

Parlons d'algorithmes

Jean-François Canet(*)

Les quelques lignes qui suivent sont inspirées par une longue carrière au cours de laquelle algorithmes, programmes, logiciels de mathématiques ont été des objets usuels de mon environnement professionnel. Les réflexions que j'essaie de transcrire ont été nourries des échanges avec les élèves, les professeurs, les inspecteurs, les développeurs. Elles essaient de refléter un certain aspect de ce qui se passe dans l'enseignement, mais ne sauraient engager personne d'autre que leur auteur qui s'exprime ici à titre strictement personnel.

1. La surprise de la nouveauté

Nouveau ou ancien

Lors de la mise en consultation de l'actuel programme de seconde et au cours des très nombreux échanges qui ont suivi (et suivent encore) sa mise en œuvre, il a été fréquent d'entendre parler de la nouveauté de la présence de l'algorithmique dans ce programme. Si l'on fait quelques pas de côté pour regarder les objets qui régissent ou accompagnent l'enseignement, cette appréciation de « nouveauté » peut surprendre.

Quelques exemples :

Les **programmes** évoquent depuis fort longtemps l'aspect algorithmique de certaines parties de notre enseignement, juste une citation pour illustrer ceci :

Circulaire du 6 avril 1971 : l'enseignement du calcul dans les établissements du second degré.

[...] Les élèves apprendront à organiser un calcul et à en dresser l'organigramme ; ils établiront par exemple la feuille de calcul des valeurs d'une fonction numérique f donnée, en blanc, avant tout choix de valeurs de la variable x ; c'est déjà l'élaboration d'un programme, une voie ouverte vers l'informatique [...].

Il faut bien situer ce passage dans son contexte historique : en 1971 aucun professeur n'imaginait pouvoir se servir en classe d'une calculatrice électronique, encore moins d'un tableur !

Lorsqu'on relit les programmes successifs⁽¹⁾ on ne peut que constater que les recommandations pour prendre en compte cet aspect algorithmique de notre discipline ont été une constante tout au long des quarante dernières années (si bien que tous les professeurs les ont toujours connues !). Mais s'agit-il seulement de recommandations générales ? De vrais « objets » d'enseignement typiquement algorithmiques ont aussi figuré dans divers programmes (la méthode du pivot de Gauss, la méthode d'Euler, de

(*) IA-IPR de Mathématiques, jean-francois.canet@ac-aix-marseille.fr

(1) Quelques extraits ont été rassemblés et mis à la disposition des enseignants sur le serveur de l'académie d'AIX :

<http://www.maths.ac-aix-marseille.fr/spip/IMG/pdf/Reflexions-2.pdf>.

façon plus marginale, algorithme de coloriage des graphes, recherche de plus courte chaîne, ...).

Les diverses publications qui accompagnent la mise en œuvre de l'enseignement se sont largement fait l'écho de cette approche, une recherche sur Publimath le montre aisément :



377 fiches trouvées | Réponses 361 à 377

361 1984 Pour une mathématique vivante en seconde.

On peut donc être surpris par cette impression de nouveauté, visiblement sincère, majoritaire parmi les enseignants.

Tentons quelques pistes d'explication

« BAC »

La stabilité d'un objet d'enseignement se mesure souvent à la fréquence de sa présence dans les dispositifs d'évaluation nationaux. Quels sont les sujets de Bac qui ont vraiment demandé d'explicitier l'aspect algorithmique ? Ceci pose un problème qui dépasse très largement le cadre de cet article. Il faut cependant bien préciser qu'il ne s'agit pas là d'une critique de la façon de procéder des enseignants⁽²⁾ (et donc des auteurs de sujets) mais d'un regard sur le mode de fonctionnement de notre système.

« ique »

Nombreux sont les professeurs qui font une place explicite dans leur enseignement, lorsque cela a un sens, à des algorithmes. Ils n'ont pas pour autant l'impression d'enseigner de l'**algorithmique**. En ce sens il faut sans doute revenir sur les attentes du programme de seconde qui peuvent être lues, sur ce sujet, de façon trop exigeantes. Rappelons le texte : « *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* ».

Si l'on garde, plus ou moins consciemment, l'approche qui consiste à penser qu'enseigner une notion exige⁽³⁾ d'avoir des définitions, des propriétés, et des théorèmes, il est normal de penser que l'algorithmique est une nouveauté que l'on ne « sait pas » enseigner.

En deux mots

Il me semble que l'essentiel est bien que dans les classes se racontent des

(2) Ces mêmes enseignants ont montré, dans un domaine voisin, leur capacité à se mobiliser pour mettre en place une épreuve expérimentale en TS.

(3) Et qu'en plus cette condition nécessaire n'est bien sûr pas suffisante !

« histoires d'algorithmes⁽⁴⁾ », qui permettent à chacun de dépasser l'aspect « magique » d'un traitement automatisé, pour percevoir (à un niveau plus ou moins détaillé) l'organisation qu'a exigé ce traitement et pour inciter à en rechercher de nouveaux. En ce sens les objets sur lesquels vont porter cet apprentissage font bien partie du patrimoine culturel des enseignants et n'ont rien de nouveau. Le regard que l'enseignement du professeur va permettre à l'élève de porter sur ces objets est certainement beaucoup moins habituel et en ce sens on comprend bien que ce programme incite à aborder une nouvelle étape.

2. Mais pourquoi donc ?

Dans son histoire d'écolier puis de collégien, l'élève de seconde a déjà rencontré de nombreux algorithmes (cela fait longtemps qu'il a appris à poser des opérations), a été amené à les utiliser pour obtenir les résultats escomptés, sans que cela exige pour autant de s'arrêter pour analyser le « comment ça marche ». Et même, n'est-ce pas la vertu première d'un traitement automatisé que de se faire « oublier » pour mieux « profiter » du résultat qu'il a produit ?

Découvrir la vérité des faits mathématiques

Si l'on prend l'exemple de la multiplication posée, le programme de l'école primaire insiste sur la nécessité d'acquérir le mécanisme mais précise aussi « *L'acquisition des mécanismes en mathématiques est toujours associée à une intelligence de leur signification* »⁽⁵⁾.

Un des objectifs essentiels de l'enseignement des mathématiques est d'établir la *vérité des faits mathématiques* et « *L'enseignement des mathématiques conduit à goûter le plaisir de découvrir par soi-même cette vérité, établie rationnellement et non sur un argument d'autorité, et à la respecter. Faire des mathématiques, c'est se les approprier par l'imagination, la recherche, le tâtonnement et la résolution de problèmes, dans la rigueur de la logique et le plaisir de la découverte* »⁽⁶⁾. Pourquoi les faits mathématiques qui relèvent d'une démarche algorithmique échapperaient-ils à cet objectif ambitieux ?

La curiosité

Cette curiosité, dont on disait naguère qu'elle poussait l'enfant à aller voir à l'intérieur d'un réveil comment naissaient les heures, est un des moteurs essentiels de notre enseignement dont on ne cesse de rappeler qu'il s'appuie sur des problèmes à résoudre (donc des questions que l'on se pose). La compréhension du monde d'aujourd'hui, dans lequel la « naissance des heures » risque fort d'être (en partie) le résultat de la mise en œuvre d'un algorithme, ne peut certainement pas faire l'impasse sur cette dimension de la connaissance scientifique.

(4) Pour reprendre le titre de l'ouvrage remarquable (mais semble-t-il épuisé) publié chez Belin en 1994.

(5) BO hors-série n° 3 du 19 juin 2008.

(6) Programmes du collège BO spécial n° 6 du 28 Août 2008.

Le raisonnement

La compréhension d'algorithmes donnés, la production d'algorithmes aptes à produire un résultat prévu, sont des occasions de mettre en évidence une capacité réelle à raisonner qui *peut* emprunter d'autres voies que celle de la rédaction d'une démonstration. La question difficile de l'évaluation des compétences de raisonnement d'un élève doit, pour trouver des réponses satisfaisantes, reposer sur une diversité de situation. Le travail algorithmique vient donc en la matière enrichir le champ des possibles.

Une inscription dans l'ensemble des cursus

Tous les programmes qui ont été publiés ces dernières années font une large place à l'utilisation d'outils informatiques (en particulier le tableur). Ces outils mettent en jeu de nombreux algorithmes (soit de façon interne à l'outil, soit au travers des « commandes » que l'utilisateur a prévues). Prendre le temps de réfléchir à certains de ces algorithmes, dégager des exemples rencontrés quelques grands principes d'organisation est donc naturel dans une classe de détermination et prépare efficacement à toutes les poursuites d'études. Par exemple tous les programmes de première vont faire appel à des simulations d'expériences aléatoires. Un travail algorithmique suffisant en seconde (comment simuler telle expérience, comment dénombrer les résultats dans une simulation, ...) va donc efficacement préparer le travail qui suivra.

3. Et comment ?

Les quelques lignes qui suivent n'ont bien sûr pas l'ambition de répondre à ce « comment ? ». Elles espèrent seulement susciter réflexions et débats à partir d'un ou deux exemples.

Quelques idées

Le programme souligne que les problèmes posés dans cette partie peuvent être en rapport avec la vie courante. C'est dans ce registre là que je choisis le premier exemple :

Léo et Léa

Demandons à la fin d'un cours à nos élèves de faire, pour la prochaine fois, le travail suivant : Choisissez un texte dont vous possédez une version numérisée ; à l'aide de votre traitement de texte remplacez tous les « o » par des « a » et tous les « a » par des « o ». Notez ce que vous avez fait et ce qui s'est passé.

Le bilan qui sera fait au cours suivant peut être très fructueux ! Entre ceux qui auront obtenu un texte qu'avec des « a » ou qu'avec des « o » et ceux qui auront eu le résultat attendu il faudra s'expliquer ! D'où le besoin de préciser l'algorithme employé.

Ce thème de la recherche d'un objet dans une collection est riche et nous savons bien que les algorithmes les plus performants font l'objet d'un travail mathématique

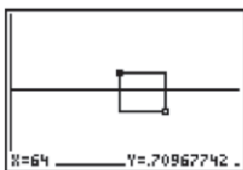
important et difficile. Sans rentrer dans les détails des algorithmes complexes, on peut faire prendre conscience que ce n'est pas la même chose de rechercher un mot dans un texte et une entrée dans un dictionnaire⁽⁷⁾, on rencontre ainsi la notion de structuration des données et celle d'indexation.

Une drôle de courbe

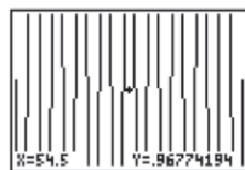
Les trois copies d'écran ci-dessous ont été obtenues sur une calculatrice :



Une « courbe »⁽⁸⁾



On « zoome »



et on obtient

Pour comprendre ce qui se passe, il est nécessaire (mais pas suffisant) de comprendre ce que fait la commande de tracé de la calculatrice. Ceci peut se faire en écrivant explicitement un algorithme de tracé.

Placer les points (ou les segments) dans la fenêtre repose la question de la définition de la « zone de tracé ». Si celle-ci est choisie automatiquement (comme c'est souvent le cas lors de tracés obtenus à l'aide de tableurs), la question qui en découle immédiatement est celle de la recherche d'extrema dans une liste de nombres.

On voit donc qu'une situation volontairement surprenante peut être le « prétexte » (ou la motivation) à la recherche d'algorithmes.

Les algorithmes de tracés ne sont pas tous élémentaires, certains essaient de prendre en compte la courbure pour optimiser le tracé. Les outils à disposition des élèves peuvent être l'occasion de se poser de nouvelles questions (voir en annexe deux tracés évoqués dans la notice du logiciel Maxima®).

En bref

À travers ces deux exemples, j'ai voulu montrer que dans l'environnement usuel de la classe et du cours de math, on trouvait, sans peine, la matière à la réflexion. Il s'agit surtout d'accepter de se poser des questions à propos de ces situations ordinaires, pour arriver à « dégager quelques grands principes ». En copiant ce qui est dit à propos de la logique et des notations, je suis convaincu que les notions d'algorithmiques rencontrées en seconde « sont à considérer comme des conquêtes de l'enseignement et non comme des points de départ » et doivent être très progressivement dégagées des échanges qui accompagnent les problèmes mis en débat dans la classe : inutile de commencer par des définitions des mots « algorithme », « variables », « boucles », etc.

(7) Je pense ici aussi bien au dictionnaire usuel qu'au dictionnaire T9 (texto sur 9 touches) qui m'aide à écrire mes messages brefs !

(8) On reconnaît sans peine le tracé de $y = 1 + \sin(\pi x)$ sur l'intervalle $[0, 94]$!

Avec quels outils ?

Le postulat énoncé ci-dessus a pour corollaire que le premier outil utilisé est la langue naturelle (voire la langue des mains ou des pieds pour reprendre le clin d'œil du document ressource). Il est cependant certain que l'on aura naturellement envie de voir « tourner » des algorithmes de façon automatique et que cela va inciter à utiliser des outils. Il faut avoir conscience que tout outil transportera avec lui ses contraintes. Dans certains cas, les contraintes transportées par l'outil sont source de travail mathématique (comment une machine peut-elle tester l'égalité de deux nombres⁽⁹⁾ ?) ; dans d'autres, ces contraintes peuvent faire écran et éventuellement détourner élèves et enseignant de l'objectif premier.

Par exemple, nous souhaitons faire travailler nos élèves sur un algorithme qui va commencer par demander un nombre N (pour calculer une valeur approchée de sa racine carrée par exemple). La façon de coder cette saisie va dépendre du système utilisé : faudra-t-il afficher le message puis lire la saisie ou bien utiliser une seule commande ? La syntaxe de la saisie fera-t-elle appel à une affectation explicite ($N = \dots$) ou non, la commande de saisie utilisera-t-elle des parenthèses ou non ? La chaîne de caractères constante sera-t-elle délimitée par de simples apostrophes ou des guillemets ? Plus complexe, la saisie va-t-elle retourner un nombre ou une chaîne ? Enfin il vous faudra bien gérer les messages d'erreurs qui ne manqueront pas d'apparaître sur certains écrans⁽¹⁰⁾. Il est donc important que l'enseignant mesure correctement le poids de ces contraintes « externes ». Il peut être raisonnable dans un premier temps de se limiter à des outils déjà familiers (tableur, calculatrice) ou à des outils minimisant à l'extrême l'apprentissage propre (superficiellement l'apprentissage de la syntaxe). Il faut bien être conscient que la simplicité d'usage entraîne inéluctablement une limitation des possibilités. Je crois cependant prudent de viser à cette simplicité.

Il faut aussi être conscient que les outils à disposition pour faire des mathématiques disposent d'algorithmes « embarqués » de plus en plus puissants. Les boucles et les tests sont souvent intégrés dans des fonctions de haut niveau comme le montre l'exemple suivant. Problème farfelu : « trouver la somme des nombres compris entre l'année de la naissance de Cardan et l'année de la mort de Gödel qui sont des carrés parfaits ». Voici par exemple comment obtenir la réponse avec Derive⁽¹¹⁾ ? :

```
Σ(SELECT(carre_parfait(n), n, VECTOR(i, i, 1501, 1978))
```

Ce problème peut se traiter dans le même esprit avec un tableur à l'aide des filtres

(9) Cette formulation simple cache des problèmes complexes suivant ce que l'on met derrière le mot nombre et aussi la façon dont ces « nombres » sont représentés.

(10) Par exemple :

```
Traceback (most recent call last):
```

```
File "<pyshell#1>", line 1, in <module>
```

```
Input('Le nombre',N)
```

```
NameError : name 'Input' is not defined
```

(11) La fonction booléenne `carre_parfait` parle d'elle-même ; ce peut être $\sqrt{n} = \text{Floor}(\sqrt{n})$.

mis à disposition et de la fonction SOUS.TOTAL (cf. annexe). Par conséquent un élève qui propose en réponse au problème farfelu « Ben je prends toutes les années, puis je sélectionne celles qui sont des carrés parfaits et je les ajoute » a bien fait une analyse « opérationnelle » du problème et il n'y a pas, *a priori*, à lui demander de « détailler » plus.

Le fait que les développeurs proposent de plus en plus de ces potentialités évoluées⁽¹²⁾ montre que les structures élémentaires de boucles et de test seront d'un emploi de plus en plus limité. Tout comme l'arrivée des structures de boucles et de test a rendu rare l'emploi des redirections (Goto, Skip, ...). Ceci ne signifie pas que les concepts de test et d'itérations ne restent pas essentiels, mais la façon de les présenter et leur formalisation dans un langage demande une grande souplesse. Ces fonctions évoluées peuvent, du reste, être « décortiquées » dans la classe pour que le travail qu'elles accomplissent soit bien compris. C'est souvent ce qui se passe avec la fonction NB.SI des tableurs.

Et quelle évaluation ?

Cette question mériterait un article à elle toute seule. Je vais me limiter à affirmer ici quelques convictions (à discuter) qui sont le fruit de la double expérience de l'option informatique des lycées (il y a une vingtaine d'années) et de l'épreuve expérimentale encore en place.

L'évaluation écrite en temps limité est assez inadaptée pour apprécier les compétences que les élèves développent dans cette partie de l'enseignement. Elle peine à rendre compte des initiatives prises. La question de la présence ou de l'interdiction d'outil pendant cet écrit conduit à des postures difficilement tenables pour l'institution : interdire toute calculatrice dans une épreuve appelée à tester de l'algorithmique paraît choquant, gérer la diversité des fonctionnalités offertes (la fonction « select » pour l'exemple cité) est un vrai casse-tête qui finit par poser des problèmes d'équité.

Plus encore que la relative inadaptation de cette modalité, il faut en redouter les effets pervers. Si l'on y prend garde, le travail écrit en temps limité sur des connaissances qui restent, à ce niveau d'enseignement, peu théorisées peut inciter à une « diafoirisation » des savoirs⁽¹³⁾.

Cette partie du programme (au long des trois années de lycée) gagne donc à être évaluée « en situation » sous diverses formes. En TP, l'évaluation peut (doit ?) porter à la fois sur le travail « en train de se faire » et sur les comptes-rendus que les élèves seront amenés à produire. L'observation « ordinaire » du travail de la classe permet aussi d'apprécier les premiers pas des élèves et renseigne sur des compétences déjà manifestes ; la présentation de brefs exposés permet d'apprécier la capacité de l'élève (ou d'un groupe d'élèves) à rechercher une information pertinente et à l'exposer de

(12) Il y aurait beaucoup d'autres « potentialités évoluées » à évoquer ; en particulier comment ne pas citer l'approche récursive de certains problèmes ?

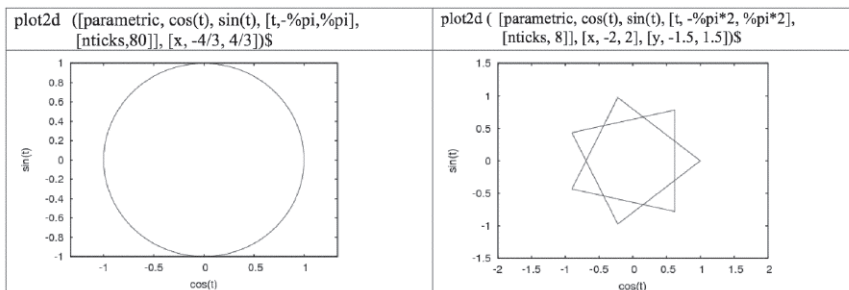
(13) Un des professeurs de l'option informatique avait coutume de dire, en riant, que la réponse à la première question de l'écrit du baccalauréat était « un tableau » et ... c'était pratiquement vrai, la question étant souvent formulée « Quelle structure de données utiliseriez-vous pour représenter ... ».

façon claire. Nous ne manquons donc pas de moyens d'évaluations.

Il faut souhaiter que la recommandation du programme sur l'évaluation : « *Les élèves sont évalués en fonction des capacités attendues et selon des modes variés : travaux écrits, rédaction de travaux de recherche, comptes-rendus de travaux pratiques. L'évaluation doit être en phase avec les objectifs de formation rappelés au début de cette introduction.* » devienne une réalité dans les pratiques de classe.

ANNEXE

1. Exemple provenant de l'aide en ligne du logiciel Maxima®



2. Mise en œuvre du « problème farfelu » avec un tableur

La colonne A contient les années considérées (de 1501 à 1978). Ceci est obtenu à l'aide de la « recopie incrémentée » et ne nécessite pas d'explicitier des « formules ». La colonne B contient le test pour savoir si le nombre correspondant en colonne A est un carré parfait. Dans la cellule C1 la fonction Sous.Total applique l'opération qui est indiquée par son premier argument (ici 9 : la somme) à son second argument (ici la colonne A) en respectant les filtres qui seront appliqués.

	A	B	C
1			=SOUS.TOTAL(9;A:A)
2			
3	1501	=RACINE(A3)=ENT(RACINE(A3))	
4	1502	=RACINE(A4)=ENT(RACINE(A4))	
479	1977	=RACINE(A479)=ENT(RACINE(A479))	
480	1978	=RACINE(A480)=ENT(RACINE(A480))	
481			

En filtrant pour ne conserver que les valeurs vraies de la colonne B (et donc les carrés parfaits de la colonne A) on lit en C1 le résultat du problème.

	A	B	C
1			10351
2			
23	1521	VRAI	
102	1600	VRAI	
183	1681	VRAI	
266	1764	VRAI	
351	1849	VRAI	
438	1936	VRAI	
481			