

# Changer de rythme avec l'algo

Mieux comprendre le monde qui nous entoure pour être de ceux qui le construisent et non de ceux qui le subissent : qu'y a-t-il derrière ce message fort ? Quelle ambition, quels objectifs pour nos élèves ?

Nous n'aurons pas la prétention de répondre globalement, dans cet article, à ces questions. Par contre, pour ce qui nous intéresse tous, en tant que professeurs de mathématiques, nous allons tenter d'apporter quelques éléments de réflexion en ce qui concerne l'apport de l'algorithmique dans les nouveaux programmes de seconde.

Un programme est un écrit politique, au sens large, qui est le résultat d'un consensus d'élites, pour répondre aux besoins de la société. Or, l'évolution des technologies a déplacé le centre de gravité des savoir-faire et des connaissances.

Par exemple : il n'y a pas si longtemps, réparer soi-même son automobile était envisageable avec un peu de courage et la lecture d'une revue technique. Aujourd'hui, la plupart des réparations ne peuvent être effectuées que par un spécialiste équipé d'une connexion informatique appropriée.

« Programmes et algorithmes sont partout, certains très simples (digicode, répondeur téléphonique), d'autres plus complexes ou invisibles pour l'utilisateur (codage et décodage de l'information numérique, cartes bancaires, réseaux téléphoniques...) » d'après rapport commission Kahane. Décembre 2000

La société actuelle réclame des personnes adaptées à cette évolution. Et c'est pour répondre à ce besoin que la formation des acteurs de demain contient en mathématiques, une part d'algorithmique.

## Comment l'enseignement des mathématiques peut-il répondre à ces besoins ?

Nos élèves d'aujourd'hui, sont ces acteurs de demain, il semble donc important de les former, d'une part pour qu'ils s'insèrent de la meilleure façon possible dans cette société nouvelle, grande consommatrice d'informatique et d'autre part pour qu'ils acquièrent tous, la culture générale pour s'adapter aux évolutions du monde qui les entoure.

C'est ainsi que dans le nouveau programme de mathématiques de seconde, une place est faite à l'algorithmique et à la programmation.

## Des difficultés de mise en place :

Beaucoup d'étudiants en mathématiques ont, dans la formation théorique, appris à utiliser un langage informatique et à programmer. Paradoxalement, on pouvait constater que cette formation n'était que rarement réinvestie dans l'enseignement secondaire. Certains collègues, ne se sentent pas compétents pour enseigner l'informatique, ce terme étant, dans les faits, polysémique. En effet, pour la plupart des enseignants de mathématiques, **informatique** est synonyme de programmation, tandis que, pour le sens commun, l'informatique est simplement l'utilisation d'un ordinateur : on confond la science informatique et la technologie. Cette ambiguïté sur le mot informatique génère une appréhension envers l'enseignement de l'algorithmique et de la programmation en classe, appréhension d'autant plus vive que les enseignants n'ont, pour la plupart, ni modèle, ni référence dans leur propre vécu scolaire.

Les nouvelles demandes institutionnelles lèvent cette ambiguïté : il faut dorénavant, utiliser non seulement l'informatique comme outil mais en plus comme interface à l'algorithmique.

Mais cet enseignement (comme toute nouveauté) peut effrayer et ne se fera pas sans difficultés. Nous en avons relevé quelques unes :

### 1. Maîtrise de l'informatique et des instruments

La maîtrise de l'informatique (internet, logiciels, langage de programmation...) peut paraître indispensable à certains collègues qui ont l'habitude de tout contrôler. Il peut en effet paraître déroutant que les élèves aient des compétences que leur professeur n'a pas nécessairement. Savoir écrire un algorithme, n'implique pas de savoir programmer dans un langage donné. Nous ne sommes pas des techniciens de l'informatique.

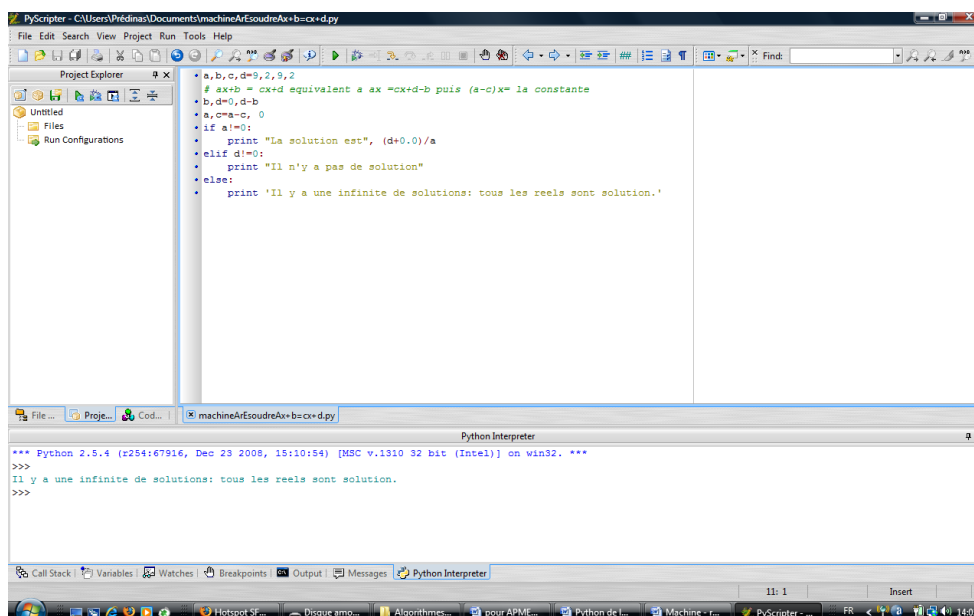
### 2. Difficulté de faire comprendre aux élèves ce qu'est un logiciel

Beaucoup d'élèves utilisent un logiciel en ignorant tout de sa conception. (Les programmes des jeux vidéo ne sont jamais visibles)

### 3. Intégration de la programmation dans les mathématiques des élèves :

Il peut sembler, à certains collègues, que la rigueur syntaxique soit une contrainte supplémentaire à l'apprentissage des élèves, sans apport mathématiques. Par exemples :

- On peut penser que « taper une formule dans Excel », ce n'est pas nécessairement faire de l'algèbre. Le cadre algébrique d'Excel, n'est pas exactement le cadre algébrique mathématique. Mais le va et vient entre les deux modes de travail, peut faciliter l'apprentissage des élèves : (Pour taper une formule de calcul de pourcentages, on peut partir de la formule mathématique et voir qu'il faut remplacer « x » par le nom de la cellule et inversement, si la « formule Excel » est donnée dans l'énoncé, l'élève peut retrouver la formule mathématique et retrouver le calcul de pourcentages) Si les élèves délèguent la partie algébrique au logiciel et réalisent un exemple générique ; ils font alors de l'arithmétique.
- La résolution d'une équation du premier degré, lorsqu'elle est programmée, peut donner l'impression de supprimer les mathématiques en délèguant tout l'aspect calculatoire à la machine ; alors qu'en se débarrassant d'un aspect que les élèves sont sensés maîtriser, on peut aborder des notions plus complexes telles que la différenciation des indéterminées, variables et paramètres et peut permettre un travail sur la nature des nombres.



```
a,b,c,d=0,2,9,2
# ax+b = cx+d equivalent a x =cx+d-b puis (a-c)x= la constante
b,d=0,d-b
a,c=a-c, 0
if a!=0:
    print "La solution est", (d+0.0)/a
elif d!=0:
    print "Il n'y a pas de solution"
else:
    print 'Il y a une infinite de solutions: tous les reels sont solution.'
```

```
*** Python 2.5.4 (r254:67916, Dec 23 2008, 15:10:54) [MSC v.1310 32 bit (Intel)] on win32. ***
>>>
Il y a une infinite de solutions: tous les reels sont solution.
>>>
```

L'utilisation de l'algorithmique et des mathématiques des élèves, ne sont pas deux enseignements parallèles mais deux modes de travail différents et imbriqués.

#### 4. Problèmes de gestion de classe :

La gestion des élèves lors de séances de programmation, s'apparente davantage au travail en petits groupes plutôt qu'à celle d'une classe entière :

- L'enrôlement est différent
- Les élèves avancent à des rythmes différents et le professeur n'avance plus de façon uniforme.

Les enseignants qui ne sont pas habitués à ce type de gestion de classe peuvent être rapidement déstabilisés. Les difficultés seront encore plus importantes pour les professeurs très « cadrants ».

#### 5. Appropriation personnalisée des problèmes par rapport aux langages :

Les enseignants n'ont pas l'expérience pour anticiper les questions des élèves.

La structure du programme imaginée par l'élève ne sera pas toujours explicite ni conforme à celle de l'enseignant. En cas d'erreur ou de difficulté de réalisation, celui-ci pourra d'autant moins aider l'apprenant.

Les problèmes étant très ouverts, l'enseignant doit veiller à ne pas trop guider l'élève afin de ne pas lui faire perdre la nature du problème à résoudre et sa propre manière de l'aborder. Trop vouloir aider risque de provoquer un blocage.

Nous donnons des propositions pour lever ces obstacles dans la formation algorithmique et programmation de l'académie d'AMIENS.

De plus, afin de se concentrer sur l'essentiel (choix des activités, intérêt didactique, moyens pédagogiques, etc.) il semble nécessaire de régler rapidement le problème du choix du langage.

Un langage commun au sein d'un même établissement voire d'une même académie pourrait faciliter la mutualisation des expériences et la continuité pédagogique pour les élèves.

Nous avons choisi Python pour les raisons suivantes : il est gratuit, libre, simple d'utilisation, dérivé des grands langages actuels, multiplateforme, avec une aide intégrée et un débogage explicite (signalement et repérage d'erreurs), proposé par l'inspection générale et préconisé pour l'agrégation interne.....

Tout autre choix de langage aurait été possible avec d'autres hypothèses de formation et des impacts sans doute différents sur les apprentissages des élèves.

Si à la lecture de tous ces obstacles, vous êtes démotivés en ayant à l'esprit l'élève au fond à droite près de la fenêtre qui n'a pas encore commandé le nouveau manuel et qui digère difficilement le thon-mayo-frite de la cantine, nous espérons raviver l'enthousiasme de vos débuts par l'évocation de bénéfices possibles pour les mathématiques.

### **Bénéfices possibles de l'utilisation de la programmation et de l'algorithmique en classe :**

- Activité des élèves :

Faire un algorithme et programmer est un changement de mode de travail qui peut remotiver certains élèves, de même que l'attrait de l'écran. Ils sont plus actifs, mutualisent davantage leurs connaissances et leurs compétences, changent de point de vue sur certains contenus mathématiques ce qui permet de valoriser des élèves, en difficulté dans l'enseignement papier-crayon.

L'algorithmique et la programmation trouvent leur place dans tous les champs d'application du programme de seconde

- Vérification des calculs

Un programme peut permettre la suppression des aspects routiniers de certains calculs numériques

- En géométrie

La tortue Logo permet l'apport du mouvement et donne un aspect ludique et plus compréhensible à la géométrie.

- En statistiques

La programmation permet de travailler sur un échantillon plus grand, des simulations plus faciles d'un point de vue de l'organisation pratique, des changements de registres plus naturels, des comptages et traitements des données moins fastidieux, d'immédiates fluctuations d'échantillonnages.

- En analyse

L'obtention de la courbe d'une fonction affine par morceaux, de l'encadrement et des valeurs approchées de certaines solutions peut être facilement automatisée (par exemple par dichotomie)

- En algèbre

La gestion des cas particuliers (exemple : résoudre une équation  $ax + b = cx + d$  dans le cas où  $a = 0$ ) est facilitée.

- Dans tous les domaines

- o Rigueur et modélisation :

Un programme est une modélisation qui fonctionne avec les contraintes qu'on lui a données, ce qui permet de voir si ce produit de notre pensée est conforme à nos attentes. La rigueur de la modélisation (boucle mal imbriquée) est très formatrice.

- o Démonstration :

Des contre-exemples réalisables pratiquement sans limite de calcul

La programmation fait à présent partie de la panoplie du mathématicien et peut permettre de vérifier des conjectures (conjecture d'Euler : (1769 infirmée en 1966). Pour tout entier  $n > 2$ , la somme de  $n-1$  puissances  $n^{\text{ièmes}}$  n'est pas une puissance  $n^{\text{ième}}$ ), de démontrer des théorèmes (le théorème des quatre couleurs démontré grâce à la liste exhaustive et au comptage de tous les cas(1976)). Elle peut également, être un bon outil de contrôle et de correction de l'erreur. Elle peut constituer une validation externe au professeur, favoriser l'autonomie des élèves, faire réfléchir sur les objets mathématiques grâce aux messages d'erreurs et faire progresser les élèves dans leur apprentissage. Pour amener nos élèves à être de ceux qui construisent le monde, il semble nécessaire, de modifier notre point de vue et d'utiliser l'informatique non seulement comme un outil qui agit à notre place, mais comme un instrument qui répond à nos demandes, sans toutefois tomber dans le piège de programmer uniquement pour programmer . Dans ce cas, les mathématiques seraient alors quasiment absentes des activités et des apprentissages des élèves, ce qu'aucun professeur de mathématiques ne semble souhaiter !

```

PyScripter - C:\Documents and Settings\AgneS\Mes documents\Python\mon projet\src\con
File Edit Search View Project Run Tools Help
• L=[]
• for i in range(1,5):
•     L.append(i**5)
• d,i=0,5
• while d==0:
•     L.append(i**5)
•     for j in range(len(L)-1):
•         for k in range(j,len(L)-1):
•             for l in range(k,len(L)-1):
•                 for m in range(l,len(L)-1):
•                     if L[len(L)-1]-L[j]-L[k]-L[l]==L[m]:
•                         print j+1,k+1,l+1,m+1,len(L)
•                         print L[j], L[k],L[l], L[m], L[len(L)-1]
•                         d=1
•                         break
•     i=i+1
•     print i
n=5.py
143
144
27 84 110 133 144
14348907 4182119424 16105100000 41615795893 61917364224

```

### **Bibliographie :**

Kahane J.P., 2002, *Enseignement des sciences mathématiques : Commission de réflexion sur l'enseignement des mathématiques : Rapport au ministre de l'éducation nationale* O.Jacob/Centre National de Documentation Pédagogique, Paris 2002 ; 284 p.; 19 cm ; ISBN: 2-7381-1138-6 2-240-008326

<http://smf.emath.fr/Enseignement/CommissionKahane/>

Chabert J.L.,Barbin E .,Guillemot M., Michel-Pajus A.,Borowczik J ; Djebbar A.,Martzloff J.C., *Histoire d'algorithmes, du caillou à la puce*, Regard de la science, Belin

[http://www.editions-belin.com/ewb\\_pages/f/fiche-article-histoire-d-algorithmes-6513.php](http://www.editions-belin.com/ewb_pages/f/fiche-article-histoire-d-algorithmes-6513.php)

23 juillet 2009, *Programme mathématiques classe de seconde*, BO n°30, ministère de l'éducation nationale

<http://www.education.gouv.fr>

### **D'autres liens utiles**

Algorithmique et programmation d'Amiens <http://pedagogie.ac-amiens.fr/maths/lycee/algo/>: pour des idées, des programmes et des renseignements

python Amiens: <http://pedagogie.ac-amiens.fr/math/AmiensPython1.0.exe> pour télécharger le langage avec des bibliothèques adaptées par Vincent Maille