes donc en train d'écouter religieusement un ancien collégien, qui dès sa classe de sixième, a pu fréquenter la salle informatique à ses heures perdues - car mon collège était aussi un lycée - en compagnie éventuelle d'un des rares enseignants qui avaient pris l'affaire en main, ou bien, le plus souvent, aux côtés de l'un de ses petits camarades qui commençaient à jouer avec la drôle de machine de la fameuse salle.

Quand je dis jouer, entendons-nous bien : les capacités graphiques de cette machine avaient fort peu de choses à voir avec celle des ordinateurs de maintenant, et les seuls jeux qu'il pouvait s'y trouver étaient ceux que nous avions nous-mêmes programmés. Il fallait d'ailleurs avoir une certaine imagination pour inventer des jeux dignes de ce nom, qui puissent être développés sur une machine uniquement capable d'afficher en noir et blanc des caractères sur 24 lignes et 80 colonnes, et encore, à condition qu'il s'agisse de la ligne du bas de l'écran. En réalité, très vite, nous nous rendions compte que le jeu le plus amusant n'était pas de se servir du programme, mais de l'écrire.

Comme plusieurs de mes camarades, j'ai donc appris à programmer sur le tas, en autodidacte, sans même l'aide d'un adulte pour nous donner les connaissances de base. Le manuel du langage était disponible, la débrouillardise aussi, et avec tout cela, pour peu que l'on se pique au jeu, on pouvait fort bien apprendre soi-même. Ce n'est que bien des années plus tard que j'ai découvert et compris les problèmes de formalisation d'un algorithme et du codage de l'information.

Rassurez-vous, je ne vais pas vous raconter ma vie pendant le reste de la soirée ; j'espère que nous avons des choses bien plus intéressantes à nous dire. Mais à ma petite échelle, je pense que ce témoignage est significatif, significatif d'une certaine volonté politique des responsables de l'éducation nationale de cette époque, et significatif des résultats et des limites de cette initiative.

### **L'option informatique**

L'étape suivante fut franchie en 1981, lorsque le ministère décida - ou autorisa, c'est selon les points de vue - la création d'une option informatique au niveau du lycée. Il s'agissait d'une petite révolution, dans la mesure où le débat faisait déjà rage entre les partisans d'une introduction franche, en bonne et due forme, de l'enseignement de l'informatique comme d'une matière à part entière, et ses adversaires. La création de l'option informatique était une victoire des premiers sur les seconds. Elle pouvait, et devait, dans l'esprit de beaucoup, être un premier pas vers des avancées plus résolues. L'option informatique était en effet une initiative qui restait partielle et prudente. Certes, pour la première fois, la volonté de faire de l'informatique une matière à part entière était clairement affichée. Certes, des salles, du matériel, étaient dévolues à cet enseignement. Certes, des enseignants se voyaient reconnaître un contingent d'heures de service à consacrer à

cette matière. Et last but not least, certes, une épreuve optionnelle figurait même au baccalauréat pour sanctionner, et couronner, cet enseignement.

Mais l'option informatique restait une demi-mesure, marquée par un certain nombre de limites. Tout d'abord, au risque d'énoncer une lapalissade, ce n'était qu'une option. Et surtout, comme souvent, les moyens eurent beaucoup plus de mal à voir le jour que les bonnes intentions. Les budgets pour le matériel, apparemment, n'étaient pas le problème le plus criant, même si en bien des endroits, ils pouvaient aussi l'être. Mais il fallait aussi disposer de locaux pour installer ce matériel. Et surtout, il fallait disposer de personnel, tant pour en assurer la maintenance que pour assurer les cours. C'est sans doute là que bât avait le plus tendance à blesser.

Après quelques péripéties et plusieurs années de service, dont je ne sais s'il faut dire qu'ils ont été bons et loyaux, l'option informatique a finalement été définitivement supprimée en 1999. Et si je ne m'abuse, c'est très peu de temps après cette date que les programmes d'enseignement, en particulier de mathématiques, se sont mis à accorder explicitement une place de plus en plus grande à l'informatique.

Deux documents ont marqué cette transition. Il s'agit d'une part du rapport publié par le Conseil National des Programmes en 1991 qui scellait pour un temps le sort de l'option informatique. Et d'autre part, du rapport de la commission Kahane, rédigé en 2000, qui consacrait l'intégration de l'enseignement de l'informatique au sein des programmes de mathématiques, après que l'option informatique eut été définitivement supprimée par Claude Allègre.

## **Des raisons invoquées (1)...**

Pour commencer, je souhaiterais revenir brièvement sur les raisons de la suppression de l'option informatique. À mon humble avis - mais n'est-ce pas mon humble avis qu'on me demande d'exprimer ? - il y avait des raisons avouées, celles qui figuraient explicitement dans le rapport, mais il y avait aussi des raisons qui pour être moins avouables, n'en étaient pas moins réelles.

Les raisons avouées, telles qu'elles furent exposées par le CNP, étaient essentiellement au nombre de trois.

La première d'entre elles portait sur le bilan de l'option informatique, à laquelle il était reproché une dérive : la tendance à n'accueillir que les meilleurs élèves des meilleures classes, et, pourrait-on ajouter, ceci dans les meilleurs lycées. Je n'ai pas trouvé de réfutation explicite de cette accusation ; il semblerait donc que même les partisans de l'option informatique, sur ce point, plaident coupable. En revanche, les raisons de cette dérive mériteraient d'être examinées. Les partisans du maintien de l'option informatique, voire de sa transformation en enseignement obligatoire, insistent sur le fait que la dérive a été le produit de la pénurie. Faute de moyens matériels, faute de moyens humains, faute d'enseignants spécialisés et qualifiés, et par-dessus tout, faute d'être devenu autre chose qu'une simple option, l'enseignement de l'informatique a inmanquablement été conduit à être rationné. Et comme on le sait, tout rationnement est par définition inégalitaire. Aucune règle explicite ne venant définir, en cas de rationnement, qui devait bénéficier de l'option, c'est par une pente toute naturelle que celle-ci a été peu à peu monopolisée par ceux qui y voyaient un atout supplémentaire pour les futurs élèves des classes préparatoires. Je ne dispose de beaucoup d'éléments tangibles pour me

faire une opinion ; apparemment, je ne suis pas le seul, car aucune étude sérieuse n'a semble-t-il été menée sur ce sujet. Mais il me semble que cette explication est rationnelle.

En réalité, donc, au-delà de la dérive sans doute réelle de l'option informatique, le vrai problème posé restait celui d'un enseignement de l'informatique dans le secondaire qui soit spécifique et obligatoire pour tous. Le ministère n'a d'ailleurs pas manqué pas de s'exprimer sur ce sujet, et cela fournit la matière de ses deux autres arguments principaux.

### **Des raisons invoquées (2)...**

L'informatique, toujours selon le ministère, se divise en deux grands domaines ; celui du savoir élémentaire, c'est-à-dire de l'utilisateur, fut-il averti, et un savoir plus pointu et plus spécifique, celui du programmeur.

Premier point, le corpus de connaissances et de savoir-faire nécessaire pour devenir un utilisateur de l'informatique : sur ce plan, toujours d'après le ministère, point n'était besoin d'un enseignement spécifique. Les élèves étaient censés, et ils le sont toujours, acquérir cette compétence en utilisant les machines dans le cadre des autres enseignements.

Second point, les compétences liées à une plus forte technicité, en particulier celle de la programmation. Pour le ministère, cet enseignement n'avait pas sa place dans le secondaire. La raison principale, pour ne pas dire unique, étant que ses compétences ne seront utilisées dans la vie active que par ceux qui deviendront des informaticiens professionnels, et que bien peu parmi les élèves choisiront cette carrière.

Cette argumentation, que je pense résumer mais ne pas caricaturer, me paraît pour le moins légère.

Comme l'ont fait remarquer bien des gens qui se sont exprimés dans ce débat, si un savoir n'a pas à être l'objet d'un enseignement spécifique au motif qu'il est transversal à d'autres matières, alors il faut immédiatement supprimer les deux tiers du contenu des programmes actuels.

Savoir interpréter un énoncé, écrire des phrases correctes, exposer ses idées de manière articulée, voilà des compétences que toutes les matières utilisent, y compris les mathématiques. Dès lors, pourquoi y consacrer plusieurs centaines d'heures d'enseignement spécifiques ? Supprimons ces heures de français ! Les collégiens et les lycéens apprendront bien tout cela en le pratiquant dans d'autres matières.

Savoir compter, calculer, résoudre un problème maniant des quantités, voilà également des savoirs transversaux utilisés dans bien des domaines, ne serait-ce qu'en physique. Alors, pourquoi ne pas supprimer une bonne partie des cours de mathématiques ? Les élèves apprendront tout cela en faisant leurs exercices de physique.

On pourrait multiplier les exemples à l'envi. Mais en tout état de cause, l'idée qu'un enseignement de l'informatique, même réduit aux simples connaissances nécessaires à un utilisateur, l'idée que cet enseignement serait superflu dans la mesure où les élèves se serviront des ordinateurs en cours de mathématiques, de physique, ou de langue, me paraît être un non-sens.

Tout d'abord, et sans être outrageusement provocateur, pour savoir si les élèves sont capables d'apprendre à utiliser les ordinateurs sur le tas, sans enseignement spécifique, on pourrait regarder ce qu'il en est

pour les enseignants. Après tout, eux aussi sont censé avoir appris sur le tas. Alors, combien d'entre eux savent réellement se servir d'un ordinateur ? Pas tant que cela, vous le savez comme moi. Combien se sentent posséder une maîtrise suffisante de la machine pour avoir envie de se présenter devant une classe de 25 ou 30 élèves, en sachant qu'ils pourront faire face à toutes ces petites bizarreries dont l'informatique est coutumière, et qui vous empêchent de travailler comme vous l'aviez prévu ? Encore moins. Ceux d'entre vous qui sont ne serait-ce qu'un peu versés dans la chose informatique savent de quoi je parle : dans le milieu enseignant comme dans tous les autres milieux sociaux, excepté celui des informaticiens, quand on sait que vous vous y connaissez en informatique, vous devenez immédiatement quelqu'un de très intéressant et de très fréquentable. Vous vous faites tous les jours de nouveaux amis, qui vous trouvent très beau, très spirituel, très gentil, et qui ont par ailleurs un petit problème avec leur ordinateur, que vous ne manquerez pas de résoudre en passant faire un tour chez eux.

Alors, pourquoi ce qui est vrai des adultes le serait-il moins des enfants ou des adolescents ? Pourquoi existerait-il des gens dont le métier est d'enseigner la maîtrise de la langue, d'enseigner la maîtrise des nombres, d'enseigner la maîtrise d'une automobile, et pourquoi, face à un ordinateur, devrait-on apprendre soi-même, sans guide, sans concept, autrement dit dans le chaos le plus total ? Je crois qu'il existe hélas une réponse inavouable à cette question, réponse sur laquelle je reviendrai tout à l'heure.

Alors, je sais que tout cela devrait être nuancé par le fait que les cours de technologie, au collège, intègrent désormais l'apprentissage du monde informatique. En toute sincérité, je manque d'éléments d'appréciations sur ce point. J'aimerais pouvoir penser que les élèves,

tous les élèves, bénéficient réellement de ce que les programmes leur promettent. Et en même temps, je ne parviens pas à m'en convaincre, et je reste extrêmement sceptique sur le niveau que les dispositions actuelles permettent réellement de donner aux élèves dans ce domaine. Alors, si vous avez des lumières, des expériences, à ce sujet, je serais très heureux d'en bénéficier.

### **Des raisons invoquées (3)...**

J'en viens à présent à l'enseignement de la programmation, qui quant à lui n'aurait donc pas sa place dans le secondaire, dans la mesure où seuls les futurs informaticiens trouveraient un intérêt à cet enseignement. Là aussi, l'argument me paraît étrange, pour ne pas dire spécieux.

Je commencerai par une observation de simple bon sens : si cet argument est réellement pris au sérieux par ceux qui l'emploient, pourquoi les mêmes ont-ils introduit l'enseignement de l'algorithmique dans les programmes de mathématiques ? Si la programmation et sa logique n'intéressent que les futurs programmeurs, et que cela justifie son bannissement de l'enseignement secondaire, pourquoi réintroduire par la fenêtre ce que l'on vient de chasser par la porte ? Il y a là une incohérence manifeste. A moins que...

Avant de dire à moins que quoi, penchons-nous sur l'argument lui-même. Que doit-on enseigner au lycée ? Doit-on introduire, ou au contraire proscrire, les contenus liés à un savoir professionnel ? Doit-on privilégier ce que l'on appelle la culture générale, et qui peut ou non recouper ces savoirs dits professionnels ? Mes compétences en pédagogie, bien insuffisantes, m'empêchent de donner à cette question une réponse ferme. Je peux donner qu'un avis de béotien.

Il me semble, quitte à enfoncer une porte ouverte, qu'il faut enseigner ce qui est utile. Utile pour se repérer dans le monde tel qu'il est, utile pour apprendre à raisonner, utile pour pouvoir aborder avec des outils intellectuels de nouveaux domaines de connaissances. Et par contrecoup, utile pour se faire sa place dans la société, c'est-à-dire y trouver un emploi.

De ce point de vue, mon appréciation de l'algorithmique et de la programmation est nuancée. Les compétences en programmation proprement dite me semblent beaucoup moins indispensables à l'honnête homme du XXI<sup>e</sup> siècle que les compétences en informatique. Je n'irais pas jusqu'à dire qu'elles n'intéressent que les seuls futurs informaticiens. Après tout, les ordinateurs font tout autant partie de notre environnement que les bactéries, l'électricité, ou les langues étrangères. Et de ce point de vue, il ne me semble pas aberrant que tous lycéens actuels aient une idée de la manière dont on écrit un logiciel, de la même manière que tous les lycéens acquièrent une idée de ce qu'est une bactérie, un circuit électrique, ou une phrase d'anglais. Il y a ensuite, évidemment, une question de dosage : où faut-il placer le curseur ? Je n'en sais rien, et je serais bien incapable de donner une règle simple qui permette de le fixer a priori.

En ce qui concerne l'apport de l'algorithmique et de la programmation à la formation du raisonnement en général, j'aurais tendance à être plus affirmatif. Pour avoir vu quelques promotions d'étudiants issus des sciences humaines suer sang et eau pour apprendre à vérifier mentalement un programme informatique, je suis absolument convaincu que l'algorithmique est une école de méthode et de rigueur intellectuelle. Et la logique, plus exactement, le mode de pensée, du

programmeur, ne coïncident pas exactement avec celui du mathématicien. Elle en est même, à vrai dire, assez éloignée.

Une fois ceci dit, il reste tout de même une question en suspens : c'est que les mathématiques, tout le monde en convient, sont elles aussi une école de rigueur. Or, chaque année, des générations d'élèves, pour une raison ou pour une autre, se révèle incapable de se plier aux exigences de cet enseignement. Là encore, l'analyse des raisons de cet échec ou de cette allergie aux mathématiques dépasse de très loin mes connaissances et mes compétences. Ces élèves sont-ils allergiques à la rigueur en général, au formalisme en général, ou seulement aux mathématiques en particulier ? Si l'allergie porte sur les mathématiques en particulier, cela est-il dû au contenu même des connaissances enseignées, ou à la manière dont celles-ci sont présentées ? Je ne me sens absolument pas apte à répondre à ces questions. Je sais que depuis fort longtemps, des mathématiciens, des pédagogues, de didacticiens, et les enseignants eux-mêmes y réfléchissent et tentent d'y apporter des réponses. Je peux que leur faire confiance, en constatant de très grandes évolutions entre l'enseignement de mathématiques telles que je l'ai moi-même vécu il y a 25 ans, et ce qui se pratique aujourd'hui.

Pour en revenir à l'algorithmique, celle-ci représente incontestablement une école de rigueur. Celle-ci serait-elle plus efficace que l'école mathématique ? Ne ferait-elle pas double emploi avec elle ? On pourrait redouter que les élèves qui accèdent facilement aux notions mathématiques soient les mêmes qui comprennent sans difficulté l'algorithmique, et vice et versa : ceux qui souffrent déjà en mathématiques seraient peut-être les mêmes qui souffriraient en algorithmique. Auquel cas, ajouter un enseignement d'algorithmique à l'enseignement des mathématiques ne serait guère pertinent, si on

considère que le but essentiel dudit enseignement d'algorithmique serait d'apporter un plus sur le plan de la rigueur intellectuelle.

De toutes les manières, toutes ces questions appartiennent en partie à l'avenir, puisque pour le moment, l'informatique et l'algorithmique ne sont pas - ne sont plus - enseignées en tant que telles.

### **...et une raison officieuse ?**

Alors, j'en viens à ce qui constitue, j'en suis convaincu, la vraie raison de la suppression de l'option informatique : une simple logique comptable, budgétaire, n'ayant que peu à voir avec l'intérêt des jeunes, même si c'est derrière celui-ci que, comme toujours, l'on s'est abrité pour la justifier. Maintenir telle quelle l'option informatique posait problème ; elle créait un précédent, un point d'appui, sur lequel allaient s'appuyer de plus en plus fermement ceux qui réclamaient que cet enseignement devienne général et obligatoire. Or, une telle généralisation ne pouvait se faire avec les moyens de fortune, en particulier humains, qui avaient peu ou prou suffi à assurer le fonctionnement de ce qui n'était qu'une simple option. Il aurait signifié la nécessaire création d'un CAPES, et sans doute d'une agrégation, d'informatique. Il aurait impliqué le recrutement massif d'enseignants spécialistes de cette matière. Et l'Education Nationale se serait peut-être, qui sait, retrouvée, comme la Sécurité Sociale, obligée de proposer une grille de salaire spécifique pour les informaticiens, car, pardonnez-moi la métaphore, on n'attire pas les mouches avec du vinaigre.

Je me trompe peut-être. Mais je suis convaincu que c'est là, dans cette volonté d'éviter un engrenage de dépenses supplémentaires, que

gît la raison principale de la suppression de l'option informatique. Les raisonnements pédagogiques de cette suppression ne sont venus qu'à titre de justification ex-post.

### Faux-semblants

Et, au risque de choquer définitivement certains d'entre vous, j'irais même plus loin en disant que l'introduction de l'algorithmique dans les programmes de mathématiques n'a été qu'une concession hypocrite, un lot de consolation que l'on a accordé aux vaincus. Plus exactement, elle n'a été qu'un alibi. L'algorithmique, c'est un peu, toutes proportions gardées, la femme du XIXe siècle bourgeois, celle qu'on n'encensait dans la littérature que pour mieux la mépriser dans la réalité. La référence à l'algorithmique a eu comme rôle essentiel de permettre de répliquer à ceux qui s'indignaient de voir la programmation bannie des lycées : « *Voyez donc, non seulement elle n'en est pas bannie, mais elle a été officiellement introduite dans les programmes !* ».

Tout cela relève, à mon sens, d'une hypocrisie aussi manifeste que navrante.

C'est déjà assez vrai de la présence de l'informatique, ou disons, de l'ordinateur, dans l'enseignement. Sans enseignants spécifiques, parfois sans budget, sans salles ou sans techniciens de maintenance, la place de l'informatique dans la réalité des collèges et des lycées reste bien en deçà de celle qu'elle occupe dans les discours officiels. On ne dispose apparemment pas d'études systématiques et documentées sur l'utilisation de l'informatique dans l'enseignement secondaire. Toutefois, selon le seul chiffre disponible, 10 % des enseignants se serviraient, à l'occasion, de l'informatique en cours. Est-il besoin de

dire que c'est un chiffre extrêmement faible ? Car 10 % des enseignants, cela signifie un nombre encore beaucoup plus bas d'enseignements. Il suffit qu'un enseignant consacre un TP par an dans une de ses classes pour faire partie des 10%. Mais du point de vue des élèves, quel est le chiffre ? Personne n'est en mesure de le dire, mais il est vraisemblablement beaucoup plus bas.

Il y aurait donc beaucoup à dire, et à redire, sur le sort réservé à l'informatique dans l'enseignement secondaire. Mais concernant l'algorithmique, l'hypocrisie, là, dépasse franchement l'imagination.

Ainsi, les enseignants de mathématiques sont-ils censés dispenser un savoir algorithmique. Fort bien. Toutefois, rien, ou presque, ne les prédispose à cela. C'est un domaine qui leur est généralement aussi étranger que peuvent l'être la physique, la biologie, ou même l'anglais. Cela n'empêche évidemment pas qu'il existe des enseignants en mathématiques curieux, autodidactes ou qui ont pu être eux-mêmes formés à l'informatique durant leurs études. Mais il s'agit forcément de cas particuliers, et pas du tout de la règle. Donc, premier point, les enseignants de mathématiques n'ont a priori aucune compétence particulière pour enseigner l'algorithmique, ni aucune vocation spéciale à le faire. Alors, j'ai appris tout récemment que la programmation était en train de devenir une matière obligatoire au CAPES, ce qui m'oblige à nuancer mon propos. Néanmoins, combien d'années, ou de décennies, faudra-t-il attendre pour que tous les enseignants de mathématiques possèdent réellement les compétences algorithmiques qu'ils sont censés transmettre depuis déjà quelques années ?

Deuxième point : comment l'algorithmique est-elle censée s'articuler avec les mathématiques ? C'est une question que les programmes passent pudiquement sous silence, et pour cause. Quelques

phrases générales ne donnent pas des directives claires. En un sens, c'est tant mieux, car à bien y réfléchir, autant je suis plutôt convaincu, de l'intérêt d'enseigner l'algorithmique comme une matière spécifique, à dose raisonnable, autant j'ai tendance à croire que présentée comme une branche des mathématiques, elle risquerait d'être franchement contre-productive.

Je m'explique. Imaginons qu'afin de respecter ce que l'on peut penser être les consignes officielles, un enseignant de mathématiques s'attache à faire de l'algorithmique chaque fois que c'est possible. Cela veut dire qu'à chaque fois qu'il explicitera une règle, un mécanisme, un mode opératoire, il le présentera - ou le fera formuler - de manière formalisée, dans un pseudo-langage. A défaut de faire de l'algorithmique de « niveau 3 » (faire concevoir des traitements) il tenterait de faire de l'algorithmique de « niveau 2 », c'est-à-dire faire formaliser, traduire, des traitements existants, dans le langage des ordinateurs.

Alors, je suis très loin d'être compétent en matière de didactique des mathématiques, et sans doute me corrigerez-vous sur ce point... comme sur d'autres. Mais, sur la base de ma petite expérience d'élève, et sur celle de mes camarades d'école, j'ai toujours eu le sentiment que les élèves qui avaient des difficultés, en mathématiques, étaient précisément ceux qui se raccrochaient le plus énergiquement aux règles, quitte à les appliquer en dépit du bon sens. Faute de comprendre qualitativement les concepts qu'ils étaient en train de manier, ils se réfugiaient dans une vision que j'appellerais juridique des mathématiques : « *Ca, on a le droit de le faire* », « *Ca, on n'a pas le droit* ». Ainsi, privées de leur sens, les règles et les objets mathématiques se mettaient à acquérir dans leur cerveau une existence

autonome, et les exercices de maths ne devenaient que l'application systématique de « recettes de cuisine » apprises par cœur, sans que jamais ils s'interrogent sur la signification de ce qu'ils étaient en train de faire. Et évidemment, systématiquement, ils appliquaient les règles en-dehors de leur domaine de validité, ou à des objets auxquelles elles ne se rapportaient pas.

Alors, insister sur ces règles, les formaliser, privilégier cette approche, ne risquerait-il pas de renforcer ce travers, de mettre l'accent sur le formel plutôt que sur le contenu ? Ne vaudrait-il pas mieux passer du temps à faire comprendre qualitativement ce qu'est un polynôme du second degré, en quoi les racines se représentent graphiquement, sur le lien intime entre représentation algébrique et représentation graphique, plutôt qu'insister sur la formalisation du calcul qui permet de trouver ces racines ? Je n'ai pas la réponse, mais je pense que la question mérite d'être posée.

Par ailleurs, je ne peux que répéter ce que j'ai expliqué tout à l'heure ; si l'algorithmique a parfois besoin de quelques connaissances mathématiques, on peut tout à fait devenir un excellent programmeur en étant un médiocre mathématicien. Pour des raisons qui m'échappent, c'est, dans une certaine mesure, mon cas personnel. Bien que je ne me considère pas du tout comme un champion de la programmation, tout ce que j'ai appris, je l'ai appris sans me faire violence ; la logique de la programmation m'a toujours semblé quelque chose de très naturel et de très accessible. Je n'en dirais pas autant, et de très loin, des mathématiques.

Alors, à identifier mathématiques et algorithmique, on risque d'une part de ne pas aider ceux qui ont des difficultés en mathématiques à mieux y réussir ; d'autre part, on peut détourner de la

programmation des gens qui auraient pu y réussir, mais qui ne se sentent pas des âmes de matheux.

Cela dit, on peut se rassurer quant aux dégâts que risque de causer l'introduction de l'algorithmique dans les programmes de mathématiques, en se disant que ces dégâts resteront presque entièrement virtuels.

Bien conscient de n'avoir formulé dans ses recommandations, ses discours et ses programmes officiels, qu'un vœu pieux n'ayant aucune vocation à être traduit concrètement, le ministère s'abstient sagement de poser la moindre question relative à l'algorithmique dans le seul endroit qui compte, à savoir le baccalauréat. Tout aussi pragmatiques, les manuels scolaires - en tout cas, tous ceux que j'ai feuilletés - jettent eux aussi un voile pudique sur la question, et sans même parler d'y consacrer quelques chapitres, voire quelques paragraphes, ne font même pas mention du terme. Tous les enseignants de mathématiques que j'ai pu interroger m'ont répondu unanimement que l'ampleur des programmes étant ce qu'elle est, une question qui n'apparaît jamais à l'examen est forcément évacuée d'un enseignement déjà trop court. Et la boucle est ainsi bouclée.

Alors, que reste-t-il, en réalité, aujourd'hui pour l'algorithmique en lycée ? Selon le mot d'un enseignant-chercheur travaillant dans un IREM parisien avec lequel j'ai discuté, il reste une poignée de militants qui, de ça, de là, s'efforcent tant bien que mal de ménager une petite place pour cet enseignement et qui tentent tant bien que mal de le faire vivre. Et il semblerait bien que de tels militants existent ici, à la Réunion. Mais c'est très loin d'être le cas partout ; et j'irais presque jusqu'à dire que vous formez une exception.

## **En guise de conclusion**

Pour terminer cet exposé, je voudrais donc, comme il est d'usage, résumer les points qui me paraissent importants et tirer quelques perspectives.

Mathématiques, informatique et algorithmique sont des domaines qui possèdent des atomes crochus et des zones de recoupement, mais qui sont néanmoins distincts. Non seulement les mathématiques ne souffriraient pas de l'existence d'un enseignement spécifiquement informatique, mais, dans une large mesure, elles en bénéficieraient.

Cet enseignement de l'informatique est une nécessité qu'il me paraît difficile de nier, du moins si l'on discute sur le seul terrain pédagogique. Il devrait comporter une part de connaissances techniques sur les ordinateurs, sur la manière dont ils codent les informations, et dont ils fonctionnent. Il devrait également aborder les principaux logiciels fondamentaux (traitement de texte, tableur et, on l'oublie trop souvent, bases de données). Il pourrait aussi, d'une manière qui resterait à préciser, initier les élèves à l'algorithmique et aux fondements de la programmation. En tout état de cause, il devrait être sous la responsabilité d'enseignants qualifiés, recrutés sur cette base disciplinaire, et ne pas reposer sur le dévouement et la débrouillardise individuelle de collègues venant d'autres matières. En clair, il faudrait s'orienter vers la création d'un CAPES, voire d'une agrégation, d'informatique. Un tel enseignement représenterait une plus-value pour les élèves, et permettrait y compris d'aborder sous un autre angle certaines notions étudiées dans d'autres cours. Encore une fois, on pourrait, et on devrait, discuter longuement des détails d'un tel enseignement, de son poids horaire et de son contenu. Mais vingt-cinq

ans - une génération ! - après l'arrivée des premiers PC dans les foyers, son principe même devrait être acquis depuis longtemps.

Malheureusement, ce n'est pas le cas, et cela risque fort de ne pas l'être dans un avenir prévisible. Alors, plaider pour que la situation change me semble nécessaire. Mais cela ne nous dispense pas de savoir quoi faire dans la situation actuelle.

Celle-ci est marquée tout à la fois par la volonté affichée des autorités de ne fermer en théorie aucune porte, et d'encourager officiellement la diffusion de l'informatique, voire de l'algorithmique. Et en pratique, la réalisation de cette volonté affichée est laissée aux bons soins du volontariat, pour ne pas dire de la débrouillardise, des enseignants, essentiellement de mathématiques.

Alors, dans la mesure du possible, je pense qu'il faut se saisir des opportunités, même faibles, qui existent, pour développer l'utilisation des ordinateurs. Je pense aussi qu'il est bénéfique de développer l'enseignement de l'algorithmique, même s'il ne peut s'agir en tout état de cause que d'une initiation, et même si des initiatives comme la vôtre, faute d'un véritable cadre national, ne peuvent constituer que des actes isolés. Toutefois, ces initiations à l'algorithmique doivent être présentées, aux élèves comme aux enseignants, comme l'introduction à une discipline spécifique, distincte des mathématiques. Comme je l'ai dit, on ne fait pas, on ne doit pas faire, de l'algorithmique pour devenir meilleur en mathématiques. C'est un contresens, et, je le crois, un contresens potentiellement dangereux. En revanche, on peut faire de l'algorithmique pour aborder de nouveaux problèmes et de nouvelles manières de raisonner face à ces problèmes.

Donc, pour finir cette fois-ci pour de bon, faisons, et faisons faire, de l'algorithmique. Faisons faire aux élèves non seulement de la formalisation d'algorithmes, mais aussi de la conception. Formons non seulement des traducteurs, mais aussi des romanciers. Mais sans perdre de vue que la bonne volonté, locale et à la marge, ne remplacera jamais une volonté centrale, coordonnée, et disons-le, politique, de promouvoir cet enseignement, non seulement en paroles, mais dans les faits.

Christophe DARMANGEAT

Université Paris 7

Le 15-04-2005