

**Pierre Audibert**

**Initiation aux mathématiques  
générales**

**Mathématiques et informatique**

**(niveau L1 ou Bac S ++)**

Cours. Exercices corrigés. Programmation.

disponible sur Internet, ainsi que d'autres,  
sur le site :

***[pierreaudibert.fr](http://pierreaudibert.fr)**, rubrique *enseignements**

---

**Table des matières**

*Niveau L1 ou prépa écoles commerciales*

Introduction : Mathématiques et informatique

- 1) Test de niveau
- 2) Calculs
- 3) Raisonnement par récurrence
- 4) Equations, inéquations, systèmes d'équations linéaires
- 5) Etude de fonctions
- 6) Intégration
- 7) Applications et bijections. Bijection réciproque
- 8) Logarithme et exponentielle
- 9) Suites numériques
- 10) Notions de combinatoire
- 11) Division euclidienne, pgcd et algorithme d'Euclide

*A cela s'ajoutent quelques chapitres au programme de Terminale S*

# Mathématiques et informatique

La caractéristique de ce cours est de lier l'informatique et les mathématiques. Dans chaque chapitre, des programmes informatiques sont donnés, qu'il conviendra de tester et d'assimiler. Ils permettent d'appliquer, d'illustrer, de vérifier ce que donne la théorie. En cas de besoin, quand la théorie est inopérante, ou du moins les connaissances théoriques que l'on possède soi-même, ils permettent le traitement expérimental des problèmes. Pour ces programmes, nous avons utilisé le langage *C* de base, auquel il conviendra d'adjoindre un logiciel de graphisme (*SDL* recommandé).<sup>1</sup>

Ce cours met aussi l'accent sur les exercices et les exemples. Ceux-ci mettent en situation la théorie, permettant de l'appliquer et par là-même de la comprendre. En matière de connaissances théoriques, ce cours donne le bagage minimal sur ce qu'il vaut mieux savoir quand on a des problèmes à résoudre ou des calculs à faire. En cas de besoin, un jour ou l'autre, on pourra toujours s'y référer. Mais une succession de propriétés et de théorèmes serait une abstraction creuse sans une multitude d'exercices.

Nous avons volontairement privilégié le chapitre sur les suites. De nombreux phénomènes réels sont modélisés par une succession de nombres  $u_0, u_1, u_2, u_3, \dots$ , correspondant à leur évolution dans le temps. Une telle suite est en général donnée par son premier élément et par une règle de passage d'un terme au suivant, du style  $u_{n+1} = f(u_n)$ . On a là l'essence d'un programme informatique : les conditions initiales, et une boucle qui détermine un terme par rapport à celui qui le précède, avec affichage des résultats à la fin.<sup>2</sup>

---

<sup>1</sup> Si vous préférez, vous pouvez utiliser tout autre langage. Seules importent la justesse et la robustesse du programme, c'est-à-dire sa capacité à donner des résultats exacts en toutes circonstances.

Une initiation rapide au langage *C* et au graphisme *SDL* est disponible sur mon site.

<sup>2</sup> Voici le programme en *C* pour une suite  $(u_n)$  telle que  $u_0 = 1$  et  $u_{n+1} = 0,5 u_n + 4$ . L'objectif est d'obtenir le terme  $u_N$  pour  $N$  que l'on se donne, par exemple  $N = 20$  :

```
int N,u,i ; /* déclaration préalable des variables comme étant des entiers */
N=20 ;
u = 1 ;      /* on met u_0=1 dans la case mémoire (la variable) u */
for (i = 1 ; i <= N ; i ++ ) u = 0.5*u+4 ; /* la nouvelle valeur de u est fonction de
                                           l'ancienne. Ainsi u va contenir u_1 puis
                                           u_2, etc. */
printf (« %d », u ) ; /* on obtient u_N (ici u_20) affiché sur l'écran */
```