

**Journées Nationales d'Albi 1996
Conférence du dimanche 27 octobre**

**Un canevas pour
l'enseignement des mathématiques
en école d'ingénieur :
de la modélisation au
logiciel**

J. Noailles

Professeur de Mathématiques à
l'ENSEEIH

Dans les écoles d'ingénieurs, y compris celles formant des ingénieurs mathématiciens appliqués (ENSEEIH, ENSI-MAG,...), les mathématiques ne mènent plus la vie autonome qui est la leur, auparavant, au lycée. D'introverties, elles deviennent extraverties soucieuses du contexte et de leur finalité.

Deux rôles ne leur sont pas contestés. Celui de point de passage obligé dans la formulation mathématique de la plupart des problèmes de l'ingénieur et celui de pourvoyeur en méthodes de calcul. En fait, formulation et calcul ne sont que les étapes les plus visibles du parcours mathématique qui va du problème réel à sa résolution «in situ».

Modéliser un problème, c'est capter les traits qui en sont l'essence et les traduire mathématiquement - comment faire autrement ? - en vue d'une résolution qui sera pratiquement toujours de nature numérique. S'il ne faut pas confondre le problème et sa formulation, on ne doit jamais oublier que celle-ci

conditionne sa solution. Que dire d'une modélisation qui conduirait à un système linéaire de dizaines de milliers de variables et d'équations et dont la matrice serait dense ! Elle ne serait d'aucune utilité !

Une fois le modèle établi, on doit en faire l'analyse mathématique : s'interroger sur l'existence, l'unicité de ses solutions. Le modèle, qui n'est qu'une approximation de la réalité, peut n'avoir aucune solution au sens mathématique, alors que celle-ci en possède physiquement une. Ce travail d'analyse peut être très délicat. Pensons à celui réclamé par une équation aux dérivées partielles non linéaire ! La démarche d'analyse fonctionnelle qu'elle exige aura de grandes chances d'être perçue comme compliquée, peu significative, voire inutile par l'élève. Pourtant celui-ci doit avoir une idée de ces difficultés, ne serait-ce que pour comprendre certains comportements numériques ultérieurs. Comment éviter, alors, de connaître les concepts de base et les théorèmes essentiels de la topologie générale, de même que les espaces fonctionnels les plus courants. A ce propos, les espaces L^2 sont incontournables pour le rôle charnière qu'ils ont entre réalité physique et modélisation mathématique. Leur structure hilbertienne, fondamentale, mérite d'être étudiée en tant que telle pour une compréhension profonde et une utilisation intelligente des éléments d'Analyse de Fourier et d'Analyse harmonique traditionnellement considérés comme deux piliers des mathématiques dites de l'ingénieur.

Tout cet effort mathématique ne débouche pas directement sur la solution, l'accès à celle-ci étant numérique. C'est le rôle de l'Analyse Numérique de nous ouvrir la voie. Les processus physiques font, en général, intervenir le temps et l'espace qu'il faut discrétiser. Un cours d'Analyse Numérique ne se contente pas de la description des méthodes qui conduisent du problème «continu» au problème discrétisé : différences finies, éléments finis, volumes finis, méthodes spectrales, méthodes particulières. S'appuyant sur l'analyse fonctionnelle et l'analyse classique, il doit justifier ce passage du continu au discrétisé en étudiant sa consistance et sa convergence.

Cette étape d'approximation consiste, très schématiquement, à remplacer des espaces vectoriels fonctionnels (donc de dimension infinie) par des espaces vectoriels de dimension finie. Mais le problème discrétisé obtenu n'est pas susceptible d'un traitement immédiat sur ordinateur. Il doit être, au préalable, décomposé en problèmes numériques de base. Au nombre de ces derniers se trouvent la résolution des systèmes linéaires, la résolution des équations algébriques, les problèmes aux valeurs propres, les problèmes de moindres carrés, les problèmes d'optimisation avec ou sans contrainte. Sans surprise, cette énumération coïncide avec celle des principales «routines» de toute bibliothèque numérique. C'est le rôle d'un enseignant en

Algorithmique Numérique de nous décrire les différents algorithmes de résolution, de nous éclairer sur leur choix grâce à l'étude de leur comportement numérique. Cette étude est d'abord celle de leur domaine d'utilisation et de la nature de leur convergence. Ainsi pour les équations non linéaires, la distinction entre méthodes localement convergentes (le choix du point initial est décisif) et méthodes dites globalement convergentes (moins dépendantes de ce choix) est essentielle. Cette étude est ensuite celle de la qualité de leur convergence : stabilité, précision, vitesse. Elle est enfin celle de leur coût numérique : coût de calcul d'une itération, coût de stockage des données. Si, dans cette démarche, l'analyse classique est une composante évidente, il en est une, plus cachée, plus subtile, due à la spécificité du calculateur qui ramène l'ensemble des réels à un ensemble de cardinal fini : l'ensemble des nombres flottants. On la décrit souvent comme le passage de l'arithmétique exacte à l'arithmétique inexacte. L'exemple suivant illustrera la nécessité de cette étude de l'erreur.

Considérons le système linéaire $Ax = b$ suivant :

$$\begin{cases} 0,550x_1 + 0,423x_2 = 0,127 \\ 0,484x_1 + 0,372x_2 = 0,112 \end{cases}$$

Supposons que deux algorithmes nous aient conduits après résolution en arithmétique inexacte :

$$\text{l'un à } \tilde{x} = (1,7, -1,91)$$

$$\text{l'autre à } \bar{x} = (1,01, -0,99)$$

Comment juger de la qualité numérique de ces deux résultats ? Deux attitudes sont naturelles. L'une consiste à vérifier qui des vecteurs \tilde{x} et \bar{x} est celui qui vérifie "au mieux" le système linéaire, l'autre, quel est celui qui est le plus proche de la solution exacte $x^* = (1, -1)$.

En utilisant la norme $\|\cdot\|_\infty$, la première démarche donne :

$$\|A\tilde{x} - b\|_\infty = 2,8 \times 10^{-4} \quad \text{et} \quad \|A\bar{x} - b\|_\infty = 9,73 \times 10^{-3}$$

En ce sens, le résultat \tilde{x} est meilleur que le résultat \bar{x} .

La seconde démarche donne :

$$\|\tilde{x} - x^*\| = 0,91 \quad \text{et} \quad \|\bar{x} - x^*\| = 0,01$$

En ce sens, le résultat \bar{x} est meilleur que le résultat \tilde{x} . C'est l'analyse de l'erreur, domaine délicat, qui permet d'expliquer la situation précédente.

Reprenant notre marche vers la solution, arrive maintenant l'heure de l'implantation informatique. Une architecture et un langage informatique ayant été choisis ou fixés, le travail va porter sur la définition des structures

de données. Prenant l'exemple du traitement des matrices creuses (i.e. avec beaucoup d'éléments nuls) de très grande taille, on s'aperçoit vite qu'il est difficile de dissocier la partie Algèbre Linéaire Numérique proprement dite de la représentation des données. C'est ce qui rend l'enseignement de cette étape particulièrement délicat. Au carrefour entre algorithmique numérique et génie logiciel, il réclame compétence et maîtrise dans les deux domaines. Les résultats obtenus doivent enfin être organisés le plus synthétiquement possible, tout en perdant un minimum d'information. Les logiciels graphiques et de visualisation sont là pour y pourvoir.

Que conclure à l'issue de ce rapide panorama ? Tout d'abord qu'il est loin d'être exhaustif puisqu'il laisse de côté les problèmes discrets, c'est-à-dire ceux qui se posent des ensembles de cardinal fini : problèmes d'ordonnement, problèmes de programmation mathématique en variables entières... Pour l'ingénieur, ils sont de nature essentiellement calculatoire et leur traitement réclame une approche plus heuristique. Ils prennent surtout place dans les cours de théorie des graphes, recherche opérationnelle et même intelligence artificielle. La seconde remarque, idéologique celle-là, émane de ceux qui, confondant l'outil (calculateurs et logiciels) avec la main (les mathématiques), pensent que le développement des calculateurs, des logiciels scientifiques et d'interfaces de plus en plus intelligents annonce le déclin des mathématiques. Comme si, bientôt, on allait pouvoir s'en passer ! Comme si un outil de plus en plus performant ne réclamait pas une main de plus en plus solide !

En effet, supposons que des ingénieurs disposent d'un parc infini de tels bibliothèques, logiciels et interfaces. Il resterait toujours à savoir : comment choisir ? comment utiliser ? Penser qu'il est possible de le faire sans une connaissance mathématique suffisante reviendrait à dire que, hormis les concepteurs et réalisateurs de logiciels, les ingénieurs en seraient réduits principalement à taper les données d'entrée sur une console et en attendre les résultats. Et même alors, comment interpréter ceux-ci sans comprendre le rôle du modèle, celui de la méthode de résolution, celui du calculateur ? Oublie-t-on qu'il y a encore très peu de temps, l'ingénieur programait lui-même ses calculs et, qu'après le temps passé à la conception, l'écriture, la mise au point et l'exploitation de ses programmes, il connaissait parfaitement les mécanismes mathématiques qu'il utilisait ? Que cet effort soit maintenant transporté au niveau de l'analyse des résultats ne change rien au besoin d'une culture mathématique solide. Bien au contraire !

N'ajoutons rien en guise de conclusion à ce que disait le physicien BOUASSE : *« Il est plus difficile d'apprendre à se passer des mathématiques que d'apprendre à s'en servir ».*