

6.5. Numériser

Nous désignerons par le verbe *numériser* toute activité ayant pour but d'adapter une procédure mathématique en vue de son exécution par un ordinateur ou une calculatrice programmable. Il s'agit bien là d'un changement de cadre car l'arithmétique d'un ordinateur est fondamentalement différente de celle des nombres réels.

Nous accorderons une place particulièrement importante à la numérisation pour plusieurs raisons :

1. Grâce aux calculatrices et ordinateurs, il est possible de réaliser rapidement des calculs beaucoup plus nombreux et plus complexes que ceux qui sont faisables « à la main ».

L'usage de ces moyens de calcul permet en conséquence de réaliser de véritables *expérimentations* numériques sur des sujets d'algèbre, analyse, probabilités, statistiques. Ils contribuent de ce fait à former une intuition qui, sans cela, ne repose que sur quelques impressions *a priori* pas toujours très fiables.

En géométrie, les activités de dessin aux instruments ont toujours permis une certaine expérimentation, néanmoins des logiciels tels que **Cabri Géomètre** d'une part permettent de réaliser des dessins beaucoup plus sophistiqués qu'auparavant et d'autre part donnent à ces figures une dimension dynamique qui renforce considérablement le côté expérimental.

2. L'expérimentation que nous venons d'évoquer se situe essentiellement à un niveau procédural. Elle est essentielle dans des sujets tels que l'analyse mathématique, dont on peut considérer que l'enseignement traditionnel néglige cette composante et se situe beaucoup trop rapidement au stade conceptuel. En témoignent les difficultés rencontrées avec les quantificateurs.
3. Cette expérimentation est également susceptible de jouer un rôle essentiel durant les phases d'exploration d'une activité de résolution de problème.
4. Enfin, *last but not least*, l'usage des calculatrices et ordinateurs est encore à nos yeux trop peu répandu.

Le travail sur ordinateur nécessite de bons logiciels. Notre rôle n'est pas de passer en revue ceux qui sont actuellement disponibles en en indiquant les avantages et les inconvénients. Attirons seulement l'attention sur la distinction à réaliser entre *logiciels scientifiques* et *logiciels pédagogiques*.

Les premiers sont souvent très puissants (et d'emploi parfois malaisé), souvent inutilement puissants pour les besoins de l'enseignement. Ils peuvent néanmoins rendre de grands services.

Les seconds sont moins nombreux. Il en existe de « petits », qui visent un problème particulier, de « plus gros » ayant vocation d'être utilisables en de nombreuses occasions. Ils sont souvent dus à l'œuvre d'enseignants « du terrain », ce qui leur confère des qualités (être proches des préoccupations des utilisateurs) et des défauts (un fini non professionnel). À chaque enseignant de faire son choix.

Dans la suite, nous mettrons en évidence une catégorie particulière de logiciels : les tableurs. Qu'on n'en déduise pas que nous excluons tous les autres. Qu'on n'exclue pas non plus de faire programmer les élèves eux-mêmes. Les logiciels qu'ils réaliseront seront moins performants et moins bien présentés que les logiciels professionnels, mais ils seront beaucoup plus convaincants. Au surplus, ce sera pour eux l'occasion de se battre avec de véritables problèmes et d'acquérir des compétences.

Programmer un algorithme est souvent assimilable à une activité de résolution de problème.

6.5.1 Numériser l'arithmétique

EXEMPLE 6.5.1 *Considérons une calculatrice qui encode les nombres, en numération décimale, à l'aide de trois informations : un signe (+ ou -), une partie décimale (la mantisse) comportant n chiffres, $0.a_1 \dots a_n$ (où le chiffre a_1 est nécessairement différent de 0) et un exposant de deux chiffres $e_1 e_2$ pouvant varier entre -99 et 99 . Combien de nombres différents peuvent-ils être manipulés par cette machine ?*

Cette description correspond à une écriture des nombres en notation scientifique. Le nombre réel associé au triplet $(\pm, 0.a_1 \dots a_n, e_1 e_2)$ est donc $\pm 0.a_1 \dots a_n \times 10^{e_1 e_2}$.

La question posée est assez simple : l'exposant peut prendre 199 valeurs différentes, la mantisse varie de $0.10 \dots 0$ à $0.9 \dots 9$ et peut donc prendre $90 \dots 0 = 9 \times 10^{n-1}$ valeurs. La machine peut donc manipuler $199 \times 9 \times 10^{n-1}$ nombres strictement positifs différents. Tenant compte des nombres négatifs et du zéro, nous arrivons à $3582 \times 10^{n-1} + 1$ nombres réels différents.

Quelle que soit la précision de la machine, elle ne peut traiter qu'un nombre fini de réels différents (de surcroît tous décimaux limités). On notera le plus grand et le plus petit (en valeur absolue) de ces réels : $0.9 \dots 9 \times 10^{99}$ et $0.10 \dots 0 \times 10^{-99}$. On note aussi que ces « nombres de la machine » ne sont pas uniformément répartis dans \mathbb{R} : plus ils sont petits (en valeur absolue), plus ils sont serrés. Ceci montre que seuls les *calculs d'erreurs relatives* sont significatifs.

Non seulement une machine ne peut traiter qu'un nombre fini de réels, mais les opérations qu'elle effectue ont des propriétés différentes des opérations dans \mathbb{R} .

EXEMPLE 6.5.2 *Fixons (arbitrairement) à 10 le nombre de chiffres significatifs de la machine précédente, et calculons $1 + 0.0000000001$ ou, si vous préférez, $0.1000000000 \times 10^1 + 0.1000000000 \times 10^{-9}$. Le résultat devrait comporter 11 chiffres significatifs, un de trop. La machine va donc arrondir ce résultat :*

$$1 + 0.0000000001 = 1$$

Tout nombre réel est donc absorbant par rapport à d'autres. Dans la foulée, il n'est pas difficile de voir que pour une machine, l'addition n'est pas associative, pas plus que la multiplication, que celle-ci n'est pas non plus distributive par rapport à l'addition. Bien entendu, la plupart des nombres n'ont pas d'inverse. (Il est intéressant de déterminer ceux qui en ont un.) Quant à la racine carrée ...

L'arithmétique d'une machine n'a pas que des propriétés négatives.

EXEMPLE 6.5.3 *L'addition et la multiplication sont commutatives.*

N'en déduisons pas qu'il nous est loisible d'effectuer les opérations dans n'importe quel ordre : dès qu'il y a plus d'une opération à effectuer, nous avons intérêt à organiser le calcul soigneusement.

EXEMPLE 6.5.4 *Pour calculer une somme $x_1 + x_2 + \dots + x_n$ de réels d'ordres de grandeurs fort différents, il y a souvent intérêt à ranger les nombres x_1, \dots, x_n du plus petit au plus grand.*

En additionnant d'abord les petits nombres entre eux, on limite l'éventuel effet absorbant des plus grands nombres par rapport aux plus petits.

Pour éviter le plus possible une augmentation importante des erreurs relatives, il est également déconseillé de soustraire deux nombres très voisins ou de diviser l'un par l'autre des nombres très petits.

EXEMPLE 6.5.5 *Un algorithme de calcul de π consiste à inscrire dans un cercle un carré, puis un octogone, un hexadécagone, etc. Le calcul du périmètre de ces polygones revient à déterminer la longueur de la corde d'un demi arc de cercle en fonction de la longueur de la corde de l'arc. Via le théorème de Pythagore, on obtient sans peine l'algorithme suivant, où s_n désigne la longueur du côté du polygone régulier à 2^{n+1} côtés :*

$$s_1 = \sqrt{2} \text{ et } s_{n+1} = \sqrt{2 - \sqrt{4 - s_n^2}}$$

Il reste alors à imprimer la suite des valeurs de $\ell_n = 2^n s_n$ pour obtenir une suite qui converge vers π .

Programmé sur le tableur **Excel**, cet algorithme nous a fourni le tableau suivant. La première colonne contient les valeurs de s_n , la deuxième, celles de 2^n , enfin la troisième colonne constituée des nombres ℓ_n doit normalement converger vers π .

On constate que jusqu'à la 15^e itération, les valeurs trouvées semblent bien se rapprocher de π . Ensuite elles s'en écartent, jusqu'à s'annuler pour $n = 28$ (comme le montrerait le tableau prolongé). Il ne sert à rien de mettre en cause le tableur et de refaire le calcul avec un autre logiciel. Le phénomène se reproduira peut-être un peu plus tard si on augmente la précision du calcul. Que s'est-il passé ?

1,41421356	2	2,82842712
0,76536686	4	3,06146746
0,39018064	8	3,12144515
0,19603428	16	3,13654849
0,09813535	32	3,14033116
0,04908246	64	3,14127725
0,02454308	128	3,1415138
0,01227177	256	3,14157294
0,00613591	512	3,14158773
0,00306796	1024	3,14159142
0,00153398	2048	3,14159235
0,00076699	4096	3,14159258
0,0003835	8192	3,14159263
0,00019175	16384	3,14159265
9,5874E-05	32768	3,14159265
4,7937E-05	65536	3,14159261
2,3968E-05	131072	3,14159291
1,1984E-05	262144	3,1415917
5,9921E-06	524288	3,14159655
2,9961E-06	1048576	3,14159655
1,498E-06	2097152	3,14151884
7,4892E-07	4194304	3,14120797
3,7461E-07	8388608	3,14245127

Il se fait que la suite s_n converge vers zéro. Le nombre $\sqrt{4 - s_n^2}$ se rapproche donc de 2, et la différence $2 - \sqrt{4 - s_n^2}$ porte sur deux nombres voisins l'un de l'autre. Les erreurs d'arrondi augmentant constamment finissent par l'emporter sur la « vraie valeur » et le résultat obtenu n'a plus aucune signification. Pour ne rien arranger, l'opération finale $2^n * s_n$ porte sur un nombre qui se rapproche de zéro et l'autre qui devient de plus en plus grand. Autant diviser deux nombres très petits.

Dans un tel cas, il faut retravailler l'algorithme, pour le mettre sous une forme qui se prête au calcul sur machine en éliminant les opérations litigieuses. Ici, nous ferons disparaître la soustraction $2 - \sqrt{4 - s_n^2}$ en multipliant et divisant par $2 + \sqrt{4 - s_n^2}$.

De façon précise :

- Pour tout n , posons $w_n = \sqrt{2 + \sqrt{4 - s_n^2}}$.
- On vérifie sans problème que $w_{n+1} = \sqrt{2 + w_n}$.
- Alors

$$\ell_{n+1} = 2^{n+1} s_{n+1} = 2^{n+1} \frac{\sqrt{2 - \sqrt{4 - s_n^2}} \cdot w_{n+1}}{w_{n+1}} = \frac{2^{n+1} s_n}{w_{n+1}} = \frac{2\ell_n}{w_{n+1}}$$

- En programmant cet algorithme transformé, on obtient une suite qui n'a pas le comportement pathologique rencontré plus haut.

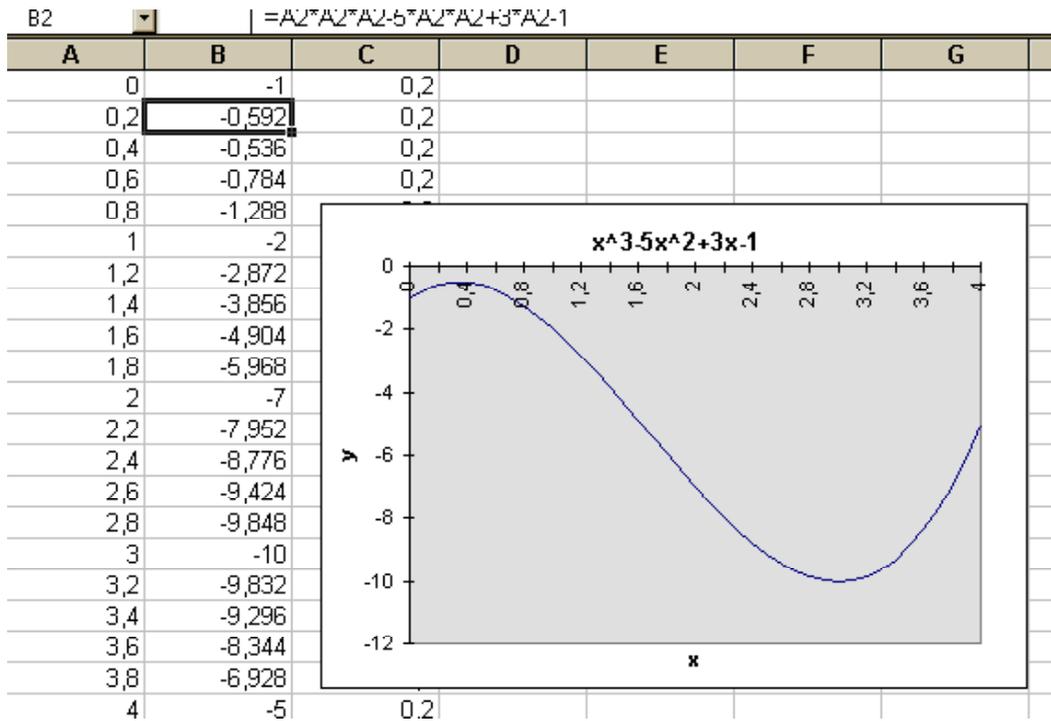
6.5.2 Numériser l'algèbre

En plus de logiciels adaptés à des sujets particuliers dont nous ne saurions faire ici un inventaire même restreint, nous devons attirer l'attention sur l'importance de l'emploi de *tableurs*. Nous venons de donner un exemple d'un tel usage à l'occasion du calcul de valeurs approchées de π . Nous en rencontrerons encore à plusieurs reprises. Les tableurs sont des outils de programmation très simples à manipuler et dont les versions actuelles permettent des réalisations remarquables, y compris sur le plan graphique. Au vu de leurs possibilités, la maîtrise de la manipulation de ces logiciels doit être considérée comme une compétence à acquérir par tous les élèves du secondaire, et cela dès les premières années de celui-ci.

EXEMPLE 6.5.6 *Étude graphique d'une fonction*

L'étude graphique d'une fonction peut être réalisée

1. en attribuant une colonne de cellules aux valeurs de la variable indépendante. Il suffit d'introduire dans la cellule de tête de cette colonne la valeur initiale et dans celle située juste en-dessous, une formule « de récurrence » du type « $= A_1 + 0,2$ ». En recopiant cette formule telle quelle dans les cellules suivantes de la même colonne, le domaine de la fonction est entièrement déterminé. Ce domaine est discrétisé, il a la forme d'une progression arithmétique. D'autres formes sont évidemment possibles.
2. en définissant la fonction dans une cellule située à côté de celle qui contient la valeur initiale de la variable et en recopiant cette définition dans les cellules situées en dessous. La fonction est automatiquement tabulée sur le domaine déterminé par la colonne des valeurs de la variable indépendante.
3. en faisant dessiner le graphique de la fonction à l'aide des commandes adéquates du menu.



Il suffit alors de modifier la valeur figurant dans la cellule de tête de la colonne-domaine pour obtenir à la fois une tabulation de la fonction sur un autre intervalle et le graphique correspondant. La présence simultanée à l'écran du tableau des valeurs de la fonction et du graphique correspondant permet à l'enseignant de centrer l'attention des élèves sur l'analyse des propriétés de la fonction plutôt que sur le calcul des valeurs et le dessin du graphique. La possibilité de modifier rapidement et avec aisance le tableau et le graphique permet de donner à cette analyse toute la finesse désirée.

Les tableaux se prêtent également très bien au calcul de sommes et de différences. Ils pourraient donc servir de support à la préparation algébrique d'un cours de calcul différentiel et intégral. Nous n'avons pas la possibilité de développer ce point de vue dans le cadre de ce travail. Contentons-nous de montrer une application au calcul des sommes des puissances k^e des naturels, $\sum_{n=1}^N n^k$.

EXEMPLE 6.5.7 *Nous nous situons dans une classe dont les élèves ont déjà rencontré les coefficients binomiaux $\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k!}$. A l'aide d'un tableur, et de la récurrence fondamentale $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$, construisons le triangle de Pascal. La cellule intersection de la ligne $n+1$ et de la colonne $k+1$ contient la valeur de $\binom{n}{k}$. Sélectionnons les cellules $n^\circ 0$ à $n+1$ de cette colonne $k+1$. Si notre tableur dispose d'une fonction « somme automatique », nous voyons apparaître la somme des valeurs sélectionnées à la dernière ligne de la fenêtre du tableur.*

Nous constatons (voir ci-dessous) que cette somme se trouve également dans la cellule située en colonne $n+2$, ligne $k+2$. De plus, il ne s'agit pas d'un hasard : faisant varier les valeurs de n et k , nous faisons toujours la même constatation. Nous pouvons donc conjecturer l'identité

$$\sum_{n=0}^N \binom{n}{k} = \binom{N+1}{k+1}$$

Le rappel de la récurrence fondamentale permet de justifier rapidement notre conjecture. Nous en déduisons par exemple

$$\sum_{n=1}^N n = \binom{N+1}{2} = \frac{N(N+1)}{2}$$

mais aussi puisque $n^2 = 2\binom{n-1}{2} + n$:

$$\begin{aligned} \sum_{n=1}^N n^2 &= 2 \sum_{n=1}^N \binom{n}{2} + \sum_{n=1}^N n = 2 \binom{N+1}{3} + \binom{N+1}{2} \\ &= \frac{N(N+1)(2N+1)}{6} \end{aligned}$$

Par récurrence, nous pouvons ainsi calculer $\sum_{n=1}^N n^k$ quel que soit k .

0	0	0	0	0	0
1	0	0	0	0	0
2	1	0	0	0	0
3	3	1	0	0	0
4	6	4	1	0	0
5	10	10	5	1	0
6	15	20	15	6	1
7	21	35	35	21	7
8	28	56	70	56	28
9	36	84	126	126	84
10	45	120	210	252	210
11	55	165	330	462	462
12	66	220	495	792	924
13	78	286	715	1287	1716
14	91	364	1001	2002	3003
15	105	455	1365	3003	5005
16	120	560	1820	4368	8008
17	136	680	2380	6188	12376
18	153	816	3060	8568	18564
19	171	969	3876	11628	27132
20	190	1140	4845	15504	38760
21	210	1330	5985	20349	54264
22	231	1540	7315	26334	74613
23	253	1771	8855	33649	100947

Feuil2 / Feuils3 / Feuils4 / Feuils5 / Feuils6 / Feuils7 / Feuils8 / Feuils9 / Feuils10 / | 4 |

Somme=455

6.5.3 Numériser l'analyse

Nous l'avons déjà remarqué au chapitre 3, les débuts de l'analyse, l'introduction de la notion de limite en particulier, sont délicats du fait de ce que nous ne disposons pas d'une procédure analytique générale pour déterminer la limite d'une suite quelconque.

La tentation est alors très forte de se contenter d'une introduction intuitive du genre $\lim_{n \rightarrow \infty} s_n = u$ si et seulement si s_n se rapproche d'aussi près que l'on veut de u lorsque n grandit. De cette façon, on installe chez les élèves un premier modèle mental. Mais, si après quelques exemples, on formule immédiatement la définition en ε, n_0 , on se situe à un niveau conceptuel beaucoup plus élevé et qui n'a pas été suffisamment préparé. Le résultat est trop souvent inconsistant.

La seule procédure vraiment générale pour étudier la convergence d'une suite est d'en calculer un grand nombre de valeurs, et de faire apparaître le phénomène de *stabilisation des décimales*.

Après suffisamment d'exemples, il deviendra naturel de se poser des questions du genre

A partir de quel terme de la suite, la 5^e décimale ne change-t-elle plus ?

Autrement dit, *quel est le plus petit n tel que $|s_n - u| \leq \frac{1}{2}10^{-5}$?* Il sera alors peut-être temps de penser à conceptualiser la notion de limite, et le formalisme des quantificateurs deviendra indispensable.

Dans un tel schéma, il est clair qu'un tableur peut rendre de grands services. L'exemple ci-contre ci-après illustre le calcul de $\lim_{x \rightarrow 0} \frac{\sin x}{x}$.

Malheureusement, comme on l'a vu plus haut, les choses ne fonctionnent pas toujours aussi harmonieusement que dans cet exemple. Des erreurs d'arrondi peuvent venir fausser les résultats au point que ceux-ci perdent toute signification.

Il ne conviendrait pas d'éviter systématiquement tout exemple de ce genre : ce serait donner aux élèves une confiance trop grande en la machine. Mais il faut certainement éviter de traiter de ce problème de fiabilité en même temps qu'on introduit un concept aussi important que celui de limite : ce serait semer la confusion la plus totale. Une fois le concept assimilé, il sera nécessaire de revenir sur la question et de faire remarquer, par exemple, que sur ordinateur une *série* de nombres converge dès que son terme général tend vers 0, ce qui, en mathématique, est en réalité faux.

L'emploi d'un tableur permettrait une approche procédurale d'autres sujets tels que la dérivée ou l'intégrale (voir par exemple au chapitre 12, la séquence d'enseignement consacrée à l'intégration).

D6	
A	B
1	1,57079633
0,5	1,04719755
0,25	1,01072102
0,125	1,00262265
0,0625	1,00065219
0,03125	1,00016283
0,015625	1,00004069
0,0078125	1,00001017
0,00390625	1,00000254
0,00195313	1,00000064
0,00097656	1,00000016
0,00048828	1,00000004
0,00024414	1,00000001
0,00012207	1
6,1035E-05	1
3,0518E-05	1
1,5259E-05	1
7,6294E-06	1

6.5.4 Numériser les probabilités

Avec les probabilités, nous abordons un domaine propice à la réalisation de véritables *expériences mathématiques*. Le concept même de probabilité résulte d'une *modélisation* de la notion de *fréquence* liée à l'observation *répétée* d'un phénomène particulier. Et nous savons que, pour faire apparaître une stabilisation de la fréquence, l'observation doit réellement être répétée de nombreuses fois. En termes pseudo-techniques, pour « gagner une décimale », il faut faire 100 fois plus d'observations.

Des expérimentations effectives (jets de pièces ou de dés, tirages de cartes, relevé des résultats du Lotto, etc) sont indispensables au début d'un cours de probabilité. Mais les circonstances concrètes de l'enseignement rendent malaisée la répétition d'une observation un très grand nombre de fois, disons *plusieurs milliers de fois*, alors que certains phénomènes n'apparaîtront qu'avec des expériences menées à une telle échelle.

EXEMPLE 6.5.8 *Combien de fois au minimum faut-il lancer une pièce de monnaie parfaitement symétrique pour que la probabilité d'avoir une séquence d'exactly dix jets consécutifs identiques soit supérieure à $\frac{1}{2}$?*

On peut calculer la probabilité d'apparition d'une suite de n jets consécutifs identiques pour $n = 1, 2, \dots$: chaque fois qu'on lance une pièce, il y a une chance sur deux pour que le lancer suivant donne le même résultat. Il y a donc une chance sur deux pour qu'un lancer soit isolé, une chance sur 4 pour qu'il soit le premier d'une suite de deux, etc. Ainsi la longueur moyenne d'une suite de jets consécutifs identiques est $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \dots = 2$.

Par conséquent, sur n jets, nous devons nous attendre en moyenne à $\frac{n}{2}$ séquences de jets consécutifs identiques. La probabilité d'une séquence d'exactly 10 jets consécutifs identiques étant $\frac{1}{2^{10}}$, la probabilité qu'il y en ait au moins une parmi $\frac{n}{2}$ séquences de longueur variable vaut $1 - \left(1 - \frac{1}{2^{10}}\right)^{n/2}$. Cette probabilité est supérieure à 0,5 dès que $n \geq 1419$.

L'exemple précédent n'est certainement pas destiné à des débutants. Il n'a pour but que de montrer l'utilité et même la nécessité de répéter certaines expériences un très grand nombre de fois si on veut faire apparaître les deux principales caractéristiques des phénomènes qui relèvent de la théorie des probabilités : une *régularité globale* qui permet de faire des prédictions à long terme, accompagnée d'une *variabilité locale* qui empêche toute prédiction à court terme et ne permet d'exclure l'apparition d'aucun événement même très peu probable.

Dans ce contexte, l'ordinateur est destiné à jouer un rôle particulièrement important en vue de *simuler* des expériences aléatoires. La *simulation* est la méthode qui accompagne nécessairement toute étude probabiliste expérimentale. Elle permet aussi de *valider* les résultats obtenus lors des études théoriques.

La simulation repose sur l'emploi d'un *générateur de nombres aléatoires*. Pour éviter toute mauvaise interprétation, il importe de faire comprendre en quoi consiste un tel générateur. Il s'agit — par exemple — de calculer une suite de nombres compris entre 0 et 1, de façon qu'ils soient uniformément répartis dans cet intervalle *comme le seraient des nombres qui auraient réellement été tirés au sort par un procédé mécanique quelconque*. Mais un ordinateur est absolument incapable de procéder à un véritable tirage au sort. Il ne peut « fabriquer » un nombre qu'à l'aide d'une procédure de calcul.

La génération d'une suite de nombres aléatoires résulte d'un procédé déterministe, seul le premier nombre de la suite étant éventuellement choisi (au hasard ?) par l'utilisateur. Les autres nombres sont calculés, chacun à partir du précédent, à l'aide d'une formule de récurrence bien choisie.

Il est facile de constater qu'il n'existe aucun générateur de nombres aléatoires qui soit parfait. En effet puisque un ordinateur ne peut traiter qu'un nombre fini de nombres, une répétition apparaît nécessairement lorsqu'on engendre une suite suffisamment longue de nombres par une formule de récurrence. La suite est donc périodique.

Certaines formules de récurrence conviennent bien pour engendrer une suite de nombres aléatoires. Nous entendons par là que la suite engendrée a une période tellement longue qu'on ne verra pas les répétitions apparaître et que les valeurs engendrées passent avec succès une série de tests d'uniformité de répartition. Dans [124], L. Råde décrit un générateur de nombres aléatoires facile à programmer et largement suffisant pour les besoins de l'enseignement :

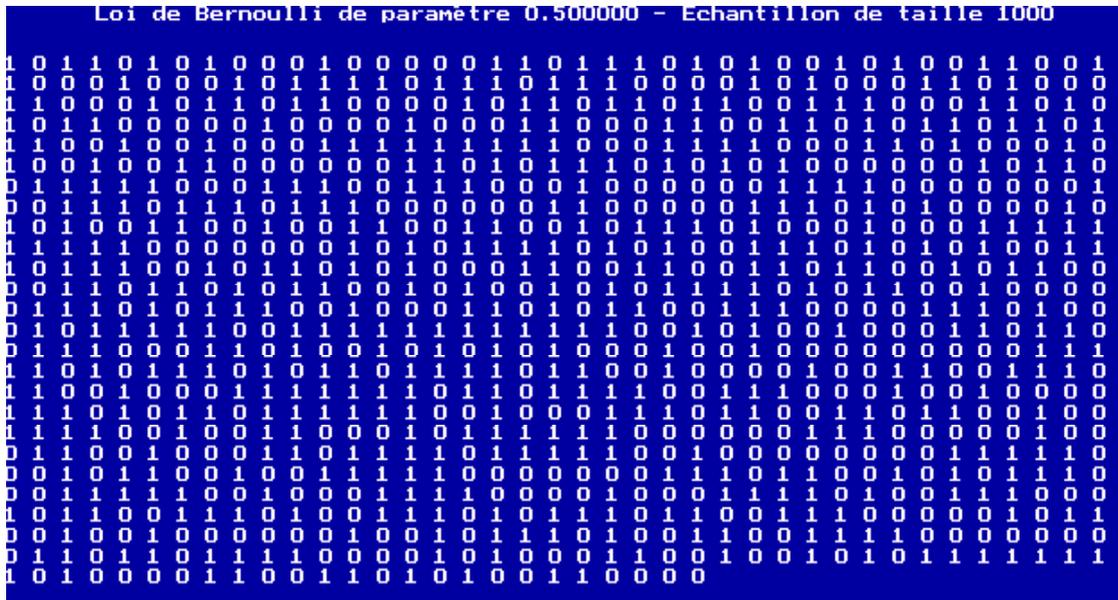
1. L'utilisateur choisit arbitrairement un nombre compris entre 0 et 1 et ayant au moins 6 à 7 décimales non nulles. Ce sera la *semence* du générateur, c'est-à-dire le premier nombre x_0 de la suite.
2. Quel que soit n , le nombre x_n est la partie décimale de $147x_{n-1}$.

Pour simuler un jeu de pile ou face, on engendre une suite de nombres par le procédé qui vient d'être indiqué et on décide que le n^{e} lancer est « pile » si $x_n \leq 0,5$, face sinon.

La programmation d'un générateur de nombres aléatoires peut aider à en faire comprendre le fonctionnement. Vu la nécessité lors d'une simulation de répéter un très grand nombre de fois un même calcul, l'utilisation d'un tableur ne permet pas d'arriver à des résultats acceptables : la prise en compte de plus d'une centaine de lignes ou de colonnes n'est guère possible. De plus tous les tableurs ne disposent pas d'un générateur de nombres aléatoires.

Il est alors nécessaire soit d'effectuer un peu de programmation dans un langage plus sophistiqué, soit d'utiliser un logiciel adéquat.

EXEMPLE 6.5.9 *Des logiciels existent qui permettent de simuler le lancer d'une pièce de monnaie, d'un dé ou plus généralement d'événements aléatoires obéissant aux grandes lois classiques : de Bernoulli, binômiale, de Poisson, etc.*



Ces logiciels comptent le nombre d'occurrences d'un événement donné, dressent des tableaux, dessinent divers graphiques, etc.

Dans bien des cas, la simulation d'une expérience nécessite une analyse soignée de la situation. Rédiger un programme de simulation est une *modélisation numérique*. Le programme doit donc être validé, et cela doit être fait avec d'autant plus de soin que nous connaissons la facilité avec laquelle des erreurs peuvent être commises en probabilité.

EXEMPLE 6.5.10 *Pour illustrer cette remarque, reprenons le problème 6.3.3 du bâton brisé en trois morceaux de longueurs aléatoires. Simulons ce problème de la façon suivante :*

1. Nous admettons que le bâton est le segment $[0, 1]$. Le couper aléatoirement en trois morceaux revient donc à choisir deux nombres x et y compris entre 0 et 1.
2. Nous convenons d'appeler x le plus petit des deux nombres et y le plus grand.
3. La fonction $\text{Alea}(a, b)$ nous fournit un nombre aléatoirement entre a et b . Nous engendrons donc x par un appel à $\text{Alea}(0, 1)$ puis y par un appel à $\text{Alea}(x, 1)$.
4. Les trois morceaux du bâton sont de longueur x , $y - x$ et $1 - y$. Nous testons donc les inégalités $x < 1 - x$, $y - x < x + 1 - y$ et $1 - y < y$, soit $x < \frac{1}{2}$, $y - x < \frac{1}{2}$ et $y > \frac{1}{2}$.

Nous appliquons l'algorithme précédent un grand nombre de fois, et nous comptons combien de fois les trois inégalités sont satisfaites. Nous savons par l'analyse géométrique effectuée précédemment que cette fréquence doit être raisonnablement proche de 0,25. Or en exécutant le programme correspondant à cet algorithme, nous trouvons systématiquement des valeurs proches de 0,19.

Nous nous trouvons donc en présence de deux modélisations qui donnent des résultats différents. Une d'entre elles n'est pas valide. Un peu de réflexion montre que notre simulation n'a pas été effectuée correctement : le choix de y devrait être indépendant de celui de x alors qu'il ne l'est visiblement pas. Les valeurs de y ne sont pas uniformément réparties entre 0 et 1. Nous devons modifier la procédure appliquée : engendrer y comme x par un appel à $\text{Alea}(0, 1)$ puis permuter éventuellement les valeurs de x et y de façon que x soit le plus petit des deux.

Nous voyons ici apparaître l'utilité dans certains cas de modéliser une situation de plusieurs façons différentes. Ces modélisations se valident mutuellement et mettent l'accent sur des aspects différents du problème étudié, enrichissant ainsi considérablement les compétences conceptuelles et procédurales de l'élève.

6.5.5 Numériser la géométrie

Nous concluons cette longue section consacrée à l'emploi des moyens de calcul en examinant les diverses possibilités de numériser la géométrie.

Quelle que soit la présentation utilisée, un problème de géométrie ne peut être traité par ordinateur qu'à travers une algébrisation. Ce sont donc deux changements de cadre successifs qui sont opérés. Mais les logiciels peuvent masquer totalement ou partiellement aux yeux de l'utilisateur le fait que l'ordinateur ne manipule que des informations numériques.

EXEMPLE 6.5.11 Lorsque Cabri-Géomètre teste si trois droites passent par un même point, il détermine en réalité si un nombre réel est nul, c'est-à-dire inférieur à un certain ε . Le choix de cet ε comporte une large part d'arbitraire et peut varier d'une machine à l'autre.

Contentons-nous de mentionner trois types de logiciels géométriques (par ailleurs largement connus) pouvant intéresser l'enseignement des mathématiques.

6.5.5.1 Les logiciels de type Cabri-Géomètre

Nous disons « LES logiciels ... » car il en existe plusieurs. A côté de Cabri-Géomètre (dû pour l'essentiel à Jean-Marie Laborde), qui est sans doute à la fois le premier, le plus connu et le plus complet d'entre eux, mentionnons, disponibles sur Internet : Dr. Geo ⁽⁴⁾ et Décllic ⁽⁵⁾. Ces logiciels sont adaptés à la géométrie synthétique, le passage par le cadre algébrique étant invisible pour l'utilisateur.

Ce sont d'abord des outils permettant la construction aisée de figures mêmes relativement complexes. Ils contribuent ainsi à *procéduraliser la géométrie*. Le fait de pouvoir construire de façon assez précise et en un temps limité des figures de qualité est de nature à aider puissamment à la formation des concepts, notamment en dégageant les propriétés qui justifient les constructions. L'apprentissage de la démonstration ne peut qu'en bénéficier.

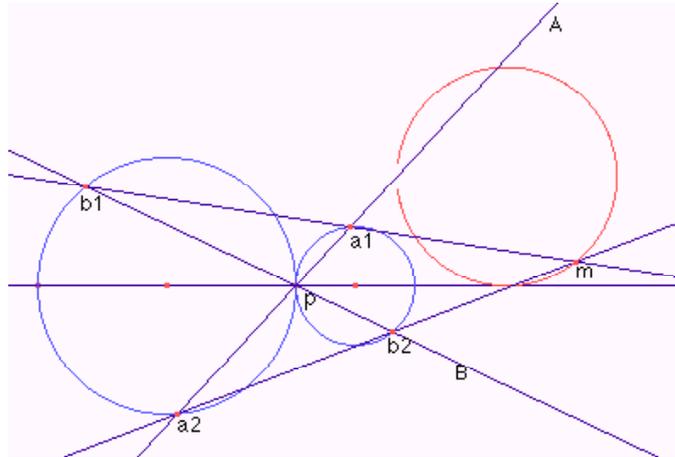
⁽⁴⁾ <http://xoom.com/FeYiLai>

⁽⁵⁾ <http://home.nordnet.fr/eostenne/declic.htm>

L'aspect le plus spectaculaire de ces logiciels est la possibilité de modifier une figure tout simplement en saisissant un point à l'aide du curseur de la souris et en le déplaçant sur l'écran. On peut ainsi se convaincre qu'une propriété qui semble apparaître sur une figure est bien vraie car indépendante de la figure particulière qui a été construite. Les démonstrations restent nécessaires car elles permettent de montrer *pourquoi* les propriétés sont vraies.

Une autre caractéristique remarquable de ces logiciels est la construction de lieux géométriques dont l'élève ignore la nature. L'ayant découverte sur l'écran, il sait désormais comment il doit orienter son raisonnement.

EXEMPLE 6.5.12 *Deux cercles sont tangents en un point p par lequel passent deux droites A , qui recoupe les cercles en a_1 et a_2 , et B , qui recoupe les cercles en b_1 et b_2 . On note m le point d'intersection des droites a_1b_1 et a_2b_2 . Quel est le lieu de m lorsque B tourne autour de p , la droite A restant fixe ?*



Sans Cabri, rares sont les élèves (et même les professeurs) qui affirmeront que le lieu est un cercle. Il reste à trouver une démonstration.

Pour plus d'exemples d'utilisation de Cabri-Géomètre, nous nous permettrons de renvoyer à la littérature existante, notamment les manuels d'accompagnement du logiciel, les textes écrits par ses auteurs, les articles [53], [54], [55], [56] dûs à Francis Denis et Sylvain Courtois parus dans *Mathématique et Pédagogie*, les deux volumes [60] de R. Cuppens, le [site Internet http://www-cabri.imag.fr](http://www-cabri.imag.fr) consacré à Cabri et la revue bimestrielle *Abracadabri*.

6.5.5.2 Le Logo

Le langage Logo a été créé vers la fin des années 60 par Seymour Papert (voir [113]). Il permet également une approche procédurale de la géométrie. Mais les propriétés qu'il permet de mettre en évidence ne sont pas nécessairement les mêmes que celles que l'on rencontre traditionnellement. A tout moment, l'état de la tortue est déterminé par deux paramètres : sa position et son cap.

EXEMPLE 6.5.13 En LOGO, on dessine un cercle à l'aide d'une instruction analogue à la suivante : REPETE 30 [AV 10 DR 12]

Autrement dit, nous ordonnons à la tortue d'effectuer 30 fois une suite de deux opérations : avancer de 10 unités, tourner à droite de 12 degrés. Nous savons qu'une telle instruction ne saurait dessiner un cercle mais bien un polygone régulier à 30 côtés. Mais ce polygone est suffisamment proche d'un cercle pour que la différence ne puisse être perçue à l'œil nu. C'est donc bien un cercle approché (tout résultat produit par un ordinateur est approché). Mais ce cercle n'est pas construit à l'aide de la propriété courante : un cercle est l'ensemble des points à distance donnée d'un point donné. Ni le centre, ni le rayon du cercle ne sont utiles pour la tortue qui obéit à l'instruction ci-dessus. La tortue ne connaît que la longueur de chaque « côté du cercle » et l'angle dont elle tourne à chaque pas. Ces deux paramètres déterminent la courbure de la courbe dessinée.

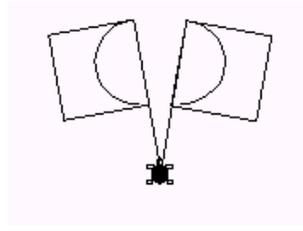
En Logo un cercle est construit comme étant une courbe à courbure constante.

A tout moment, la tortue avance un peu et tourne un peu. C'est une géométrie *locale* qui la gouverne. La programmation Logo se prête donc bien à l'étude de propriétés locales. Elle pourrait par exemple être utilisée pour dessiner des courbes solutions d'une équation différentielle $y' = f(x, y)$, sans résoudre l'équation.

Puisqu'elle permet de rencontrer des propriétés différentes des propriétés usuelles, la « géométrie-tortue » constitue un outil qui complète les outils géométriques usuels sans les remplacer. En permettant un certain dépaysement, elle peut relancer la motivation. Ce serait une erreur de la confiner — comme on le fait souvent — à l'école primaire.

Elle a aussi une autre caractéristique : en programmant en Logo, l'élève a réellement une activité de type informatique. Il doit rédiger des routines et des sous-routines, mettre en place des itérations, exploiter éventuellement la récursivité. Il peut aussi remarquer comment certaines propriétés d'une figure peuvent se traduire au niveau du programme Logo.

EXEMPLE 6.5.14 Si une programme LOGO dessine une figure F , pour dessiner l'image de F par une symétrie axiale, il suffit de remplacer partout l'instruction DR (tourne à droite) par l'instruction GA (tourne à gauche) et vice-versa. L'axe de la symétrie est déterminé par la position et le cap de la tortue au départ du programme et il faut bien entendu ramener la tortue à ce point entre les deux exécutions.



Cette approche informatique met bien en évidence les invariants d'une symétrie axiale. De plus, en Logo, les angles sont nécessairement orientés.

Nous pouvons aussi obtenir, à peu de frais, des figures dont le côté esthétique ne peut être négligé et qui possèdent une symétrie circulaire prégnante. Imaginons une tortue situé au point P_0 , avec un cap α et considérons une séquence d'instructions Logo qui amène la tortue en un point P_1 et avec le cap $\alpha + 60$ (tous les angles sont mesurés en degrés). Que se passe-t-il si nous exécutons la séquence d'instructions une deuxième fois, la tortue repartant de P_1 ? Elle occupera alors un point P_2 , avec le cap $\alpha + 120$. Après une troisième exécution, elle sera en P_3 avec le cap $\alpha + 180$. Continuant de la sorte, nous voyons qu'après six exécutions de la séquence d'instructions, la tortue sera en un point P_6 , avec le cap α .

Mais tous les segments $[P_0P_1]$, $[P_1P_2]$, \dots , $[P_5P_6]$ ont même longueur. Il est alors clair que $P_0 = P_6$ et que les points P_0, P_1, \dots, P_5 sont les sommets d'un hexagone régulier. Et cela quelle que soit la séquence d'instruction exécutée (la seule condition étant qu'elle augmente le cap de 60°).

EXEMPLE 6.5.15 Considérons le programme suivant :

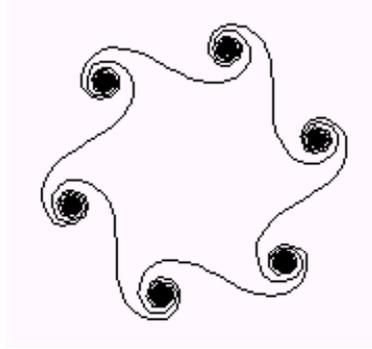
```

POUR ROSACE :L :A
REPETE 120 [AV :L TG :A DONNE "A :A+3]
FIN

```

Exécutons *ROSACE* 5 1 . Le programme ne comporte qu'une seule instruction d'avancer, *AV :L*, dans laquelle *:L* aura toujours 5 pour valeur et une seule instruction de tourner à gauche *TG :A* dans laquelle *:A* vaut au départ 1° . Mais ces deux instructions sont exécutées 120 fois, et à chaque fois la valeur de *:A* augmente de 3° .

A l'issue du programme, la valeur de $\angle A$ a augmenté de 360° , autrement dit elle est redevenue de 1° (ce qui permettra un enchaînement harmonieux avec la seconde exécution de *ROSACE 5 1*). Quant à la tortue, elle a tourné de $1 + 4 + 7 + \dots + 357 = 21480$ degrés, soit 300° vers la gauche, ou encore 60° vers la droite. En enchaînant 6 exécutions, (*REPETE 6 [ROSACE 5 1]*) on obtient par conséquent une figure qui a nécessairement une symétrie d'ordre 6, comme le confirme la figure suivante :



Le fait que les caractéristiques géométriques intéressantes d'une figure se retrouvent dans un programme qui dessine cette figure n'est en soi pas spécialement étonnant, mais cela rend les changements de cadre considérés d'autant plus fructueux.

On trouvera d'autres exemples d'utilisation du Logo dans la brochure *La géométrie hors des sentiers battus*, [51], publiée en 1988 par le service de l'Organisation des Etudes du Ministère de l'Education Nationale. (Voir aussi [109]).

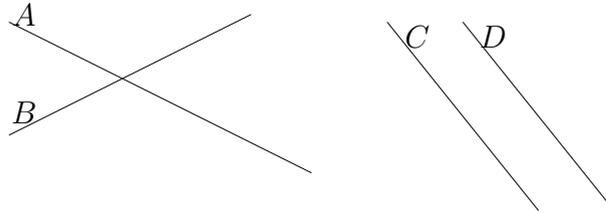
6.5.5.3 La représentation de l'espace

Un autre domaine géométrique pour lequel il nous semble important de rappeler l'intérêt d'un traitement informatique est celui de la représentation plane des figures de l'espace. Un des gros problèmes rencontrés par les élèves est en effet de « voir dans l'espace ». Cette expression ne doit évidemment pas être prise au pied de la lettre : les élèves (ou les adultes) qui « ne voient pas dans l'espace » sont parfaitement capables de monter dans un autobus sans rater la marche.

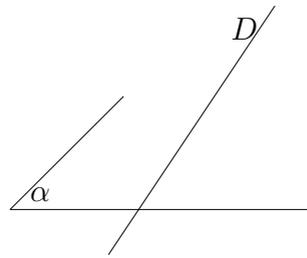
« Ne pas voir dans l'espace » signifie « Ne pas être capable de reconstituer mentalement la réalité spatiale à partir d'une de ses représentations planes ».

La raison de cette non-reconstitution est simple : dans n'importe quelle représentation plane de l'espace, chaque point du plan représente une infinité de points de l'espace. Toute représentation plane est donc ambiguë. Malheureusement, bien des figures « simples » utilisées traditionnellement dans les débuts de l'étude de l'espace ne permettent pas de lever cette ambiguïté et en conséquence ne permettent pas non plus de « voir » la situation.

EXEMPLE 6.5.16 *La figure située ci-dessous à gauche représente aussi bien deux droites non coplanaires que deux droites sécantes, celle de droite représente aussi bien deux droites non coplanaires que deux droites parallèles.*

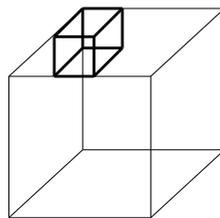


Quant à la figure qui suit, elle représente un plan et une droite qui coupe ce plan. Mais où est le point d'intersection ?



Des figures telles que les précédentes sont ce que nous appelons des *figures flottantes* : les éléments qui constituent la figure ne sont accrochés ni entre eux, ni à un référentiel. Un premier moyen d'aider à « voir dans l'espace » est de bannir les figures flottantes et de ne considérer que des figures dont les éléments sont « accrochés » à des objets géométriques familiers tels qu'un polyèdre, celui-ci étant évidemment très souvent un cube. Mais même dans ce cas, des ambiguïtés peuvent empêcher l'élève de reconstituer mentalement la figure spatiale.

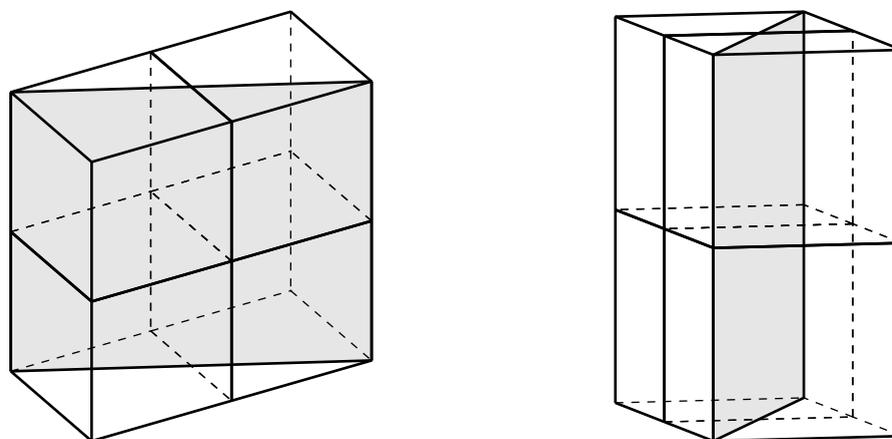
EXEMPLE 6.5.17 *La figure ci-dessous comporte un grand cube et un petit cube. Le petit cube est-il situé au-dessus ou à l'intérieur du grand ?*



Pour rencontrer des problèmes de ce genre, l'école primaire fait en sorte que l'élève puisse prendre en main un objet ayant la forme voulue et le manipuler à sa guise. Ce type d'activité est encore réalisable au début du secondaire, mais devient de plus en plus difficile au fur et à mesure que les situations rencontrées deviennent plus complexes. Un logiciel de géométrie spatiale doit donner aux élèves la possibilité non seulement de construire des figures accrochées à des référentiels, mais aussi d'examiner ces constructions sous divers angles de vue. En faisant tourner les objets spatiaux à l'écran autour de divers axes, on facilite la reconstitution mentale de la situation.

Divers logiciels existent qui permettent d'atteindre cet objectif. On nous pardonnera de mentionner le logiciel **Reseau** réalisé par les auteurs du présent document durant l'année 1996-97, dans le cadre d'une recherche portant sur l'enseignement de l'algèbre linéaire au troisième degré du secondaire (voir [28] ⁽⁶⁾). Ce logiciel permet de créer divers objets géométriques en exploitant les possibilités d'un réseau cubique, et de représenter ces objets non seulement sous n'importe quel angle de vue, mais aussi selon trois systèmes différents de représentation spatiale : la perspective cavalière, la projection orthogonale et la perspective centrale.

EXEMPLE 6.5.18 Les figures ci-dessous ont été construites à l'aide du logiciel RESEAU. Elles présentent la même situation sous deux angles de vue différents.



⁽⁶⁾ Le rapport complet de cette recherche devrait être publié prochainement par les services de l'Organisation des Études