# Échanges

# À propos du CAPES et des calculs numériques

François GRAMAIN
Université de Saint-Étienne

**Résumé**. Prenant prétexte de l'épreuve d'analyse du CAPES 1997, on donne quelques indications sur le fonctionnement des calculettes et le calcul des erreurs d'arrondi.

## 0. Introduction

Cette note a pour but de donner quelques indications sur le mode de fonctionnement des calculettes et d'en tirer quelques conclusions sur la façon de procéder pour quelques calculs simples. Elle ne prétend en aucun cas être une synthèse théorique et le fil conducteur en est la première épreuve du CAPES 1997 et, plus généralement, les questions de calcul numérique posées à ce concours (*).

Les sujets de la première composition de mathématiques du CAPES comportent souvent un calcul numérique, mais les indications concernant ce calcul sont très variables :

En 1996, il n'y a aucun commentaire sur la question *Calculer une valeur approchée à* $10^{-5}$ *près de* $\lambda = y(1/3)$ *et* $\mu = y(2/3)$.

(*) Pour la théorie, on peut consulter le volume 2, et particulièrement la section 4.2, de *The Art of Computer Programming* (Addison-Wesley 1981) de Donald E. Knuth ou n'importe quel bon livre d'algorithmique numérique. Pour d'autres exemples, lire, calculette en main, le chapitre 6 du livre de Thierry Lambre *L'épreuve sur dossier à l'oral du CAPES, II Analyse* (Ellipses 1998).

En 1992, on précise *Calculer à la machine des valeurs approchées de* $x_5$, $x_6$, $x_7$, $x_8$. *Donner une valeur approchée du zéro* $x_*$ *de* $r$ *à* $10^{-4}$ *près : on justifiera le résultat.*

En 1988, il est clairement écrit au II qu'*on précisera l'algorithme utilisé pour calculer* $\sum_{1\leq p\leq n} 1/p$ *et on justifiera que la précision demandée* [c'était $10^{-9}$] *est effectivement obtenue*, et au III qu'*il n'est pas demandé d'étudier les problèmes posés par les erreurs d'arrondi.*

En 1997, l'énoncé est particulièrement précis, puisque, dans le préambule intitulé *Objet et notations du problème*, il est écrit : *Pour chaque calcul numérique, on décrira la méthode de calcul utilisée et on justifiera le résultat en tenant compte des erreurs d'arrondi.*

Le candidat doit donc faire preuve d'un peu de psychologie pour choisir la qualité de justification du calcul qu'il va présenter dans sa copie, mais il doit aussi en savoir un minimum sur les possibilités de sa calculette et ses limites. C'est l'objet des deux premiers paragraphes de cette note. Dans un troisième paragraphe, nous verrons comment calculer avec précision une somme partielle d'une série, calcul assez classique au CAPES. Nous nous limiterons à l'exemple du CAPES 1997 qui, par ailleurs, ne demande pas seulement *une* valeur approchée, mais *la* valeur décimale approchée à $10^{-n}$ (où $n = 3$, 6 ou 7) près par défaut de certaines sommes. Le dernier paragraphe présente une méthode, peu connue, qui permet d'annihiler (ou presque) les erreurs d'arrondi dans les calculs du paragraphe 2.

## 1. Les nombres dans une calculette

La plupart des calculettes, sinon toutes, travaillent avec des nombres stockés sous forme d'une mantisse (nombre entier à nombre fixe de chiffres décimaux, 10 me dit la notice de ma HP25 préhistorique) et d'un exposant de 10 (à deux chiffres sur ma HP25). Il est très facile de vérifier cela sur sa calculette.

Pour l'expliquer j'utiliserai la notation $=_c$, qui représente le signe = de la calculette, c'est-à-dire le résultat affiché par la calculette à la suite d'un calcul. Le sens exact de $=_c$ dépend donc de la calculette utilisée et, par exemple, $1 \div 3 =_c 0{,}333\ 333\ 333$ pour ma HP25. Ensuite une multiplication par 2 fournit $(1 \div 3) \times 2 =_c 0{,}666\ 666\ 667$, ce qui n'est qu'un arrondi d'affichage comme on le voit apparaître en retirant, par exemple, 0,666 pour obtenir $(1 \div 3) \times 2 - 0{,}666 =_c 0{,}\ 000\ 666\ 667$ que l'on multiplie par 1000, ce qui donne enfin

$$((1 \div 3) \times 2 - 0{,}666) \times 1\,000 =_c 0{,}\ 666\ 666\ 600$$

avec sept fois le chiffre 6. Ceci montre clairement que la notice ne mentait pas : HP25 garde effectivement en mémoire dix chiffres significatifs exacts.

Pour voir comment les résultats des calculs sont arrondis, il suffit de regarder $2 \div 3 =_c 0{,}666\,666\,667$ et de lui faire subir le traitement précédent de sorte que

$$((2 \div 3) - 0{,}666) \times 1\,000 =_c 0,\,666\,666\,700.$$

Ainsi le calcul a été fait avec dix chiffres significatifs, le dernier ayant été arrondi, ce qui n'arrive évidemment pas dans le produit $(1 \div 3) \times 2$. Avec une calculette plus moderne, pour voir ce que je viens de montrer, il faut souvent exécuter les calculs à chaque étape, par exemple avec une touche EXE ou EXEC, voire, comme sur la CASIO *fx-7900GC* de mon fils, stocker les résultats partiels dans des mémoires, car la calculette oublie les décimales cachées dès qu'elle affiche un résultat. On vérifie ainsi que la CASIO *fx-7900GC* travaille avec 13 chiffres significatifs, mais n'en affiche que 10. De façon analogue, pour vérifier que TI-89 travaille avec 14 chiffres significatifs, il suffit de faire entrer (ENTER) dans la ligne d'édition de l'écran de calcul HOME les résultats obtenus en mode approché (exécuter les calculs par la commande ◊ ENTER, sinon TI-89 renvoie le résultat exact sous forme d'une fraction irréductible, du moins tant que numérateur et dénominateur n'ont pas plus de 614 chiffres !). En effet, TI-89 refuse de donner plus de 12 chiffres significatifs dans sa zone d'affichage, mais consent à afficher toute sa mémoire sur la ligne d'édition.

Je laisse au lecteur le soin de trouver un moyen simple de savoir combien de chiffres comporte l'exposant de 10 de sa calculette favorite. D'autre part, dans toute la suite, j'admets que la calculette est suffisamment bien câblée pour que les opérations algébriques et les calculs de valeurs de fonctions transcendantes simples soient justes, à l'arrondi final près, du moins en l'absence de dépassement de capacité.

## 2. De petites sommes

La question qui m'intéresse dans cette note est l'estimation de l'erreur d'arrondi faite par une calculette sur la somme d'un assez grand nombre de termes. Mais il faut commencer par le commencement, en ajoutant deux termes : il résulte clairement de ce qu'on vient de voir que, pour $\varepsilon$ suffisamment petit, on a $1 + \varepsilon =_c 1$. Plus précisément, si votre calculette travaille avec $n$ chiffres significatifs, vous vous attendez à obtenir $1 + \varepsilon =_c 1$ dès que $|\varepsilon| \leq_c 9.10^{-n}$.

En fait, c'est un peu plus compliqué que cela, car les calculs sont faits dans le module électronique appelé *accumulateur* avec des mantisses comportant au moins $n + 1$ chiffres. Par des procédés analogues à ceux du paragraphe 1 on vérifie que TI-89 (pour qui $n = 14$) stocke en mémoire les données en les tronquant à 14 chiffres significatifs (il n'y a pas d'arrondi par excès ou par défaut, mais seulement troncature) : la valeur stockée de 0,123 456 789 012 345 6 est 0,123 456 789 012 34. Les opérations sont exécutées dans son accumulateur avec au moins 15 chiffres significatifs et le résultat est renvoyé avec les 14 chiffres significatifs donnés par l'arrondi de Jules Tannery (arrondi qu'il préconisait dans son ouvrage *Leçons d'arithmétique*, Armand Colin, Paris, 1894) : si le 15e chiffre significatif est $\geq 5$, on arrondit le 14e au chiffre supérieur (avec propagation de la retenue éventuelle) et le 14e chiffre reste inchangé si le 15e est $\leq 4$. Ainsi, pour TI-89, on a $1 + \varepsilon =_c 1$ seulement pour $-0{,}6.10^{-14} <_c \varepsilon <_c 5\cdot10^{-14}$), où je pense que la notation $<_c$ est claire.

Il est bien connu des informaticiens que l'arrondi de Jules Tannery n'est pas le plus sûr, et pourtant c'est celui qui a été choisi pour la réglementation européenne sur les conversions entre l'euro et les monnaies nationales. On peut alors rêver de fortune : supposons que j'aie un ami dans le service informatique de ma banque… Par le transfert itéré (et virtuel) d'une petite somme entre mon compte et la caisse de la banque, il peut augmenter sans peine ma fortune de la façon suivante (comme je n'ai pas envie de me retrouver en prison, je donne un exemple qui fonctionne pour ma calculette à 14 chiffres et pas pour la banque à 2 chiffres après la virgule) : Si mon compte contient la somme $u_0 = 1$, en transférant N fois la somme de $0{,}5\cdot10^{-13}$ de la caisse à mon compte et réciproquement, je me retrouve à la tête de $U(N) = \sum_{0\leq k\leq 2N} u_k$ où $u_k = (-1)^{k+1}\, 0{,}5\cdot10^{-13}$ pour $k \geq 1$ . Il est clair que $U(N) = u_0$, mais on constate que $U(N) =_c u_0 + 10^{-13}\,N$, au moins pour N ne dépassant pas $9\cdot10^{-13}$. Le lecteur trouvera quelle nouvelle somme transférer pour de nouveau décupler son capital ! Ce qui s'est passé est simple : par l'arrondi Tannery, $1 + 0{,}5\cdot10^{-13}$ (calculé exactement dans l'accumulateur) devient $1 + 10^{-13}$ et $1 + 10^{-13} - 0{,}5\cdot10^{-13}$, qui est égal à $1 + 0{,}5\cdot10^{-13}$, est arrondi à $1 + 10^{-13}$. Certains peuvent avoir des scrupules à dépouiller ainsi leur banquier, mais le banquier n'est pas dépouillé, bien au contraire : s'il possède en caisse la somme $v_0 = 2$ (la banque est supposée plus riche qu'un particulier), au bout de N opérations il possède $V(N) = \sum_{0\leq k\leq 2N} v_k$ où $v_k = (-1)^k\, 0{,}5\cdot10^{-13}$ pour $k \geq 1$ , de sorte que $V(N) =_c v_0 + 10^{-13}\,N$. N'est-ce pas une manière efficace de créer des richesses ?

Redevenons sérieux. Comme, dans la suite, il s'agit seulement de garantir une majoration de l'erreur d'arrondi, je ne ferai pas de calcul très fin, mais il ne faut pas s'étonner du fait que la précision obtenue réellement soit meilleure que la précision prévue.

## 3. Comment calculer une somme partielle de série

Une conséquence immédiate du phénomène $1 + \varepsilon =_c 1$ est que, pour faire la somme de plusieurs nombres d'ordres de grandeur assez différents, il faut commencer par ajouter les termes de valeurs absolues les plus faibles si l'on veut que la calculette en tienne effectivement compte : on peut donner l'exemple trivial où l'on calcule la somme de dix termes égaux à $10^{-10}$ et d'un terme égal à 1, ce qui donne sur HP25

$$10^{-10} + \ldots + 10^{-10} + 1 =_c 1{,}000\,000\,001$$

et

$$1 + 10^{-10} + \ldots + 10^{-10} =_c 1{,}000\,000\,000.$$

Pour TI-89, il suffit de remplacer $10^{-10}$ par $10^{-14}$.

Si, comme pour le CAPES 1997, on doit calculer une somme partielle d'une série à termes positifs décroissants, il **faut** donc la calculer dans l'ordre décroissant des indices. Par exemple, avec HP25, le résultat du calcul de $\sum_{1\le k\le n} k^{-8}$ se stabilise dès que $n \ge 15$ à 1,004 077 356 si on somme pour les $k$ décroissants et à 1,004 077 355 si on somme pour les $k$ croissants. Le même phénomène apparaît aussi sur TI-89, mais pour des valeurs de $n$ plus élevées (45 pour la sommation à $k$ croissant, 55 dans l'autre sens) et la précision de TI-89 permet de vérifier que la neuvième décimale de la somme est bien 6 et pas 5. Cependant, les compensations entre arrondis par excès et par défaut peuvent masquer ce phénomène, comme le montrent les calculs suivants exécutés par HP25 (j'espère que la notation des sommes indique clairement l'ordre de sommation) :

$$\sum_{1\le k\le 100} k^{-2} - \sum_{100\ge k\ge 1} k^{-2} =_c 3\cdot 10^{-9} \text{ et } \sum_{1\le k\le 200} k^{-2} - \sum_{200\ge k\ge 1} k^{-2} =_c 2\cdot 10^{-9}.$$

Revenons au CAPES 1997 : la question 1.3.1 concerne le calcul approché de $\zeta(3/2) = \sum_{n\ge 1} n^{-3/2}$. Il s'agit de déduire sa valeur décimale à $10^{-3}$ près par défaut de l'encadrement $\zeta(3/2) = S_n + r_n$, où $S_n = \sum_{1\le k\le n} k^{-3/2}$ et $2/\sqrt{n+1} \le r_n \le 2/\sqrt{n}$, démontré au 1.2.1 pour tout nombre entier $n \ge 1$. Puis on demande s'il est raisonnable (et pourquoi) d'espérer trouver par le même procédé la valeur décimale de $\zeta(3/2)$ à $10^{-7}$ près par défaut.

Avant d'aborder la question du calcul numérique, il faut déjà dire quelques mots d'un problème strictement mathématique : un encadrement aussi précis qu'il soit du nombre réel $x$ peut très bien ne pas permettre la détermination de la valeur décimale approchée de $x$ à $10^{-n}$ près par défaut, même pour $n = 0$ : par exemple, aussi petit que soit $\varepsilon > 0$, l'encadrement $1 - \varepsilon < x < 1 + \varepsilon$ ne permet pas de dire si la partie entière de $x$ est 0 ou 1. En fait, tout dépend des décimales de $x$ : si la première décimale après la virgule de $x$ est < 5 , alors, pour $\varepsilon < 1/2$, on peut conclure, mais le cas général est celui où l'on ne connaît pas $x$, et en particulier pas sa première décimale. Plus généralement, le fait de savoir que $x$ est dans un intervalle (déterminé) d'amplitude $10^{-n}/2$ permet avec une probabilité 0,5 de déterminer sa valeur décimale approchée à $10^{-n}$ près (il suffit au lecteur qui n'est pas convaincu de faire un dessin pour voir comment un intervalle de longueur 1/2 peut se situer par rapport à $\mathbf{Z}$, puis de faire l'homothétie de rapport $10^{-n}$).

La question posée au CAPES 1997 est donc délicate et on ne peut affirmer qu'elle est soluble qu'une fois que le calcul a abouti : si, par malchance, toutes les décimales de $\zeta(3/2)$, de la quatrième à la vingtième, étaient nulles, il serait impossible à une calculette actuelle de donner les trois premières décimales (ajoutons, par prudence, en un temps décent). L'auteur du problème propose la solution suivante (Revue de Mathématiques Spéciales, 108e année, 1997-1998, 2, octobre 1997, 319-340) : choisir un $n$ (par exemple 100) pour lequel l'encadrement

$$a_n = S_n + \frac{2}{\sqrt{n+1}} \leq \zeta(3/2) \leq b_n = S_n + \frac{2}{\sqrt{n}}$$

est d'amplitude < $10^{-3}$, puis calculer les nombres $a_n$ et $b_n$ pour des valeurs croissantes de $n$ jusqu'à ce que leurs trois premières décimales coïncident (ce qui mène à 121) ; enfin un naïf calcul d'erreur d'arrondi justifie le résultat. Remarquons d'abord qu'un tel procédé est difficilement programmable sur une calculette et que la comparaison automatique des trois premières décimales de deux nombres prendrait un temps important à l'exécution. De plus, ce procédé oblige à sommer des termes décroissants, donc à perdre de la précision. Il est clair qu'il ne peut pas permettre de trouver la valeur décimale approchée à $10^{-7}$ près par défaut : comme l'auteur l'indique, il travaille avec une calculette à 12 chiffres significatifs, la précision garantie sur la somme des deux premiers termes (1 et $2^{-3/2}$) de $S_n$ ne peut donc pas être meilleure que $0{,}5 \cdot 10^{-11}$, ce qui le conduit à une erreur qu'il ne peut pas majorer par moins de $2 \cdot 10^{-7}$ pour le calcul de $S_{46\,415}$ qu'il propose.

Il est plus raisonnable de tirer à pile ou face : on choisit un $n$ tel que $b_n - a_n$, qui est $< 1/\left(n\sqrt{n}\right)$ d'après un calcul immédiat, soit < $10^{-3}/2$ (par exemple $n = 160$), on calcule $S_n$ en commençant par les plus petits termes de la somme, puis $a_n$ et $b_n$ et on constate avec soulagement qu'un naïf calcul d'erreur d'arrondi garantit le résultat $\zeta(3/2) = 2{,}612...$ (l'erreur sur chaque terme est au plus $10^{-10}$ pour HP25, et il y a 159 additions pour le calcul de $S_n$, donc l'erreur d'arrondi ne dépasse pas $10^{-7}$).

Pour déterminer la valeur décimale approchée à $10^{-7}$ près par défaut, ce n'est pas l'erreur d'arrondi qui est la plus gênante (même pour une calculette à 10 chiffres) si l'on calcule $S_n$ dans le sens des $k$ décroissants. Oublions le fait qu'un encadrement, aussi bon soit-il, peut ne pas permettre d'aboutir, jouons encore à pile ou face en choisissant $n$ tel que $b_n - a_n$ soit $< 10^{-7}/2$, par exemple $n = 8{\cdot}10^4$, et calculons $S_n$ dans le sens des $k$ décroissants. Le calcul naïf de l'erreur ne suffit plus, mais il suffit de tenir compte du sens de sommation : pour $k > 10^4$, on a $k^{-3/2} < 10^{-6}$, donc l'erreur d'arrondi de HP25 sur le dixième chiffre significatif est au plus de $10^{-16}$ (et je suis large dans l'estimation : je pourrais dire la moitié de cela), cette erreur cumulée sur $7{\cdot}10^4$ termes donne une erreur d'au plus $7{\cdot}10^{-12}$. Un calcul analogue montre que, pour $10^4 \geq k > 10^2$, l'erreur cumulée est au plus de $10^{-9}$, et pour les 100 premiers termes l'estimation naïve est de $10^{-8}$. La calculette donne donc $a_n$ et $b_n$ avec une erreur d'au plus $1{,}2{\cdot}10^{-8}$, et il est raisonnable d'espérer aboutir.

Le reproche naturel qu'on peut faire à cette méthode consiste à dire que, si elle a échoué, comme on somme à $k$ décroissants, le calcul de $S_n$ ne peut pas être utilisé pour calculer, par exemple, $S_{2n}$. Mais ce reproche est infondé car l'erreur faite sur le calcul de $S_{2n}$ - $S_n$ = $\sum_{n+1\leq k\leq 2n} k^{-3/2}$ est négligeable devant l'erreur faite sur $S_n$, si $n$ est assez grand. En effet, nous venons de voir que la contribution principale à l'erreur d'arrondi provient des termes de petit indice ; ainsi, pour HP25 et pour n = $8{\cdot}10^4$, l'erreur sur $S_{2n}$ - $S_n$ est majorée naïvement par $8{\cdot}10^{-12}$ de sorte que l'erreur sur $S_{2n}$ est encore majorée par $1{,}2{\cdot}10^{-8}$. Le calcul de $a_{2n}$ et $b_{2n}$ fournit donc un encadrement d'amplitude majorée par $\frac{1}{2}\left(10^{-7}/\left(2\sqrt{2}\right)+2,4\cdot 10^{-8}\right)$, donc au plus égale à $2{,}6{\cdot}10^{-8}$, et la valeur décimale cherchée est obtenue avec une probabilité d'au moins 1 - 0,26, soit 74%. Il faut bien voir que le phénomène $1 + \varepsilon =_c 1$ n'est vraiment gênant que s'il se produit à de multiples reprises dans le calcul numérique.

La seule raison de désespérer de la méthode est le temps : ma vieille HP25 a mis 3 minutes et 20 secondes pour calculer $S_{160}$, il faut donc prévoir plus de 27 heures pour $S_{80\,000}$ et l'épreuve de CAPES dure 5 heures… Mais, pour une calculette plus moderne, la situation est différente : la calculette de mon fils a mis 2 heures et 45 minutes pour calculer $S_{80\,000}$, ce qui n'est pas complétement rédhibitoire mais empêche, pendant un bon moment, le candidat d'utiliser sa calculette pour répondre aux questions suivantes. Le calcul d'arrondi montre que l'encadrement ainsi obtenu permet de garantir la septième décimale $\zeta(3/2) = 2,\ 612\ 375\ 3\ldots$ Il est vrai que l'on n'avait qu'une chance sur deux d'aboutir, mais l'énoncé du CAPES, écrit pour une calculette à 12 chiffres, aurait été plus prudent de parler de la précision $10^{-9}$

plutôt que $10^{-7}$. Quant à TI-89, elle fait le calcul en 40 minutes, sans aucun travail de programmation puisqu'une fonction sommation existe dans le menu de calcul (elle calcule la somme dans le sens des indices croissants, donc il faut lui faire sommer les $(n - k)^{-3/2}$ de $k = 0$ à $k = n - 1$), et même l'estimation naïve de l'erreur montre que la précision obtenue est plus que suffisante.

Dans la suite du problème, l'encadrement du terme reste $r_n$ est affiné, ce qui permet de calculer rapidement la valeur décimale approchée de $\zeta(3/2)$ à $10^{-7}$ près par défaut. L'estimation naïve de l'erreur d'arrondi faite sur le calcul de $S_{450}$ est $4{,}5 \cdot 10^{-8}$, ce qui ne permet pas d'espérer garantir la précision demandée. Il faut donc un calcul du type précédent et une sommation *dans le bon sens*, du moins pour une calculette à 10 chiffres significatifs (si on dispose de 12 chiffres significatifs, il n'y a plus de difficulté). Le calcul de la valeur décimale approchée à $10^{-6}$ près par défaut de la constante d'Euler, demandé au 1.3.4.b, s'obtient sans précaution particulière. Remarquons enfin que les termes à sommer $k^{-3/2}$ ou $k^{-1}$ (pour $k$ entier naturel non nul) ont une expression très simple qui justifie le fait d'admettre que la calculette les donne avec au plus une erreur de 1 sur le dernier chiffre significatif (et la retenue éventuelle qui peut se propager vers les chiffres antérieurs).

## 4. Une ruse remarquable

Un artifice de programmation permet de calculer la somme $S_n = \sum_{1 \le k \le n} u_k$ avec une excellente précision, sans tenir compte du sens de variation de la suite des $|u_k|$ dans l'ordre de sommation, mais au prix d'une augmentation non négligeable du temps de calcul. Cet algorithme est présenté par Claude Brezinski dans ses deux petits livres (*Introduction à la pratique du calcul numérique*, Dunod 1988 et *Algorithmique numérique*, Ellipses 1988) et il est dû à Madame M. Pichat (Correction d'une somme en arithmétique à virgule flottante, Numer. Math., 19 (1972), 400-406). Un algorithme voisin se trouve dans l'exercice 19 de la section 4.2.2 du livre de D. E. Knuth. Le principe en est très simple : à chaque pas du calcul, quand on ajoute $u_{k+1}$ à $S_k$, on évalue l'erreur d'arrondi $e_{k+1}$ commise dans cette opération :

$$e_{k+1} =_c \begin{cases} (-S_{k+1} + S_k) + u_{k+1} & \text{si } |S_k| \ge |u_{k+1}|, \\ (-S_{k+1} + u_{k+1}) + S_k & \text{sinon.} \end{cases}$$

On ajoute en fin de calcul la somme de ces erreurs d'arrondi à la valeur calculée de $S_n$, et on obtient une bien meilleure approximation de la somme cherchée. La théorie dépasse le cadre de cette petite note, mais il suffit de faire le calcul pas à pas, par exemple dans le cas d'une des sommes étudiées au paragraphe précédent, pour comprendre le fonctionnement et l'efficacité de cet algorithme : le calcul de $e_{k+1}$ est étudié pour supprimer le phénomène $1 + \varepsilon =_c 1$ et donne donc une bonne approximation de l'erreur commise à chaque pas de la sommation. De là à donner une estimation raisonnable de l'erreur commise sur $S_n$, il y a un pas théorique que franchiront peut-être les candidats au futur (et éventuel) CAPES de Mathématiques et Informatique.

Ce que prouve la comparaison des performances de ma vieille HP25 et de celles des calculatrices de la dernière génération, c'est qu'il est de plus en plus difficile de poser des questions pertinentes de calcul numérique dans une épreuve écrite de CAPES, ou même de Baccalauréat, c'est-à-dire des questions testant les connaissances et la réflexion des candidats (au petit niveau de ce que j'explique ici) et non la qualité de leurs instruments de calcul. On risque de s'acheminer vers l'abandon pur et simple de telles questions intéressantes (le sujet de CAPES 1998 sur les suites récurrentes aurait pu donner lieu à un calcul numérique, mais ne l'a pas fait) ou vers des questions autrement plus difficiles d'algorithmique.