L'ADDITION EST-ELLE COMMUTATIVE ?

Jean LEFORT
Irem de Strasbourg

Introduction : Un résultat original.

C'est un titre en forme de boutade, bien sûr. Mais à l'heure du développement du calcul symbolique il me semble important de comprendre ce que fait une calculatrice genre TI 89 ou un logiciel tel que Derive ou Maple (et bien d'autres que je ne manipule pas). Evidemment si je demande à la machine, à travers le logiciel qu'elle utilise, **3 + 2** ou **2 + 3** j'obtiendrai la même réponse **5** et c'est heureux. Mais dès que l'on utilise des outils un peu plus sophistiqués il se passe des choses étranges.

Par exemple, quand les élèves abordent les suites, ils apprennent à calculer la somme des termes dans le cas particulier des suites arithmétiques ou géométriques. Ils découvrent très vite, ou le professeur leur fait découvrir, l'usage de la commande qui correspond à la notation mathématique $\sum_{n=a}^{b} f(n)$ (sans que cette notation soit enseignée). Dans le cadre de Maple, que j'utilise ici, la syntaxe est :

> **sum(f(n),n=a..b);**

$$\sum_{n=a}^{b} f(n)$$

Cette commande est bien pratique car elle permet de ne pas apprendre les formules donnant la somme des termes d'une suite arithmétique ou géométrique. En effet, on a :

> **A:=sum(a*x+b,x=0..n);**
G:=sum(k*a^x,x=0..n);

$$A := \frac{a\,(n+1)^2}{2} - \frac{a\,(n+1)}{2} + b\,(n+1)$$

$$G := \frac{k\,a^{(n+1)}}{a-1} - \frac{k}{a-1}$$

Ceci prouve que les logiciels savent faire beaucoup de choses pour la plus grande satisfac-

tion des utilisateurs qui n'ont plus à passer des heures pour retrouver des formules beaucoup moins usuelles que celles que je viens de prendre comme exemple.

Seulement, voilà, cet avantage se paye ailleurs avec un résultat pour le moins troublant. Nous allons tout simplement calculer la somme d'entiers successifs. Cela n'est pas bien méchant mais pourtant le logiciel va fournir des réponses pour le moins originales.

> **sum(n,n=0..10); sum(n,n=1..10);**

55

55

Rien que de très normal.

> **sum(n,n=10..0); sum(n,n=10..1);**

– 45

– 44

Non seulement la somme n'est pas commutative, mais encore 0 n'est pas élément neutre et obtenir un résultat négatif en additionnant des nombres positifs, il y a quelque chose qui ne va pas. Et pourtant le logiciel avait l'air d'être très savant ! Faut-il tout jeter ? Certes pas, mais plutôt essayer de comprendre.

Un logiciel très savant

En fait le logiciel est très savant. Quand on lui demande $\sum_{n=a}^{b} \mathrm{f}(n)$, il commence par chercher une fonction F telle que $F(n+1) - F(n) = f(n)$. Il existe pour cela un certain nombre de méthodes qui s'apparentent à la recherche de primitives (cf. section suivante). Quand il a trouvé une telle fonction, il calcule tout simplement $F(b+1) - F(a)$ qui vaut très exactement $f(a) + f(a+1) + \ldots + f(b)$. Il est très facile de voir si le logiciel a trouvé cette fonction F. Il suffit pour cela de lui demander

> **sum(n,n=0..x–1);**

$$\frac{1}{2}x^2 - \frac{1}{2}x$$

Dans le cas présent, c'est-à-dire pour la somme des entiers successifs, la fonction F est la fonction qui a x associe $\frac{x(x-1)}{2}$. Mais nous voyons immédiatement que si dans l'expression $\sum_{n=a}^{b} \mathrm{f}(n)$ les bornes a et b semblent jouer le même rôle, il n'en est pas de même dans l'expression $F(b+1) - F(a)$. Ceci explique que si on intervertit les bornes, le logiciel fournit la réponse $F(a+1) - F(b)$ qui n'a aucune raison d'être égale à la réponse précédente. Et c'est d'ailleurs bien ce que nous avons constaté.

> **F:=x->x*(x-1)/2:**
> **'F(11)-F(0)'=F(11)-F(0);**
> **'F(1)-F(10)'=F(1)-F(10);**
> **'F(2)-F(10)'=F(2)-F(10);**

$$\mathrm{F}(11) - \mathrm{F}(10) = 55$$

$$\mathrm{F}(1) - \mathrm{F}(10) = -45$$

$$\mathrm{F}(2) - \mathrm{F}(10) = -44$$

Nous retrouvons bien les résultats donnés par la commande **sum** . Mais qu'en est-il si le logiciel ne trouve pas une telle fonction F ?

Bien sûr, comme pour les primitives, une telle fonction existe toujours ce qui ne veut pas dire qu'elle s'exprime avec les fonctions élémentaires ou même les fonctions connues du logiciel. Voyons un exemple.

> **sum(1/(cos(x)^2+1),x=0..n);**

$$\sum_{x=0}^{n} \frac{1}{\cos(x)^2+1}$$

Le logiciel renvoyant la formule d'origine, c'est que le logiciel ne trouve pas, directement, une réponse plus compacte. Regardons alors ce qui se passe pour des valeurs numériques :

> **S1:=sum(1/(cos(x)^2+1),x=0..4);**

$$S1 := \frac{1}{2} + \frac{1}{\cos(1)^2+1} + \frac{1}{\cos(2)^2+1} + \frac{1}{\cos(3)^2+1} + \frac{1}{\cos(4)^2+1}$$

> **S2:=sum(1/(cos(x)^2+1),x=4..0);**

$$S2 := -\frac{1}{\cos(1)^2+1} - \frac{1}{\cos(2)^2+1} - \frac{1}{\cos(3)^2+1}$$

Le logiciel reste très logique puisque s'il existe une fonction F alors

$$S1 = F(5) - F(0) = f(0) + f(1) + f(2) + f(3) + f(4)$$

tandis que

$$S2 = F(1) - F(4) = F(0) - F(5) + (F(1) - F(0)) + (F(5) - F(4)) = = -S1 + f(0) + f(4)$$

ce qui est bien le résultat annoncé pour $S2$.

Une autre façon de faire

Il existe, dans Maple, une autre commande qui ajoute les termes successifs. C'est la commande **add**. Mais avec celle-ci les résultats sont encore plus brutaux. D'une part cette commande ne s'exécute que pour des valeurs numériques et d'autre part elle renvoie la valeur 0 dès que la deuxième borne est inférieure à la première.

> **add(n,n=0..10);**

55

> **add(n,n=10..0);**

0

Cela provient du fait que cette commande n'est rien d'autre qu'une boucle. Elle ne s'exécute donc pas si dès le premier test la valeur de la variable est supérieure à la valeur finale, ce qui est le cas dans le deuxième exemple.

Comparaison du discret et du continu

Nous venons de voir la logique de la commande **sum** . Cette logique peut facilement être comparée à celle du calcul intégral. En effet, si F est une primitive de f cela signifie que $f(x) = \lim_{h \to 0} \frac{F(x+h) - F(x)}{h}$ que l'on peut comparer à la formule vue ci-dessus et qui peut s'écrire $f(n) = \frac{F(n+1) - F(n)}{1}$. D'une certaine façon, sur les entiers non nuls on ne peut pas faire plus petit que 1, tandis que sur les réels, h peut vraiment être aussi proche de 0 que l'on veut.

Quand on calcule une intégrale définie par une méthode approchée, par exemple celle des rectangles, on effectue une somme finie de termes. Cette somme finie doit néanmoins répondre à la relation $\int_a^b f(t)\mathrm{d}t = -\int_b^a f(t)\mathrm{d}t$

aux approximations près. Ceci nous permet de mieux comprendre l'apparition d'un résultat négatif avec une somme de termes positifs. Avec cette réflexion, nous nous attendons alors à trouver que $\sum_{n=a}^{b} f(n) = -\sum_{n=b}^{a} f(n)$. Et pourtant ce n'est pas exactement ce résultat que nous avons. L'écart qui subsiste provient des bords.

Reprenons l'exemple du calcul approché de $\int_a^b f(t)\mathrm{d}t$ à l'aide de la méthode des rectangles. Posons $h = \left|\frac{b-a}{n}\right|$. Si $a < b$ nous calculons, par exemple:

$$R1 = h(f(a) + f(a+h) + \ldots + f(b-h)).$$

Mais si nous échangeons le rôle de a et b ce serait plutôt :

$$R2 = h(f(b) + f(b-h) + \ldots + f(a+h)).$$

Nous voyons bien qu'il y a un léger écart en valeur absolue, écart égal à $|f(b)-f(a)|.|h|$. Dans le cadre de l'intégration, selon le sens des bornes, dans un calcul approché par la méthode des rectangles on fait apparaître une borne ou l'autre.

C'est quelque chose d'analogue qui a lieu avec le calcul d'une somme, mais cette fois ci ce sont les deux bornes qui interviennent ou non, comme nous l'avons vu dans les sections précédentes.

En effet, si $\sum_{n=a}^{b} f(n) = F(b+1) - F(a)$ et $\sum_{n=b}^{a} f(n) = F(a+1) - F(b)$ alors on obtient :

$$\sum_{n=a}^{b} f(n) = -\sum_{n=b}^{a} f(n) + F(a+1) - F(a) + F(b+1) - F(b)$$
$$= -\sum_{n=b}^{a} f(n) + f(\mathrm{a}) + f(b).$$

Bien sûr dans la méthode des rectangles, l'écart tend vers 0 avec h tandis que pour des sommes, on ne peut pas descendre en dessous de 1 et les bornes restent apparentes. Nous n'avons affaire qu'à une analogie et non pas à un décalque.

Conclusion

L'analogie entre le discret et le continu peut être prolongée sur bien des points comme l'intégration par parties qui a son pendant dans les sommations. Mais ceci est un autre sujet.

Ce qu'il faut retenir, c'est qu'une machine ne fait que ce pour quoi elle a été programmée. Si dans la majorité des cas le résultat est conforme à nos habitudes, ce n'est pas systématiquement vrai. Les concepteurs de logiciel sont amenés à faire des choix qui sont des choix d'informaticien et non pas de mathématicien. Nous venons de voir un exemple et il y en a bien d'autres comme le calcul des racines carrées ou le calcul intégral (voir l'article *Le calcul formel dans l'enseignement des mathématiques* par Michel MIZONY, *in* Repères, n°62, janvier 2006).

Quand on utilise une machine qui fait du calcul symbolique il y a intérêt à comprendre le principe des algorithmes utilisés. Ils sont en général explicités dans l'aide. Cela évite bien des erreurs qui, si elles sautent aux yeux dans un calcul simple, risquent de passer inaperçues au sein d'une procédure ou dans un calcul un peu long. D'une façon générale, se rappeler que les logiciels implantés dans ces machines ont une conception bien à eux des mathématiques.