LES ERREURS D'ARRONDI

I. Arithmétique en virgule flottante

Du fait des implacables contraintes matérielles, les représesentations de nombres dans un microprocesseur sont bien évidemment bornées[1], tandis que les mathématiciens calculent avec des nombres qui possèdent bien souvent une infinité de chiffres après la virgule. Vouloir faire des mathématiques avec une calculatrice, ou un ordinateur, semble relever de la gageure ! Et pourtant, on y parvient, non sans au préalable avoir réfléchi de très près aux problèmes qui se posent.
Les constructeurs de calculatrices gardent jalousement le secret sur leurs systèmes de représentation des nombres. Mais on peut penser, sans prendre trop de risques, qu'ils utilisent les normes dites IEEE, mises en place depuis 1985 pour les ordinateurs. C'est celles que nous décrirons ici.

1) Écriture d'un réel x en virgule flottante

- Les nombres machines s'écrivent sous la forme

$$x = \pm(0, d_1 d_2 ... d_N) \times 10^p$$

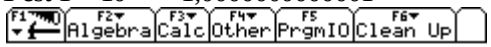
où N est le nombre de chiffres significatifs de la calculatrice, ici égal à 14 et où d_1, d_2, ..., d_N est un chiffre du système décimal.

$m = (0, d_1 d_2 ... d_N)$ est appelé la mantisse du nombre x.

Si l'on suppose de plus que $d_1 \neq 0$, le nombre est dit écrit en *virgule flottante normalisée*.
Remarquons que m est nécessairement compris entre 0,1 et 0,99999999999999, soit 1/10 et $1 - \frac{1}{10^{14}}$.

p est appelé l'exposant du nombre x : il peut varier entre –999 et 999.

- Le successeur de 1 est $1 + 10^{-13} = 1{,}0000000000001$



10^{-13} est appelé *l'epsilon machine.*

Dans un tel système,on peut représenter un nombre fini de nombres réels, nombre égal à :

$$2 \times 9 \times 10^{13} (2 \times 999 + 1) + 1.$$

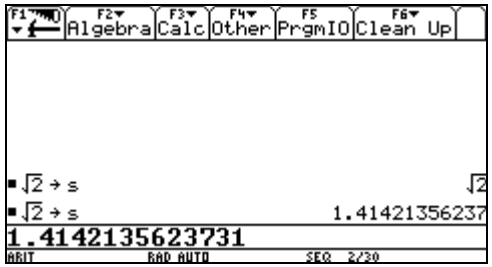
En effet, on a : 2 possibilités pour le signe,
9 pour le choix de d_1
10^{13} pour le choix de d_2, ..., d_{14}
$2 \times 999 + 1$ pour l'exposant et on rajoute un nombre avec le 0.

2) Les erreurs d'affectation

- **La notion d'erreur d'affectation**

Il résulte de ce que l'on vient de dire qu'une calculatrice, hormis celles travaillant en calcul formel[2], commet presque systématiquement une erreur en travaillant avec un nombre mathématique qu'elle ne peut pas mémoriser intégralement.

Par exemple, quand on croit travailler avec $\sqrt{2}$, la calculatrice affiche en fait le nombre *décimal* 1,41421356237 (12 chiffres significatifs) et elle effectue ses calculs avec 1,4142135623731, soit 14 chiffres significatifs, ce que l'on observe aisément si l'on fait apparaître les chiffres de réserve (voir l'écran ci-dessous) :



Par conséquent, en remplaçant $\sqrt{2}$ par 1,41421356237, ou par 1,4142135623731, on commet une erreur, certes très petite, mais inévitable.
Cette erreur est appelée *erreur d'affectation.*

- **Estimation de l'erreur d'affectation sur le résultat affiché**

Elle est facile à évaluer. Reprenons notre exemple précédent.
Dans le premier cas, comme la calculatrice a arrondi, on est sûr que :

$$1{,}4142135623\mathbf{65} \leqslant \sqrt{2} < 1{,}4142135623\mathbf{75}.$$

L'erreur absolue est donc encadrée par :

$$-5 \times 10^{-12} \leqslant \sqrt{2} - 1{,}41421356237 \leqslant 5 \times 10^{-12}$$

ce que l'on peut écrire

$$|\sqrt{2} - 1{,}41421356237| \leqslant 5 \times 10^{-12}.$$

[1] En général les nombres sont codés sur 64 bits pour un ordinateur.. permettant la représentation de 2^{64} nombres réels différents.

[2] Nous supposeront toujours dans la suite être dans une situation de calcul approché, par exemple sur une Voyage 200.

On peut aussi calculer les erreurs relatives, car les erreurs absolues dépendent du nombre mémorisé : on conçoit qu'elles ne soient pas identiques avec $\sqrt{2}$ et $\sqrt{123456789}$ ...

Dans le cas qui nous intéresse, l'erreur relative vaut à peu près $\frac{5\times10^{-12}}{\sqrt{2}} \approx 3,57\times10^{-12}$.

- **Estimation de l'erreur d'affectation en tenant compte des chiffres de réserve**

Dans le deuxième cas, la calculatrice a cette fois-ci tronqué et l'on a :

$$1,4142135623731 \leqslant \sqrt{2} < 1,4142135623732$$

L'erreur absolue vaut alors :

$$0 \leqslant \sqrt{2} - 1,4142135623731 \leqslant 0,0000000000001 = 10^{-13}$$

ce qui donne cette fois une erreur relative de $\frac{10^{-13}}{\sqrt{2}} \approx 7,07\times10^{-14}$.

- On dispose du résultat général suivant qui donne une majoration de l'erreur commise. On note habituellement fl(a) le nombre avec lequel la calculatrice travaille.

Théorème

> Dans une arithmétique flottante à t chiffres on a $|a - \text{fl}(a)| \leq 5\ |a|\ p\ 10^{-t}$, avec $p = 1$ dans le cas de l'arrondi et $p = 2$ dans le cas de la troncature.

II. Opérations en arithmétique à virgule flottante

En général, le résultat d'une opération portant sur deux nombres machine donne rarement exactement un nombre machine : la calculatrice doit arrondir avant d'afficher la réponse. Après les erreurs d'affectation, vues précédemment, c'est la deuxième source d'erreur quand on travaille avec une calculatrice !

1) Quelques exemples de calcul

On suppose pour fixer les idées que l'on travaille avec seulement 3 chiffres significatifs, avec des exposants compris entre –15 et 15. Considérons les nombres normalisés suivants :

$x = +0,125 \times 10^{6}$
$y = -0,128 \times 10^{6}$
$z = +0,437 \times 10^{12}$
$t = +0,657 \times 10^{7}$
$u = +0,215 \times 10^{-10}$

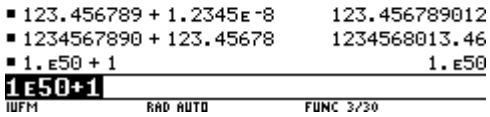
- **Additions et soustractions**

On additionne, ou on soustrait, en mettant en évidence un exposant commun, puis en effectuant l'opération

$x + t = (+0,125 \times 10^{6}) + (+0,657 \times 10^{7}) = (+0,0125 + +0,657) \times 10^{7}$
$= +0,6695 \times 10^{7} = 0,669 \times 10^{7}$

$x + z = (+0,125 \times 10^{6}) + (+0,437 \times 10^{12}) = (+0,000000125 + +0,437) \times 10^{12}$
$= +0,437000125 \times 10^{12} = +0,437 \times 10^{12}$

(remarquons que dans ce cas, $x + y = y$, bien que x soit non nul !)



Le problème particulier des différences évanescentes

Les erreurs précédentes peuvent paraître insignifiantes : elles tiennent à ce qu'on ne peut pas avec un système fini représenter l'infinité des nombres réels. Mais elles peuvent prendre un caractère plus pernicieux, par exemple avec le problème dit des *différences évanescentes*[3]

Soit par exemple à effectuer la différence des nombres *très voisins* x et y suivants :

$x = +0,324 \times 10^{6}$ et $y = +0,323 \times 10^{6}$

On peut écrire :

$x - y = (+0,324 \times 10^{6}) - (+0,323 \times 10^{6}) = (+0,324 - +0,323) \times 10^{6}$
$= +0,001 \times 10^{6} = 0,100 \times 10^{4} = 1000$

Mais en y regardant de plus près, x peut provenir par exemple de 324 499, arrondi à 324 000 = $+324 \times 10^{6}$ et y de 323 000.

$x - y$ en réalité vaudra 1499, à rapprocher de 1000 donné de la calculatrice : on comprend que les trois 0 de 1000 ne sont pas du tout significatifs.

Un calcul d'erreur précise cette remarque.

A priori, x représente un nombre qui, compte tenu des arrondis, est dans l'intervalle $[+0,3235 \times 10^{6} ; +0,3245 \times 10^{6}[$, tandis que $-y$ représente un nombre de l'intervalle $]-0,3235 \times 10^{6} ; -0,3225 \times 10^{6}]$.

On peut donc affirmer que que $x - y$ se trouve dans l'intervalle $]0 ; +0,200 \times 10^{4}[$.

L'erreur sur le résultat renvoyé, $+0,100 \times 10^{4}$, est donc de 100%, après un seul calcul, ce qui est énorme ! On ne peut pas l'accepter.

La différence entre deux nombres très voisins a fait littéralement s'évanouir des chiffres significatifs, d'où le terme employé[4]. C'est une situation qu'il faut apprendre à identifier : elle n'est susceptible d'apparaître qu'avec des différences, qu'on peut bien souvent contourner par des moyens algébriques.

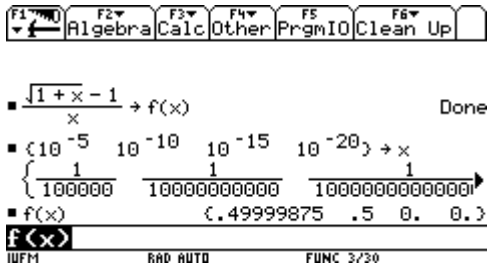
---

[3] C'est le principal problème que l'on rencontre avec les additions-soustractions : à tel point que si l'on peut éviter de faire une soustraction par transformation algébrique, on ne s'en privera pas, quitte à ce que les calculs soient plus compliqués !

[4] On parle aussi d'erreur de cancellation.

Ainsi par exemple[5], c'est la situation que l'on rencontre quand on cherche à conjecturer à la calculatrice la limite quand x tend vers 0 de $f(x)=\dfrac{\sqrt{1+x}-1}{x}$

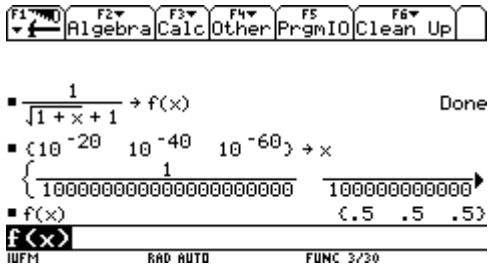
Ne prendre aucune précaution de calcul n'est pas très sage, et conduit à conjecturer que la limite vaut 0 :



On doit au contraire être très prudent, à cause de la soustraction : et nous sommes bien dans une situation de différences évanescentes pour des puissances de 10 très proches de 0. Tous les chiffres significatifs d'ailleurs disparaissent puisque le résultat renvoyé est 0.
Peut-on s'affranchir de cette différence ? C'est très simple algébriquement :

$$f(x)=\frac{\sqrt{1+x}-1}{x}=\frac{1+x-1}{x\left(\sqrt{1+x}+1\right)}=\frac{1}{\sqrt{1+x}+1}$$

Le calcul est alors correct même pour des valeurs de x très petites. On peut conjecturer que la limite est bien 0,5.



- **Multiplications ou divisions**

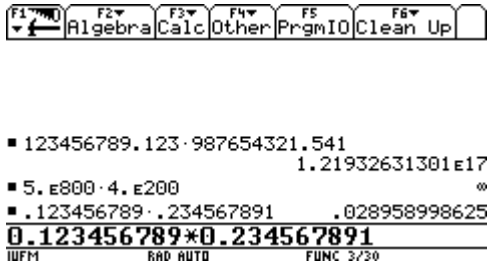
On multiplie, ou on divise, en multipliant, ou en divisant, les mantisses et en additionnant ou en soustrayant les exposants.

$x \times y = (+0{,}125 \times 10^{6}) \times (-0{,}128 \times 10^{6}) = (+0{,}125 \times -0{,}128) \times 10^{12}$
$= -0{,}016000 \times 10^{12} = -0{,}160 \times 10^{11}$

$x \times z = (+0{,}125 \times 10^{6}) \times (+0{,}437 \times 10^{12}) = (+0{,}125 \times +0{,}437) \times 10^{18}$
$= +0{,}054625 \times 10^{18}$ **sur-dépassement de capacité ou overflow**

[5] Voir aussi le problème de la résolution de l'équation du second degré *Raymonde, ou du méfait des différences évanescentes.*

(dans ce cas la Voyage 200 renvoie l'infiniet l'on se rend compte du problème)

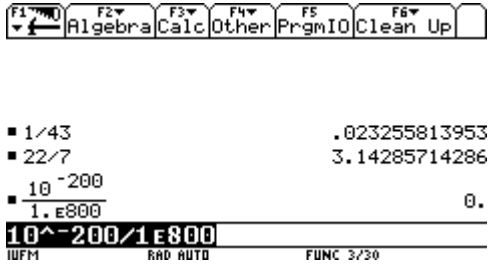


Pour la division :

$x / z = (+0{,}125 \times 10^{6}) / (+0{,}437 \times 10^{12}) = (+0{,}125 / +0{,}437) \times 10^{-6}$
$= +0{,}286041 \times 10^{-6}$

$u / z = (+0{,}215 \times 10^{-10}) / (+0{,}437 \times 10^{12}) = (+0{,}215 / +0{,}437) \times 10^{-22}$
$= +0{,}491990 \times 10^{-22}$ **sous dépassement de capacité ou underflow**

(Attention le résultat sera alors arrondi à 0 et les calculs pourront être poursuivis)

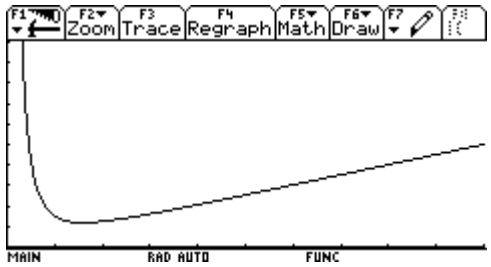


III. Propagation des erreurs d'arrondi lors d'un calcul

1) Le problème des erreurs d'arrondi

Le paragraphe précédent montre les problèmes que l'on peut rencontrer sur *un* calcul : mais bien souvent, dans un programme par exemple, des dizaines voire des centaines de calculs sont enchaînés... Les erreurs d'arrondi d'un calcul particulier vont alors se trouver ballotées au gré des calculs qui suivent : elles étaient petites, elles vont grossir un peu, parfois exagérément au point de se substituer au calcul que l'on fait…
Par exemple dans le cas d'Alfred Logarithme, une erreur très petite, de l'ordre de 10^{-14} se trouve multipliée par un nombre très grand (25! de l'ordre de 10^{25}) et fausse complètement le résultat final en 25 étapes...
La situation est dans ce cas relativement explosive, à cause de la croissance impressionnante de la factorielle. Le cas général est moins spectaculaire, mais il n'en demeure pas moins que lors de tout calcul, les erreurs d'arrondi finiront par prendre une place qu'il vaudra mieux estimer.

Il y a d'ailleurs ici un véritable conflit à gérer : d'un côté l'analyse numérique affirme que pour accroître la précision d'une méthode, on doit répéter un grand nombre de fois les itérations (erreur par exemple de type k'/n^2) ; de l'autre, l'algorithmique rappelle que plus on fait de calculs, plus les erreurs d'arrondi sont importantes (erreurs du type kn). L'erreur totale est la somme des deux erreurs précédentes : par exemple elle est de la forme $\frac{k}{n^2}+k'n$. L'étude rapide d'une telle fonction montre que l'erreur globale présente un minimum : au dela de cette valeur, il est inefficace de poursuivre les calculs à cause des erreurs d'arrondi.

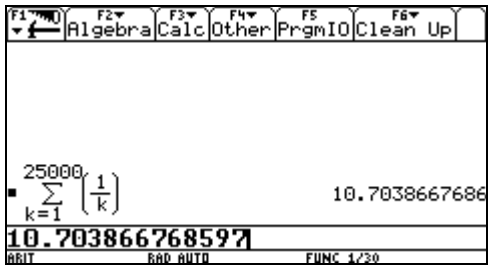


2) Étude d'un exemple

Étudions la propagation des erreurs d'arrondi et d'affectation, lorsque l'on calcule :

$$S=\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{45000}$$

Le résultat donné par la TI92 est le suivant :



c'est-à-dire 10,7038667686 arrondi avec 12 chiffres significatifs, et 10,703866768597 tronqué avec 14 chiffres significatifs.
Quelle confiance peut-on accorder aux résultats de la calculatrice ?

Estimation de l'erreur
Tentons de faire un calcul d'erreur sur ces deux résultats : on considère, pour simplifier, que l'erreur d'affectation est commise pour chaque terme de la somme alors qu'elle ne l'est pas en fait pour ceux dont la division tombe juste…

- Pour le premier calcul, si l'on suppose que chacune des fractions est arrondie à 12 chiffres significatifs[6], on sait qu'alors l'erreur commise sur un terme est majorée par 5×10^{-13}. L'erreur totale est donc majorée par

$$25000 \times 5 \times 10^{-13} = 1{,}25000 \times 10^{-8}$$

A priori, on est donc sûr que :

$$10{,}7038667686 - 1{,}25 \times 10^{-8} \leqslant S \leqslant 10{,}7038667686 + 1{,}25 \times 10^{-8}$$

soit que

$$10;7038667\mathbf{561} \leqslant S \leqslant 10;7038667\mathbf{811}$$

Selon ce calcul, seules les 7 premières décimales sont sûres !

- Pour le second résultat, chacune des fractions est tronquée à 14 chiffres significatifs. L'erreur commise sur un terme est majorée par 10^{-14} et l'erreur totale par :

$$25000 \times 10^{-14} = 2{,}5 \times 10^{10}$$

Cette fois, on a l'encadrement suivant :

$$10{,}703866768597 \leqslant S \leqslant 10{,}703866768597 + 2{,}5 \times 10^{-10}$$

soit que

$$10{,}703866768\mathbf{597} \leqslant S \leqslant 10{,}703866768\mathbf{847}$$

Cette fois, les 9 premières décimales sont sûres.

Un résultat plus précis
- En utilisant le logiciel de calcul formel Derive, en lui demandant d'afficher 30 chiffres significatifs… On obtient comme valeur approchée :

10.7038667686185373285500051205

On peut donc considérer
d'une part que le résultat arrondi avec 12 chiffres significatifs est :
10,7038667686;
et c'est bien là le résultat donné par la TI92 ;
et d'autre part que le résultat tronqué avec 14 chiffres significatifs est :
10,703866768618;
alors que la calculatrice donne 10,703866768*597*, en commettant cette fois une erreur, bien plus petite que celle que nous avons prévue précédemment, de $-2;1 \times 10^{-11}$.
Sur cet exemple, on constate que l'erreur, que l'on sait irrémédiable pour tout calcul obtenu par ordinateur, ne s'est propagée que sur les chiffres de réserve de la calculatrice : c'est d'ailleurs le rôle fondamental de ces derniers que de garantir une meilleure tenue des calculs.

- Au vu du résultat fourni par Derive, on constate que les calculs d'erreur du paragraphe précédent pêchent par excès de pessimisme. Dans un tel calcul, il est en effet rare, voire impossible, que chacun des termes soit systématiquement approximé de la façon la plus défavorable qui soit. Par ailleurs, dans le cas de l'arrondi, les erreurs, tantôt par excès, tantôt par défaut, vont se compenser, au moins partiellement.
Finalement, la réalité est bien éloignée du calcul théorique…

Une approche statistique
Appelons ε l'erreur commise sur la somme S précédente (possédant n = 25000 termes) et e_k l'erreur d'arrondi commise sur le terme d'ordre k de cette somme. On a donc

[6] On peut penser qu'en fait, la calculatrice effectue le calcul sur 14 chiffres et n'arrondit qu'une fois pour afficher le résultat.

$$\varepsilon = \sum e_k \text{ et } |e_k| \leqslant 5 \times 10^{-13}.$$

Faisons alors les deux hypothèses statistiques suivantes :

(1) les erreurs e_k sont des variables aléatoires indépendantes les unes des autres ;

(2) l'espérance mathématique de e_k est égale à 0 : autrement dit, l'arrondi n'a pas plus de raison de se faire par excès que par défaut.

Remarquons tout de suite que $V(e_k) \leqslant (5 \times 10^{-13})^2$

D'après l'hypothèse (1), on a $E(\varepsilon) = E\left(\sum e_k\right) = \sum E(e_k) = 0$.

D'après l'hypothèse (2), on a $V(\varepsilon) = \sum V(e_k) \leqslant n \times (5 \times 10^{-13})^2$ et donc :

$$\sigma(\varepsilon) = \sqrt{\text{var}(\varepsilon)} \leqslant \sqrt{n} \times 5 \times 10^{-13}$$

D'après l'inégalité de Bienaymé-Tchebichev, on peut écrire :

$$p\left(|\varepsilon| \geq t\sigma(\varepsilon)\right) \leq \frac{1}{t^2}$$

Ainsi la probabilité que l'erreur dépasse $10 \times \sqrt{n} \times 5 \times 10^{-13} \approx 8{;}7 \times 10^{-10}$ est inférieur à 1% : on obtient un résultat beaucoup plus optimiste que celui du paragraphe précédent, mais il n'est certain qu'à 99% !

3) Correction de l'arithmétique de l'ordinateur sur l'ordinateur précédent

- Non seulement il y a dans le calcul précédent des erreurs d'affectation, mais aussi des erreurs d'arrondi dans les additions.

Étudions un exemple particulier de ces erreurs d'arrondi, au moment où la TI92 vient de calculer

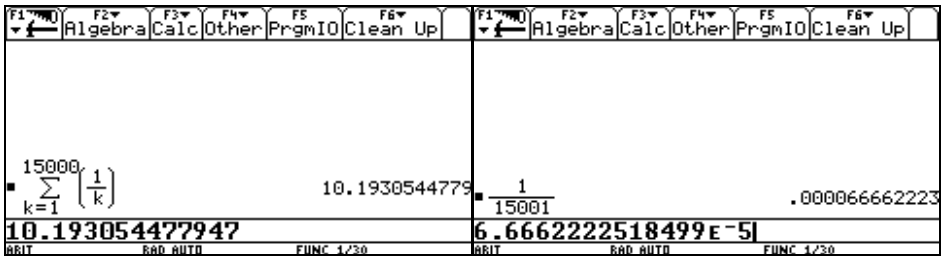
$$s = \sum_{k=1}^{15000} \frac{1}{k}$$

et qu'elle doit ajouter pour poursuivre le calcul $\frac{1}{15001}$.

Examinons précisément quels calculs sont faits par la calculatrice. En faisant apparaître les chiffres de réserve, on a :

$s \approx 10{,}193054477947$

$\frac{1}{15001} \approx 0{,}000066662222518499$



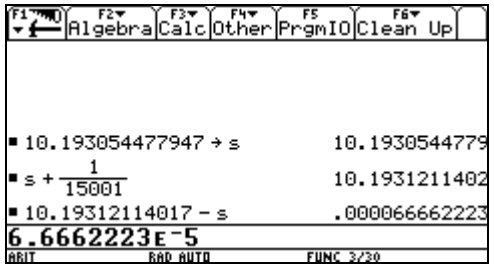
La somme effectuée à ce moment est donc la suivante, les chiffres en grisé n'étant pas pris en compte par la calculatrice :

	1	0	,	1	9	3	0	5	4	4	7	7	9	4	7					
+		0	,	0	0	0	0	6	6	6	6	2	2	2	2	5	1	8	4	9	9
	1	0	,	1	9	3	1	2	1	1	4	0	1	6	9	5	1	8	4	9	9

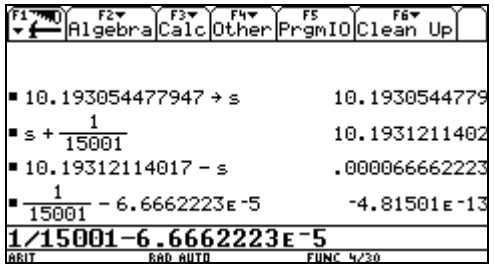
6 chiffres significatifs de $\frac{1}{15001}$ sont donc purement et simplement occultés dans ce calcul : c'est un perte importante, mais qu'il est possible de limiter en réfléchissant un peu.

En effet, effectuons la suite de calculs suivants :

d'abord, $(s + \frac{1}{15001}) - s$, qui met en évidence la *valeur* de $\frac{1}{15001}$ qui a été incorporée à s ;



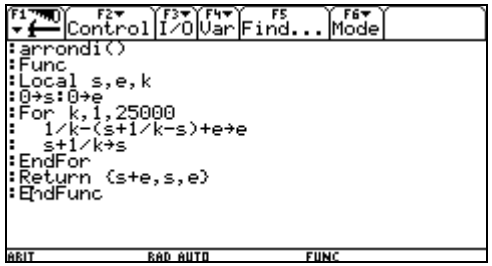
par suite, 1/15001, duquel on ôte le résultat précédent, donne l'erreur que l'on commet en n'utilisant pas tous les chiffres de $\frac{1}{15001}$ (voir la dernière ligne de calcul sur l'écran ci-dessous).



Remarquons que le résultat n'est pas 0, car les règles usuelles de calcul dans $\mathbb{R}$ ne sont pas conservées pour l'ensemble des nombres de la calculatrice…

- Les remarques précédentes nous conduisent à écrire une fonction, pour atténuer les erreurs d'arrondi dues aux sommes successives : en effet, on peut facilement cumuler dans une variable *e* les erreurs d'arrondi calculées comme précédemment. Il suffit alors que la fonction retourne *s* + *e* au lieu de *s* pour que l'obtienne un résultat plus précis.

C'est ce que fait la fonction suivante :

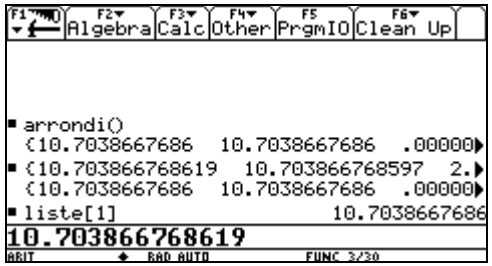


Examinons les résultats obtenus (ne pas oublier de lancer le programme par diamant enter, pour se mettre en mode approché) :

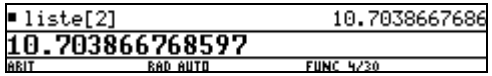
Le calcul est cette fois beaucoup plus long que précédemment : on n'y gagne pas à remplacer une fonction de la calculatrice (*Σ*), par un programme en TI Basic. Mais la réponse obtenue est particulièrement intéressante.

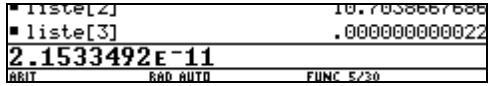
Tout d'abord si l'on compare avec ce que donne Derive, le résultat à 14 chiffres, 10;703866768619, est correct : c'est même exactement l'arrondi d'ordre 14. La calculatrice, connaisant les décimales de *e*, sait que le chiffre qui suit est 5, ce qui lui permet d'arrondir !

La propagation des erreurs d'arrondi a donc été neutralisée, même sur les chiffres de réserve…



Mais on peut encore affiner en faisant apparaître les chiffres de réserve de réserve de *s* et de *e* :





On obtient *s* ≈ 10;703866768597 et *e* ≈ 0;000000000021533492

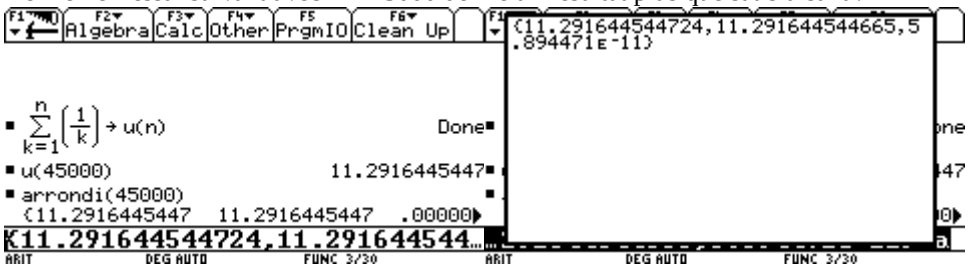
	1	0	,	7	0	3	8	6	6	7	6	8	5	9	7				
+			0	,	0	0	0	0	0	0	0	0	0	0	2	1	5	3	3	4	9	2
	1	0	,	7	0	3	8	6	6	7	6	8	6	1	8	5	3	3	4	9	2	

En comparant avec Derive,

10.703866768618537328550005120 5

on obtient cette fois-ci un résultat dont les 14 premières décimales sont justes…

De même l'essai suivant avec *n* = 45000 donne un résultat plus que satisfaisant :



La somme vaut :

	1	1	,	2	9	1	6	4	4	5	4	4	6	6	5				
+			0	,	0	0	0	0	0	0	0	0	0	5	8	9	4	4	7	1
	1	1	,	2	9	1	6	4	4	5	4	4	7	2	3	9	4	*4*	*7*	*1*

Résultat juste avec 16 décimales comme le montre la valeur ci-dessous, donnée par Dérive :

11,291644544723948517790393986146560032065854403356

Annexe

Quelques propriétés de l'ensemble *M* des nombres calculatrices

- C'est une partie finie de $\mathbb{R}$: à ce titre, il possède un plus grand élément et un plus petit élément strictement positif.

- Les propriétés usuelles des calculs dans $\mathbb{R}$ ne sont pas conservées.

L'addition, par exemple, n'est plus *associative* !
Ainsi par exemple, si l'on considère les deux expressions suivantes :

$u = (y + x) - x$

$v = y + (x - x)$

Mathématiquement, elle sont égales du fait de l'associativité de l'addition. Pour la calculatrice, cela dépend. Ainsi, avec $x = 1$ et $y = 10^{-15}$, on obtient :

$u = (10^{-15} + 1) - 1 = 1 - 1 = 0$

$v = 10^{-15} + (1 - 1) = 10^{-15}$

La différence entre les deux peut paraître faible, mais si l'on calcule $\frac{(y+x)-x}{y}$ et $\frac{y+(x-x)}{y}$, on trouve respectivement 0 et 1...

L'addition n'est plus *régulière*. En effet, sans $\mathbb{R}$, $a + x = b + x$ entraîne $a = b$. Avec une calculatrice, on a $1 + 10^{-30} = 1$: pourtant $10^{-30} \neq 0$.

Sur une calculatrice, une série dont le terme général tend vers 0 est convergente.

Quelques précautions dans les calculs...

Il faut impérativement veiller à conserver le maximum de chiffres significatifs...
On a déjà vu la propagation presque inexorable des erreurs d'arrondis sur les chiffres de réserve.
Attention principalement au problème des différences évanescentes, que l'on rencontre quand on soustrait deux nombres très proches. On doit donc systématiquement se méfier dès que dans un calcul une soustraction est faite.
On évite aussi au maximum la perte des chiffres significatifs en sommant par paquets les quantités de même ordre de grandeur.