

Le bulletin de l'APMEP - N° 547

AU FIL DES MATHS

de la maternelle à l'université...

Édition Janvier, Février, Mars 2023

Suites



APMEP

Association des Professeurs de Mathématiques de l'Enseignement Public

ASSOCIATION DES PROFESSEURS DE MATHÉMATIQUES DE L'ENSEIGNEMENT PUBLIC

26 rue Duméril, 75013 Paris

Tél. : 01 43 31 34 05

Courriel : secretariat-apmep@orange.fr - Site : <https://www.apmep.fr>

Présidente d'honneur : Christiane ZEHREN



Au fil des maths, c'est aussi une revue numérique augmentée :
<https://afdm.apmep.fr>

version réservée aux adhérents. Pour y accéder connectez-vous à votre compte via l'onglet *Au fil des maths* (page d'accueil du site) ou via le QRcode, ou suivez les logos ▶.

Si vous désirez rejoindre l'équipe d'*Au fil des maths* ou bien proposer un article, écrivez à aufildesmaths@apmep.fr

Annonces : pour toute demande de publicité, contactez Mireille GÉNIN mcgenin@wanadoo.fr

ÉQUIPE DE RÉDACTION

Directrice de publication : Claire PIOLTI-LAMORTHE.

Responsable coordinatrice de l'équipe : Cécile KERBOUL.

Rédacteurs : Marie-Ange BALLEREAU, Vincent BECK, François BOUCHER, Richard CABASSUT, Séverine CHASSAGNE-LAMBERT, Frédéric DE LIGT, Mireille GÉNIN, Cécile KERBOUL, Valérie LAROSE, Alexane LUCAS, Lise MALRIEU, Marie-Line MOUREAU, Serge PETIT, Daniel VAGOST, Thomas VILLEMONTÉIX, Christine ZELTY.

« **Fils rouges** » numériques : Gwenaëlle CLÉMENT, Nada DRAGOVIC, Fanny DUHAMEL, Laure ÉTÉVEZ, Marianne FABRE, Robert FERRÉOL, Cédric GROLLEAU, Louise GROLLEAU, Yann JEANRENAUD, Armand LACHAND, Agnès VEYRON.

Illustrateurs : Pol LE GALL, Nicolas CLÉMENT, Sixtine MARÉCHAL.

Équipe TeXnique : Laure BIENAIMÉ, François COUTURIER, Isabelle FLAVIER, Philippe PAUL, François PÉTIARD, Guillaume SEGUIN, Sébastien SOUCAZE, Sophie SUCHARD, Michel SUQUET.

Maquette : Olivier REBOUX.

Correspondant Publimath : François PÉTIARD.

Votre adhésion à l'APMEP vous abonne automatiquement à *Au fil des maths*.

Pour les établissements, le prix de l'abonnement est de 60 € par an.

La revue peut être achetée au numéro au prix de 15 € sur la boutique en ligne de l'APMEP.

Mise en page : François PÉTIARD

Dépôt légal : Mars 2023. ISSN : 2608-9297.

Impression : Imprimerie Corlet

ZI, rue Maximilien Vox BP 86, 14110 Condé-sur-Noireau



Démonstrations et programmes

Dans cet article, Didier Dacunha-Castelle questionne le lien entre mathématiques et informatique, en particulier le rapport entre démonstrations en mathématiques et programmes informatiques.

Didier Dacunha-Castelle

Il y a un large consensus pour enseigner l'informatique dans le secondaire, les élèves écrivant et exécutant des programmes. Reste à savoir comment faire une place convenable à une discipline nouvelle et répondre à la question : faut-il lier informatique et mathématiques dans l'enseignement secondaire, point de vue semble-t-il aujourd'hui minoritaire, ou faire entrer l'informatique dans les cursus comme nouvelle discipline indépendante des mathématiques ? Ceci au moment où l'informatique tend à prendre une place distincte des mathématiques dans la société, l'apport des mathématiques étant souvent occulté aux yeux des non spécialistes (par exemple dans l'Intelligence Artificielle).

Je questionnerai ici le rapport entre démonstrations en mathématiques et programmes informatiques et partirai de deux propositions qui font souvent bondir mes interlocuteurs :

- on peut associer à toute démonstration un programme ;
- une démonstration est exacte si et seulement si le programme associé est correct (« tourne » correctement).

Commençons par donner les ingrédients des théories mathématiques comme l'arithmétique ou la géométrie.

D'abord des *règles de déduction*. La règle la plus connue en est le *modus ponens* : de la proposition A et de la proposition $A \rightarrow B$, on déduit la proposition B .

Après les règles viennent les *axiomes*. Chaque théorie est définie par des axiomes, par exemple les fameux postulats d'Euclide de la géométrie. Ce sont des propositions décrétées vraies par consensus, portées par des considérations précises, d'ordre pratique ou intuitif par ceux qui travaillent dans le cadre d'une certaine théorie définie justement par un ensemble d'axiomes (qui en particulier ne doivent pas être redondants et si possible non contradictoires de manière évidente !). *Chaque théorie est donc construite à partir de ses axiomes, considérés comme nécessaires et suffisants pour tout faire à l'intérieur de cette théorie.* Tout faire voulant dire appliquer l'ensemble de règles logiques.

La plupart des mathématiciens et tous les autres utilisateurs des mathématiques sans exception, en particulier les élèves, se servent inconsciemment d'un certain nombre d'axiomes, comme ceux de l'arithmétique élémentaire, de la géométrie euclidienne (sauf s'il leur est demandé de citer les axiomes qu'ils utilisent) mais plus rarement ceux de la théorie des ensembles¹.

Une remarque du mathématicien et informaticien Jean-Louis Krivine² m'a marqué, quand nous étions étudiants et qu'il s'intéressait à la logique mathématique, chose très rare en France à l'époque : « *Toutefois, ce ne peut être là le fin mot de l'histoire. En effet, que savons-nous au bout du compte ? Eh bien, que prouver un théorème,*

1. À mon avis, la théorie des ensembles demeure le cadre fondamental des mathématiques malgré les tentatives pour la supplanter.

2. Voir ses articles sur la correspondance de Curry Howard sur son site [▶](#), certains sont destinés à un large public.



une fois fixés les axiomes des mathématiques dans lesquelles on a choisi de travailler, consiste à écrire un texte qui doit suivre des règles absolument rigoureuses et se terminer par l'énoncé du théorème en question. Mais pourquoi donc se fatiguer tellement à prouver des théorèmes, s'il ne s'agit que d'un jeu d'écriture formel ? »

Personne ne peut croire sérieusement que les mathématiques se réduisent à un jeu purement combinatoire et ce depuis des milliers d'années. Les mathématiques valent en premier lieu par leur efficacité dans les applications très variées et notamment parce que la physique a des lois s'écrivant mathématiquement. Comprendre pourquoi les mathématiques ne sont sûrement pas un jeu formel mérite réflexion. Je ne peux dans le format de cet article aborder tous les aspects de ce problème, notamment ceux sur l'origine des mathématiques et pourquoi la physique « s'écrit » en langage mathématique. Je me limiterai à ce qui concerne le rapport entre mathématiques et informatique avec en vue l'enseignement secondaire.

Au milieu du xx^e siècle, deux logiciens, Curry puis Howard, ont démontré qu'à toute preuve (démonstration) d'un théorème relevant de la logique élémentaire³ (celle que nous souhaitons voir appliquer assez tôt par nos élèves et qui ne l'est pas toujours par nos étudiants), on pouvait associer un programme. Il fallait donc définir ce qu'était un programme à une époque où l'informatique n'existait pas. Les logiciens de l'époque sont passés par un chemin purement mathématique. Je pense, peut-être à tort, qu'aujourd'hui chacun d'entre nous a une idée assez précise sur ce qu'est un programme. La démonstration de cette correspondance dite de Curry-Howard est passée par l'invention de deux langages utiles tant pour les mathématiques (le choix des règles de déduction) que pour l'informatique (le choix des instructions) :

- la logique combinatoire de Curry (six règles [1]);
- le lambda calcul de Church (trois règles [1]).

3. La logique du premier ordre puis la logique du deuxième ordre (sans le tiers exclu).

Ces langages permettent d'écrire toutes les démonstrations mathématiques. Le passage de la logique combinatoire voisine du langage machine (celui compris par les ordinateurs) au lambda calcul se fait à partir d'une *abstraction* considérable. Ce dernier langage, c'est un point essentiel, peut se « lire » soit en version mathématique, soit en version informatique. Entrer dans la démonstration de la correspondance preuve/programme nécessite d'explorer ces langages, je ne peux le faire ici faute de place.

La correspondance prend la forme : à toute preuve d'un théorème correspond un programme.

À l'ensemble des preuves d'un théorème correspond donc l'ensemble des programmes qui exécutent le même travail, qualité appelée en informatique *spécification* du programme (la spécification n'a pas de définition formelle en informatique, en voici une, surprenante, pour les programmes correspondant à des preuves : la spécification correspond à un théorème précis, explicitable par une formule!).

La correspondance preuve/programme fournit un dictionnaire maths-info dont voici quelques exemples élémentaires :

- à *théorie de la démonstration* correspond *programmation* (dans un langage donné) ;
- à *règle logique* correspond *instruction* ;
- à *preuve* (ou *démonstration*) correspond *programme* ;
- à *application d'une règle de déduction* correspond *application d'une instruction*, ainsi au *modus ponens* correspond l'application d'une instruction à un argument ;
- à *théorème* (proposition prouvée) correspond *spécification* (tâche à effectuer) d'un programme (fonctionnant correctement) ;
- à la *preuve par récurrence* correspond la *boucle for* ;
- à la *règle de l'absurdité* (faux implique n'importe quoi) correspond *exit* ;





- à l'entier 3 correspond le programme qui consiste à exécuter trois fois le programme qu'il prend comme argument, interprétation des entiers due à Church.

Il faudrait aussi développer les analogies entre *fonctions et programmes*, notamment l'opération fondamentale de *composition*, que les deux notions partagent.

Je vais être maintenant un tout petit peu provocant : savoir si les démonstrations fournissent des énoncés « vrais » est une question qui n'a pas de sens absolu. La notion purement mathématique de vérité est problématique faute de fondements suffisants. Par contre, si la notion de rigueur en mathématique n'est pas définissable, choisir un niveau de rigueur a un sens pragmatique dans l'enseignement ou dans la recherche.

On ne peut pas dire d'un programme qu'il est vrai ou faux : il s'agit seulement de savoir s'il fonctionne ou non. La principale raison d'être des démonstrations mathématiques est tout simplement de fournir des programmes corrects, et pas du tout d'établir des « vérités universelles ». Pourquoi $2 + 2 = 4$ est-il « vrai » ? Parce que, si vous programmez en partant de $2 + 2 = 3$, l'exécution de vos programmes va tourner au désastre. On peut dire des choses analogues pour le théorème de Pythagore.

Allons plus loin en apprenant des erreurs de nos glorieux ancêtres. Hilbert, étant donné que les mathématiciens ne pouvaient se passer de *l'infini*, voulait mécaniser les mathématiques, en prouvant que toute démonstration pouvait se faire en utilisant des *structures finies*. Ensuite, on pourrait tranquillement se servir de l'infini et vaquer à nos occupations mathématiques habituelles. Mais quels sont, au juste, ces objets finis ? Hilbert a cru qu'il s'agissait des *formules mathématiques* avec leurs règles de démonstration évoquées plus haut. Il s'est trompé, à son époque ces mystérieux objets finis, les programmes, n'avaient pas encore été découverts. Son erreur a été montrée dans les

années 1930, mais ce n'est que trente ans plus tard que le lien preuve-programme a été trouvé dans un cadre significatif. Au-delà de l'erreur de Hilbert, il demeure que derrière les notions abstraites, dont l'infini sous toutes ses formes, se cache une manipulation concrète de structures finies, les programmes, objets matériels et manipulables suivant les méthodes de programmation.

Dès lors, on peut dire à des élèves qu'une démonstration est exacte si et seulement si le programme associé dans la correspondance tourne correctement. Nous ne sommes plus du tout dans le ciel des idées, ciel encore cher à bien des mathématiciens, mais bien sur un terrain très concret. J'aurais tendance à penser qu'il est plus simple d'apprendre ce qu'est une boucle que la récurrence⁴ et que le parallèle entre certaines instructions en informatique et leur traduction en logique élémentaire est un changement de cadre didactique très intéressant ! Je ne peux ici détailler tous les ponts qui lient mathématiques et informatique et je n'ai pas l'expérience suffisante de tout ce que l'on peut tirer de cette correspondance pour faire toucher du doigt ce lien profond à des élèves ou à des étudiants. Mais ce lien doit nous faire réfléchir. Il est certain que l'informatique doit s'enseigner par l'usage de l'ordinateur (c'est ce qui la différencie des mathématiques) et donc par la pratique de la programmation. Mais coder et seulement coder sans ouverture sur ce lien conduira inévitablement à former des programmeurs sans perspectives professionnelles.

Revenons aux démonstrations. Comme le remarquait Daniel Reisz [3], même si le formalisme mathématique était parfaitement connu du lycéen au mathématicien professionnel, presque personne ne ferait de preuves formelles, simplement parce que faire des mathématiques de manière purement formelle est impraticable, quoique théoriquement possible ! Ceci dit, il faut absolument essayer de faire comprendre aux élèves ce qu'est un « raisonnement » mathématique. Ceci passe par une question dont les réponses ne peuvent

4. Je renvoie sur ce sujet à l'article de Jean-Paul Roy [2].



être que pragmatiques. Pourquoi avons-nous le droit, comme l'ont nos élèves et nos étudiants, de mélanger le langage naturel et le langage formel des mathématiques pour faire une démonstration ? La question est posée dans cette revue à propos de l'égalité [4]. Le fondement théorique pour aborder le problème de l'usage de la langue ordinaire « mélangée » avec le symbolisme mathématique est pour l'essentiel le théorème de complétude de Gödel⁵, inaccessible au niveau Secondaire et passé sous silence en général dans le Supérieur. Il n'est pas raisonnable de ne jamais parler aux apprenants de ce problème qu'ils rencontrent au quotidien.

Je me suis intéressé à l'enseignement des maths, il y a plus de soixante ans, à l'époque de la réforme dite des maths modernes, au milieu de la confusion régnante. Le délire de l'« axiomatisation » de la géométrie élémentaire a effaré, à juste titre, bien des personnes. Nous avons alors jeté le bébé avec le bain, la logique élémentaire avec des définitions moquées par le *Canard enchaîné*. Il est évident qu'enseigner cette logique est nécessaire pour apprendre la programmation. J'ai consacré ma vie professionnelle à développer et à enseigner probabilités et statistiques, y compris dans l'enseignement secondaire. J'ai vu dans les probabilités, un moyen (sans trop l'afficher, un peu lâchement) d'introduire un peu de logique et de théorie des ensembles ! La statistique, si méprisée dans ma jeunesse, explose aujourd'hui avec l'Intelligence Artificielle. Mais la logique reste assez mal aimée des courants dominants comme les probabilités l'ont été si longtemps.

Il faut donc réfléchir, en particulier dans le cadre de l'APMEP et de son groupe de travail Mathématiques et Informatique , à cette évidence : [le lien entre informatique théorique et mathématiques porte une interaction et une correspondance entre les concepts de base de ces disciplines, concepts utilisés pour construire leur enseignement](#). Il ne faudrait pas clore brutalement

cette réflexion en décrétant qu'il faut construire deux disciplines autonomes dont l'histoire des cursus montre (hélas) qu'elles seront séparées dans le futur par des intérêts contradictoires de postes et d'horaires.

Les mathématiques ne se réduisent pas à faire des démonstrations. Intuition, heuristique et pragmatisme ont leur place comme le soulignait Daniel Reisz [3]. Le lien entre mathématiques et informatique ne se résume pas au lien entre preuves et programmes (la simulation en est un exemple significatif). Surtout, ce lien ne concerne pas seulement les élèves et les disciplines scientifiques. La situation actuelle des mathématiques au lycée me conforte dans l'idée que le problème que pose ce lien doit être traité en prenant son temps et avec tous les enseignants concernés.

En trente ans, l'ordinateur est devenu plus commun et son usage dans ces domaines plus convivial. Par exemple, on ne peut plus enseigner la statistique à ses élèves sans enseigner la programmation qui accompagne le traitement de données. Il me semble donc probable que l'apport « didactique » de l'informatique aux mathématiques sera important assez tôt dans la scolarité.

Pour terminer, il me semble que la notion d'algorithme soulève une difficulté. Installée depuis des siècles côté mathématique, avec pignon sur rue aujourd'hui, sa définition et son usage restent flous à mes yeux, même si on les trouve un peu partout ! Un algorithme est-il là pour décrire des programmes qui vont dépendre d'un langage de programmation mais aussi du système d'exploitation de l'ordinateur ? Les critères pratiques pour passer des algorithmes aux programmes sont difficiles à manier. L'algorithmique ne peut être une véritable théorie aujourd'hui, sauf à en préciser et à en restreindre les contours et les objectifs et cela n'est pas trivial ! Je pense donc qu'il vaut mieux se centrer sur le concept de programme.

*
* *

5. Démonstré par Gödel dans les années 1930. Ce théorème est fondamental, il porte sur la notion de modèle en logique. Il est hélas moins connu que le théorème d'incomplétude qui fait souvent le délice des philosophes (et de réunions mondaines).





Un mot sur le futur au-delà de l'enseignement

Avec le développement de la correspondance preuves-programmes, la théorie des démonstrations est devenue un domaine captivant et vivant des mathématiques. À ses débuts, la correspondance de Curry-Howard ne concernait pas les axiomes et donc pas toutes les mathématiques. En fait, elle ne concernait pas l'infini et seulement la correspondance entre structures mathématiques finies et programmes, rien de vraiment révolutionnaire. Traduire les axiomes en programmes n'avait rien de trivial. Il a fallu attendre les années 1990 pour que Timothy Griffin, très jeune informaticien, donne le programme correspondant au tiers exclu (équivalent au raisonnement par l'absurde) ; ceci à la surprise générale : tous les spécialistes croyaient la chose impossible pour des tas de mauvaises raisons. Ce résultat est *révolutionnaire*. Le programme associé au tiers exclu consiste, dit très grossièrement, à sauver l'environnement de l'ordinateur en cas de problème, souvent signalé par un bip désagréable. Par exemple en langage C, à sauver la position du pointeur sur la pile et le travail en cours. Jean-Louis Krivine a ensuite donné les programmes correspondant aux axiomes de la théorie des ensembles puis finalement en 2020 à toutes les variantes de l'axiome du choix, de fait à toutes les mathématiques. Ce qui est encore plus intéressant, c'est que pour faire ce travail, il utilise la logique combinatoire mais doit ajouter *des règles qui correspondent à des instructions*, à des objets *bien connus en informatique* comme *la pile en premier lieu*, l'ordonnateur de tâches, l'horloge etc. Si l'on réfléchit à ces résultats,

on constate que les axiomes n'ont *rien de gratuit*. Les programmes qui leur correspondent sont associés à des notions trouvées par les informaticiens de façon quasiment expérimentale. Les systèmes d'exploitation des ordinateurs modernes sont largement construits à partir de ces notions. Cela ancre encore plus les mathématiques dans une réalité matérielle. Les mathématiciens, dont Turing et von Neumann, ont inventé l'informatique (avant que les ordinateurs n'existent) ; Griffin, Krivine et quelques autres permettent une avancée simultanée des deux disciplines que je ne peux détailler. Ils ouvrent aussi un chemin pour réfléchir aux origines des mathématiques, et c'est une autre histoire qui commence, elle aussi passionnante, mais qui demande de s'ouvrir sur d'autres disciplines !

Références

- [1] Jean-Louis Krivine. *Lambda-calcul, types et modèles*. Elsevier-Masson, 1990.
- [2] Jean-Paul Roy. « Sur la récurrence et la dichotomie au lycée ». In : *Au fil des maths* n° 541 (juillet-septembre 2021). APMEP. , p. 63-68.
- [3] Daniel Reisz. « Qu'entendons-nous par « Démontrer que... » ? » In : *PLOT* n° 39 (2014). APMEP. .
- [4] François Boucher. « Petite enquête sur l'égalité. » In : *Au fil des maths* n° 540 (avril-juin 2021). APMEP. , p. 73-80.



Ancien professeur à l'université Paris Sud-Orsay et à l'École Normale Supérieure, Didier Dacunha-Castelle est maintenant retraité. Pour en savoir plus, n'hésitez pas à consulter son site internet . dacunhacastelledidier@gmail.com

© APMEP Mars 2023

Sommaire du n° 547

Suites

Éditorial	1	Renforcer la culture scientifique de nos élèves par la lecture — Jessica Gouirand-Thuillet	54
Opinions	3	Ouvertures	58
Les positions de l'APMEP — Claire Piolti-Lamorthe, présidente de l'APMEP	3	Démonstrations et programmes — Didier Dacunha-Castelle	58
Renvoyez l'ascenseur (2) — Agnès Veyron	7	Preuves visuelles II — François Boucher	63
La dyscalculie existe-t-elle? — Serge Petit	13	Récréations	69
 Des <i>patterns</i> dans les classes! — Claire Piolti-Lamorthe, Sophie Roubin, Jana Trgalová & les membres du groupe PAREP ¹	19	 Un peu de e-magie! — Dominique Souder	69
Avec les élèves	29	Au fil des problèmes — Frédéric de Ligt	71
 Suites logiques en maternelle — Sandrine Lemaire	29	Au fil du temps	74
 Des suites au collège : pourquoi pas des fractales? — Lise Malrieu	36	 Pascal, triangle arithmétique, combinaisons et récurrence — Dominique Baroux & Martine Bühler	74
Le rapporteur <i>Recto-Verso</i> — Patrice Pellegrin	41	Modélisation mathématique et activités économiques — Pierre Arnoux & Véronique Le Payen Poublan	84
 Vous avez dit SUITES... — Mireille Génin	43	Le CDI de Marie-Ange — Marie-Ange Ballereau	89
 Haricots en suite... — Sébastien Corneau	45	Matériaux pour une documentation	91
Un jeu entre amis pas si anodin — Vincent Billoud, Fabrice Richard & Charlotte Vulliez	50		



CultureMATH



APMEP

www.apmep.fr