Quelques problèmes de calculs liés au système binaire

Jean-François Colonna
Daniel Justens

Notre système de calcul est décimal, conséquence historique et biologique de nos deux mains à cinq doigts. Mais la machine travaille en base deux, qui permet une représentation physique des nombres par le passage (1) ou non (0) d'une impulsion électrique. Le système binaire permet la transition vers le système octal (base 8) qui synthétise toute suite de trois chiffres « binaires » en un seul. Théoriquement, tout nombre peut s'écrire indifféremment dans chaque système de comptage. Mais en pratique il est loin d'en être ainsi. En effet, le fait pour un nombre de se présenter ou non sous une écriture finie va impliquer l'existence ou non d'un certain niveau d'erreur lors des différentes opérations à effectuer. Face aux nombres irrationnels tous les systèmes semblent égaux : ces nombres ne sont jamais qu'approchés ! Mais il n'en va pas de même pour les nombres rationnels qui s'exprimeront de manière exacte ou non en fonction de la base choisie. Un changement de base implique alors une modification parfois significative du niveau d'erreur lors d'une succession d'opérations arithmétiques. Alors que dire des opérateurs non arithmétiques, tous approchés linéairement ?

Notre confiance en la machine doit être adaptée en conséquence. Et parfois mise en doute. En voici un petit exemple. Considérons la suite de calculs établis comme suit. Soit a un nombre réel. Définissons $b = a+1$ et $x_1 = 1$. Et calculons $x_2 = b\, x_1 - a$. On vérifie que $x_2 = (a+1)\,.1 - a = 1$. Selon la même procédure, on peut calculer x_3, x_4, ... , x_n. Et vérifier que cette suite est composée uniformément de « 1 » quel que soit le nombre a choisi au départ. Encodons à présent notre procédure dans un tableur (excel ou open office) et testons quelques valeurs particulières « bien choisie » pour a. (Tous les fichiers de calculs sont disponibles sur notre site infinimath).

a =	1023,1
b=a+1	1024,1

x =bx-a
1,00000000000000000000
0,99999999999988600000
0,99999999988347100000
0,99999988066258500000
0,99987778655361100000
0,87484120955321000000
-127,17511729655700000000
-131263,13762340400000000000
-134427602,34012900000000000000
-137667308579,62600000000000000000
-140985090717418,00000000000000000000
-144382831403708000,00000000000000000000
-147862457640538000000,00000000000000000000
-151425942869675000000000,00000000000000000000
-155075308092834000000000000,00000000000000000000
-158812623017871000000000000000,00000000000000000000
-162640007232602000000000000000000,00000000000000000000
-166559631406908000000000000000000000,00000000000000000000
-170573718523814000000000000000000000000,00000000000000000000
-174684545140238000000000000000000000000000,00000000000000000000

La plupart des inputs « a » donnent bien la suite de « 1 » attendue. Mais que se passe-t-il lorsque l'on introduit a = 1023,1 ? On arrive à la suite de valeurs présentées dans le tableau ci contre, qui est bien loin de ce que l'on doit trouver.

Analysons ce résultat étonnant. Le choix de 20 décimales est éclairant. Le nombre de zéros situés à droite des premières itérations l'est également : il montre que le système travaille « en *double* précision », c'est-à-dire avec *deux* fois 8 chiffres significatifs (exacts). Ceci peut se vérifier également en calculant la suite de nombres

$$(10^n + 1) - 10^n,$$

comme il est fait dans le second tableau. Dès la valeur n=15, une erreur apparaît, limitant ainsi la précision de toute opération.

n=	10^n +1	10^n	(10^n+1) - 10^n
1	11	10	1
2	101	100	1
3	1001	1000	1
4	10001	10000	1
5	100001	100000	1
6	1000001	1000000	1
7	10000001	10000000	1
8	100000001	100000000	1
9	1000000001	1000000000	1
10	10000000001	10000000000	1
11	100000000001	100000000000	1
12	1000000000001	1000000000000	1
13	10000000000001	10000000000000	1
14	100000000000001	100000000000000	1
15	1000000000000000	1000000000000000	0
16	10000000000000000	10000000000000000	0
17	100000000000000000	100000000000000000	0
18	1000000000000000000	1000000000000000000	0
19	10000000000000000000	10000000000000000000	0
20	100000000000000000000	100000000000000000000	0

Certes, dans la plupart des applications, ce niveau de précision est suffisant. Mais ce n'est pas le cas dans une suite de calculs qui diverge très rapidement et dans laquelle une erreur même extrêmement petite sur la valeur de l'un des x de la suite est amplifiée systématiquement. Quelle est donc la nature bien particulière du nombre 1023,1 qui conduit à ces résultats aberrants ? On obtient en fait ce type de suites de valeurs absurdes pour tout a de la forme $4^k - 0{,}9$ ou encore $2^{2k} - 0{,}9$, k étant un nombre naturel. Le lecteur curieux peut ainsi tester 3,1 ; 15,1 ; 63,1 et ainsi de suite, pour constater que lorsque k croît, la divergence devient plus rapide. Notre suite est construite de telle sorte que toute différence initiale, si petite soit-elle, va s'amplifier de manière exponentielle, vu la multiplication introduite dans la procédure. Les nombres 2^{2k} s'écrivent de manière exacte en base 2. Qu'en est-il de la fraction 9/10 ? Un calcul simple montre que 0,9 s'écrit 0,111001100110011... présentant bien une forme périodique illimitée (voir nos fichiers de calculs). Que se passe-t-il lorsque l'on travaille en double précision ? On connaît bien l'approximation qui est souvent faite et qui dit que $2^{10} \approx 10^3$ avec une erreur de l'ordre de 2%. Pour obtenir une précision de l'ordre de 15 ou 16 décimales (ce qui sémantiquement implique la base 10), il faut, en base 2, aller jusqu'à une cinquantaine de chiffres « après la virgule ». Les puissances de 1/2 donnent en fait les erreurs absolues maximales en base deux. Elles sont calculées, respectivement à la 16^{e} et 17^{e} décimales, dans le tableau ci-contre qui donne également les erreurs absolues pour la valeur 0,9. On constate que ces erreurs sont maximales étant, pour une précision de 10^{-k}, approximativement égales à $1/2\ 10^{-k}$ (et même un peu supérieures) pour les valeurs $k = 16$ et $k = 17$. De là les résultats étonnant obtenus pour toute succession de calculs de nature divergente.

puissances de 1/2	Valeur décimale	erreur en base 2
50	8,88E-016	5,55E-016
51	4,44E-016	0,00000000000000011102
52	2,22E-016	
53	1,11E-016	
54	0,00000000000000005551	5,55E-017
55	0,00000000000000002776	0,00000000000000002776

Des problèmes vont donc également se poser lors de la mise en place de tous calculs générateurs de chaos mathématique. Un deuxième exemple va nous montrer explicitement que les propriétés de distributivité, d'associativité ou de commutativité ne sont pas vérifiées lors de la mise en place de calculs effectifs utilisant la machine. Pour cela, nous allons introduire la procédure chaotique connue sous l'appellation de « dynamique de Verhulst », selon le nom du célèbre mathématicien belge. Le but du mathématicien était de décrire l'évolution d'une population sous contrainte, de manière à échapper à l'absurdité du modèle exponentiel manquant de réalisme. Voyons ce qu'il en est en temps discret. Soit une population y mesurée à l'instant t (appartenant à N) : y_t. L'instant suivant, cette population va croître proportionnellement à elle-même mais en subissant également un facteur d'atténuation expliqué par la limitation des ressources et la saturation de sa niche écologique. Verhulst étudie alors une procédure évolutive selon l'équation :

$$y_{t+1} - y_t = a \times y_t \times \left(1 - \frac{y_t}{K}\right)$$

Cette équation exprime une croissance proportionnelle à elle-même (exponentielle) mais atténuée par un facteur proportionnel à la taille de la population. La constante K est supposée rendre compte des conditions naturelles propres à l'espèce concernée. Ceci n'est pas notre sujet. Concentrons-nous sur l'aspect formel de l'équation qui peut s'écrire en posant $K = 1$:

$$y_{t+1} = y_t + a\ y_t - \frac{a}{K}\ y_t^2 = y_t + a\ y_t - a\ y_t^2$$

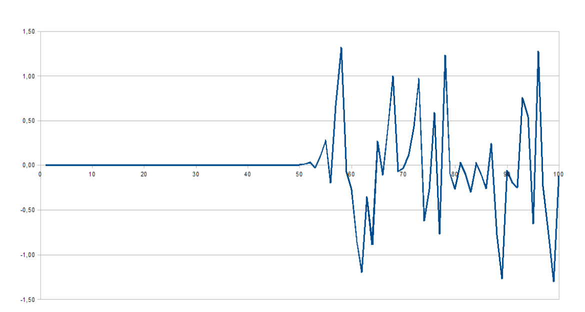
On peut donner plusieurs notations formellement identiques de cette expression en utilisant les propriétés de commutativité, d'associativité et de distributivité dans les réels. Mais qu'en est-il des valeurs numériques obtenues ? Examinons et comparons les résultats obtenus pour les valeurs a=3 en partant de x_0=0,5 en donnant à notre expression les écritures formellement équivalentes :

$$y_{t+1} = (a+1)\, y_t - a \times (y_t^2) \quad \text{et} \quad y_{t+1} = (a+1)\, y_t - (a\, y_t) \times y_t$$

On obtient le tableau de valeurs ci-contre. De petites erreurs apparaissent dès la dixième itération, montrant ainsi la non associativité dans les calculs informatiques.

Dynamique de Verhulst			
a =	3,00000000000000000000		
	(a+1) yt – a yt^2	(a+1) yt – (ayt) . yt	Différences
x =	0,50000000000000000000	0,50000000000000000000	0,00000000000000000000
1	1,25000000000000000000	1,25000000000000000000	0,00000000000000000000
2	0,31250000000000000000	0,31250000000000000000	0,00000000000000000000
3	0,95703125000000000000	0,95703125000000000000	0,00000000000000000000
4	1,08039855957031000000	1,08039855957031000000	0,00000000000000000000
5	0,81981109571643200000	0,81981109571643200000	0,00000000000000000000
6	1,26297368488640000000	1,26297368488640000000	0,00000000000000000000
7	0,26658715339901200000	0,26658715339901300000	0,00000000000000000000
8	0,85314248252388300000	0,85314248252388500000	0,00000000000000000000
9	1,22901364363449000000	1,22901364363449000000	0,00000000000000000000
10	0,38463096581878500000	0,38463096581879300000	-0,00000000000000710543
11	1,09470092367507000000	1,09470092367508000000	-0,00000000000001199041
12	0,78369335781513400000	0,78369335781510300000	0,00000000000003108624
13	1,29224759400986000000	1,29224759400988000000	-0,00000000000002198242
14	0,15927884336663700000	0,15927884336655400000	0,00000000000008348877
15	0,56100612363390800000	0,56100612363365400000	0,00000000000025424107
16	1,29984088227140000000	1,29984088227124000000	0,00000000000016098234
17	0,13060457141332100000	0,13060457141393200000	-0,00000000000061106675
18	0,47124562343111200000	0,47124562343307800000	-0,00000000000196537231
19	1,21876518091552000000	1,21876518091782000000	-0,00000000000230437891
20	0,41889502502597200000	0,41889502501833900000	0,00000000000763300534

Ces différences vont aller en s'amplifier pour exploser littéralement après une cinquantaine d'itérations et donner dans les deux colonnes des résultats totalement indépendants. Le graphique ci-dessous représente ces



différences successives entre les deux expressions formellement identiques de calculs. D'autres expressions tout aussi équivalentes peuvent être testées avec des conclusions identiques. On peut calculer :

$$y_{t+1}=((a+1)-(a\,y_t))\,y_t \quad \text{ou} \quad y_{t+1}=a\,y_t+(1-(a\,y_t))\,y_t \quad \text{ou} \quad y_{t+1}=y_t+a(y_t-(y_t^2))$$

et effectuer une centaine d'itération. On arrive à la ligne de valeurs (voir nos fichiers disponibles sur infinimath) :

	(a+1) yt – a yt^2	(a+1) yt – (ayt) . yt	((a+1) – (a yt)) yt	a yt +(1– (a yt)) yt	yt + a(yt – (yt)^2)
100	0,00006712709491283020	0,11945743944522900000	1,25649567629486000000	0,65804664793056400000	1,04282303339009000000

On peut légitimement se poser la question angoissante : quelle est la *vraie* valeur de x_{100} ?

Encadré

Pierre-François Verhulst (1804 - 1849) est un mathématicien belge et plus précisément bruxellois, élève de Quetelet. Il est surtout connu pour sa construction de la courbe qu'il a lui même intitulée « logistique », sans fournir d'explication satisfaisante à cette dénomination. Délaissant les modèles exponentielles de Malthus, Verhulst propose une modèle de croissance sous contrainte particulièrement bien adapté à l'évolution des populations puisqu'il conduit à l'existence d'une taille maximale pour toute population, fonction des conditions initiales. Verhulst a appliqué à plusieurs reprises son modèle à la population belge avec des conclusions variées. Détail amusant, la thèse de Verhulst, soutenue en 1825, portait sur la résolution des équations binomiales. Un demi siècle plus tard, lorsque Conan Doyle décida de donner à Sherlock Holmes un ennemi mortel, il en fit un mathématicien ... dont les travaux portaient précisément sur ce sujet. Verhulst a-t-il servi de modèle à Moriarty ?