



**HAL**  
open science

## Erreurs en arithmétique des ordinateurs

Vincent Lefèvre, Jean-Michel Muller

► **To cite this version:**

Vincent Lefèvre, Jean-Michel Muller. Erreurs en arithmétique des ordinateurs. Images des Mathématiques, 2009. hal-00598554

**HAL Id: hal-00598554**

**<https://hal.science/hal-00598554v1>**

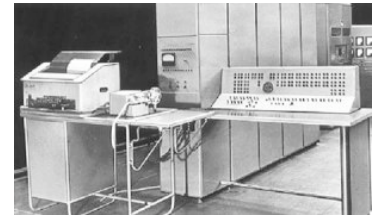
Submitted on 6 Jun 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Erreurs en arithmétique des ordinateurs

Le 20 juin 2009, par **Vincent Lefèvre** et **Jean-Michel Muller**



**I**l est très difficile d'écrire un très gros programme informatique sans laisser au moins une erreur à l'intérieur. Heureusement, les erreurs sont souvent soit facilement détectables (et donc vite corrigées), soit peu dangereuses. Il arrive hélas parfois qu'une erreur pouvant avoir des conséquences graves passe inaperçue lors des tests du programme. Ceci peut être particulièrement préoccupant lorsque l'erreur est localisée dans l'un des programmes ou des circuits qui effectuent les opérations arithmétiques en machine : en effet, ces programmes ou ces circuits sont utilisés très souvent, par de nombreux logiciels, et parfois dans ce qu'on appelle des applications « critiques », comme le contrôle d'un véhicule ou d'un processus potentiellement dangereux. Bien entendu, les erreurs dangereuses ne sont pas toutes arithmétiques, mais c'est sur celles-ci que nous nous concentrerons.

Donnons quelques exemples particulièrement frappants d'erreurs récentes en arithmétique des ordinateurs.

- Dans la version 6.0 du système de calcul formel Maple, si on entrait :

```
21474836480413647819643794
```

la « quantité » affichée et mémorisée était :

```
413647819643790)+'.(-.(
```

- L'opérateur de division de la première version du processeur Pentium d'Intel donnait parfois des résultats incorrects. Par exemple le calcul de

```
8391667/12582905
```

donnait 0.666869... au lieu de 0.666910... Ce problème a fait couler beaucoup d'encre en 1994. Il ne semble toutefois pas qu'il ait eu des conséquences graves... autrement que, peut-être, pour la carrière de l'ingénieur qui a commis la bévue.

- En 1998, à bord du navire lance-missiles américain USS Yorktown, un membre de l'équipage a par erreur tapé un « zéro » sur un clavier, au lieu d'une valeur non nulle. Ceci a provoqué une division par zéro. C'est d'habitude un problème bénin, mais visiblement le programmeur de l'application utilisée n'avait pas songé que ce problème pourrait arriver. Il s'en est suivi une cascade d'erreurs qui a fini par entraîner l'arrêt du système de propulsion du bateau.

Il y a de nombreux autres exemples de systèmes donnant des résultats incorrects à cause d'une mauvaise implantation de l'arithmétique. Ne paniquons pas : de tels problèmes restent rares. Ils ne le sont toutefois pas assez pour que l'on puisse complètement les négliger.

Il arrive que des erreurs proviennent d'une mauvaise spécification : les diverses équipes travaillant sur un même projet, ou encore le concepteur et l'utilisateur d'un programme ne se sont pas suffisamment mis d'accord sur ce que doit faire exactement le programme. Par exemple, la sonde Mars Climate Orbiter s'est écrasée sur Mars en septembre 1999 à cause d'une étourderie ahurissante : une partie des concepteurs des logiciels supposait que l'unité de mesure était le mètre, et l'autre partie que c'était le pied (l'unité anglaise).

Même lorsque le système informatique n'est pas en cause, certains calculs sont intrinsèquement difficiles et conduiront à des erreurs, même sur des machines irréprochables.

Considérons la suite de nombres  $3/2, 5/3, 9/5, 17/9, \dots$ , où chaque nouveau terme se calcule comme suit :

- on appelle  $x$  l'avant-dernière valeur calculée, et  $y$  la dernière ;
- le nouveau terme vaut

$$2003 - \frac{6002}{y} + \frac{4000}{xy} .$$

Par exemple, si on veut calculer le terme qui vient après  $17/9$ , on pose  $x = 9/5$ ,  $y = 17/9$ , et on trouve aisément

$$2003 - \frac{6002}{y} + \frac{4000}{xy} = 33/17 .$$

Cette suite de nombres *tend* vers 2 (ce qui veut dire que les termes  $u_n$  s'approchent de plus en plus de 2 lorsque  $n$  croît). Pourtant si on calcule ces termes sur n'importe quel ordinateur (sauf avec un système qui fait du calcul exact avec des nombres rationnels), on aura l'impression qu'elle tend vers 2000. Par exemple, la table suivante montre ce qu'on obtient avec une arithmétique virgule flottante de base 10 avec une précision de 10 chiffres (comme celle d'une calculette, par exemple).

$n$	Valeur calculée	Valeur exacte
2	1,800001	1,800000000
3	1,890000	1,888888889
4	3,116924	1,941176471
5	756,3870306	1,969696970
6	1996,761549	1,984615385
7	1999,996781	1,992248062
8	1999,999997	1,996108949
9	2000,000000	1,998050682
10	2000,000000	1,999024390

En annexe, le lecteur mathéux trouvera une explication de ce comportement étrange.

---

## La virgule flottante

---





### Annexe 1 : Comment calculer des fonctions sur plusieurs millions de chiffres ? (piste rouge)

Lorsque l'on désire calculer un sinus, un logarithme, une exponentielle, etc., avec une précision allant de quelques dizaines à quelques milliers de chiffres, la solution la plus couramment retenue consiste à approcher la fonction désirée par un polynôme. Pour obtenir de bien plus grandes précisions à un coût raisonnable, on utilise la moyenne arithmético-géométrique de Gauss-Legendre. Si on part de deux valeurs  $a_0$  et  $b_0$ , et que l'on construit des suites de valeurs  $a_n$  et  $b_n$  à l'aide des relations

$$\begin{aligned} a_{n+1} &= \frac{a_n + b_n}{2} \\ b_{n+1} &= \sqrt{a_n b_n} \end{aligned}$$

on peut montrer que les deux suites  $a_n$  et  $b_n$  convergent très vite (c'est-à-dire s'approchent très vite) d'une valeur  $A(a_0, b_0)$ , qui est appelée moyenne arithmético-géométrique de  $a_0$  et  $b_0$ . La table ci-dessous donne les premières étapes du calcul de  $A(1, 2)$ .

$i$	$a_i$	$b_i$
0	1	2
1	1.5	1.4142135623730
2	1.4571067811865	1.4564753152197
3	1.4567910481542	1.4567910139395
4	1.4567910310469	1.4567910310469

En utilisant (pour un  $x$  grand) la relation

$$\frac{\pi}{2A(1, 4/x)} = \log(x) + \frac{4 \log(x) - 4}{x^2} + \frac{36 \log(x) - 42}{x^4} + \dots,$$

on se ramène au calcul d'une moyenne arithmético-géométrique bien choisie pour calculer des logarithmes avec une très grande précision.

### Annexe 2 : la suite qui tend vers 2, mais à l'air de tendre vers 1000 en machine (piste rouge)

Cherchons à voir pourquoi la suite de nombres donnée en exemple dans l'introduction de cet article a un comportement aussi étrange. Dans cette partie, je supposerai que le lecteur possède un bagage mathématique de niveau L1 ou classe prépa.

La suite discutée plus haut est la suite de nombres  $u_0, u_1, u_2, \dots$  définie par :

$$\begin{cases} u_0 &= 3/2 = 1.5 \\ u_1 &= 5/3 = 1.666666 \dots \\ u_{n+1} &= 2003 - \frac{6002}{u_n} + \frac{4000}{u_n u_{n-1}} \end{cases}$$

Il n'est pas très difficile de montrer (par exemple par récurrence), qu'une suite  $v_n$  satisfaisant la relation

$$v_{n+1} = 2003 - \frac{6002}{v_n} + \frac{4000}{v_n v_{n-1}}$$

est de la forme

$$v_n = \frac{\alpha + \beta \cdot 2^{n+1} + \gamma \cdot 2000^{n+1}}{\alpha + \beta \cdot 2^n + \gamma \cdot 2000^n},$$

où  $\alpha, \beta$  et  $\gamma$  dépendent des « points de départ »  $v_0$  et  $v_1$ . On en déduit aisément que si  $\gamma$  est non nul, alors la suite tend vers 2000, et que si  $\gamma$  est nul, mais pas  $\beta$ , elle tend vers 2 (si  $\beta$  et  $\gamma$  sont nuls, la suite perd quelque peu de son intérêt).

Les points de départ  $u_0 = 3/2$  et  $u_1 = 5/3$  ont été choisis pour que  $\gamma$  soit nul. La suite tend donc vers 2. Mais numériquement, dès que l'on a commis des erreurs d'arrondi (aussi infimes soient-elles), on se retrouve sur une « trajectoire » correspondant à une valeur de  $\gamma$  certes extrêmement petite mais non nulle, ce qui suffit à faire tendre les termes calculés vers 2000.

## Notes

[▲1] Certes, ceux-ci nous affichent leurs résultats en base 10, mais avant de le faire ils effectuent une conversion de base. Ce sont les calculs *internes* qui sont effectués en base 2.

[▲2] La photo illustrant l'article représente l'ordinateur ternaire « Setun » développé à l'Université de Moscou (1958). Cet ordinateur a été fabriqué par l'usine mathématique de Kazan. 50 ordinateurs ont été fabriqués, dont 30 ont été exploités dans les Universités d'Union Soviétique. **Source.**

[▲3] S'il ne choisit rien, c'est l'arrondi au plus près qui est pris par défaut.

Pour citer cet article : **Vincent Lefèvre** et **Jean-Michel Muller**, « Erreurs en arithmétique des ordinateurs » — *Images des Mathématiques*, CNRS, 2009. En ligne, URL : <http://images.math.cnrs.fr/Erreurs-en-arithmetique-des.html>