BIPM. 70.2

P. CARRÉ

Note sur les sous-programmes de Précision Etendue Améliorée, Version 1969.

Complément à "Travaux du mois de décembre 1969"

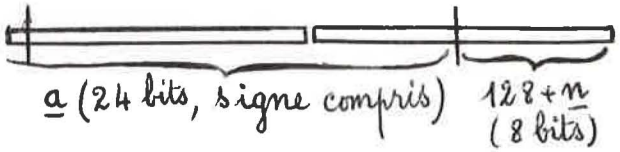
I. Rappels. Représentation des nombres en virgule flottante

Tout nombre x peut se mettre sous la forme

$$x = a \times 2^n \quad \begin{cases} 0{,}5 \leqslant a < 1 \\ -1 < a \leqslant -0{,}5 \end{cases}$$

Dans les sous-programmes de calcul fournis par IBM on représente en mémoire a et n selon l'un ou l'autre des modèles ci-dessous.

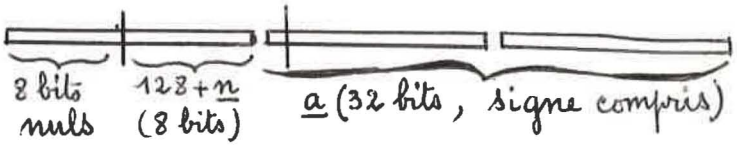
1/ Précision normale



Le 1er bit de a donne le signe, le suivant a pour poids 2^{-1}, le 24e a pour poids 2^{-23}. On voit donc que a ne peut varier que par sauts de 2^{-23} soit en valeur relative 2^{-22} si $a \approx 0{,}5$ et 2^{-23} si $a \approx 1$; mais un arrondi correct du dernier bit gardé pourrait permettre une précision de représentation deux fois meilleure.

n a pour valeurs extrêmes - 128 et + 127

2/ Précision étendue



On voit immédiatement que le 32e bit de a a une valeur relative comprise entre 2^{-30} (soit 10^{-9}) et 2^{-31} (soit $10^{-9,3}$).

La précision de représentation et, dans une certaine mesure, la précision de calcul pourraient être 2 fois meilleures au moyen d'un arrondi correct du dernier bit mais cela n'est pas garanti. On doit donc compter avec des erreurs relatives de 10^{-9} dans chaque multiplication ou division (L'erreur relative dans une addition ou une soustraction est, évidemment, fonction des données).

Les limites de variation de n ne posent pas de problème, le nombre maximal que l'on peut représenter étant pratiquement $2^{127} \approx 10^{38,2}$ et le plus petit nombre positif non nul étant $2^{-128} \times \frac{1}{2} \approx 10^{-38,8}$.

II. But et caractères généraux de la précision étendue améliorée (P.E.A.)

On voit donc, en gros, que l'on dispose de sous-programmes à 10^{-7} et de sous-programmes à 10^{-9}, les uns ou les autres pouvant être appelés par les ordres Fortran habituels (mais pas les deux, dans un même programme). Or, nous faisons certaines expériences à mieux que 10^{-8} près. Il n'est pas tolérable que des années d'effort expérimental soient anéanties par quelques opérations consécutives, chacune introduisant une erreur de 10^{-9}.

IBM diffuse des sous-programmes à 10^{-18} mais, malgré les améliorations que je leur ai apportées,

1°/ ils sont lents ;
2°/ ils ne comportent que les 4 opérations élémentaires ;
3°/ ils consomment beaucoup de place en mémoire (pour s'y loger et pour loger les variables) ;
4°/ ils sont utilisables en Fortran au moyen des ordres d'appel explicite des sous-programmes et non au moyen des formules de sorte qu'on peut programmer en Fortran tout sauf les formules ;
5°/ enfin leur précision est surabondante pour longtemps encore.

Il m'a donc semblé utile de faire des sous-programmes ayant les caractères suivants :

1°/ bonne rapidité ;
2°/ comportant les 4 opérations, la racine carrée, le logarithme, l'exponentielle, l'arc tangente et le sinus-cosinus ;
3°/ ne prenant pas une place exagérée en mémoire ;
4°/ utilisables en Fortran sans modification des ordres, y compris ceux d'entrées-sorties ;
5°/ précision égale à la précision théorique.

La solution adoptée a consisté à récupérer les 8 bits perdus à gauche du 1er mot pour y mettre les bits 33 à 40 de $\underline{a}$.

En arrondissant correctement lors de chaque opération (entrée comprise) le dernier bit conservé, on peut réduire l'erreur de représentation et l'erreur de calcul à une valeur au plus égale à 2^{-40} sur $\underline{a}$, d'où une erreur relative comprise entre 2^{-39} (soit $10^{-11,7}$) et 2^{-40} (soit 10^{-12}).

En contre-partie, on remplace des opérations portant sur deux mots-machine par des opérations portant sur trois parmi lesquels un peu plus de deux et demi seulement sont significatifs. L'expérience montre que ce petit demi-mot libre contenant initialement l'exposant donne beaucoup de souplesse à la programmation.

III. Caractères particuliers du sous-programme d'entrées et sorties

. Pour mettre au point des programmes de calcul il faut pouvoir fournir des nombres à la machine et il est commode qu'elle puisse imprimer les résultats.

. J'ai donc d'abord écrit des sous-programmes provisoires d'entrée de nombres (par cartes perforées) et d'impression de nombres (sur la machine à écrire) compatibles avec la précision souhaitée.

. Lorsque les premiers sous-programmes de calcul furent mis au point (addition, soustraction, multiplication, division, racine carrée, logarithme, exponentielle) j'ai étudié puis adapté à la précision obtenue dans le calcul le sous-programme I.B.M. qui gère toutes les entrées-sorties en Fortran. Dès lors, il fut possible de faire fonctionner en P.E.A. des programmes précédemment écrits, sans leur apporter aucune modification.

. En outre ce sous-programme diffère du sous-programme original par les points suivants :

- en entrée, il accepte indifféremment le point ou la virgule, en format E ou F ;
- en sortie, format F, il réalise l'arrondi automatique†dans le format et il assure l'impression des nombres avec une virgule, d'ailleurs supprimée et remplacée par un blanc si le nombre de décimales spécifié est zéro.

†conformément au nombre de décimales spécifié

IV. Principes mathématiques des sous-programmes arithmétiques et fonctionnels.

Pour tous ces sous-programmes, j'ai cherché un compromis entre la rapidité d'exécution et l'encombrement de la mémoire.

1°/ Addition

Le principe est le suivant, en supposant $n' \leqslant n$

$$a \times 2^{n} + a' \times 2^{n'} = (a + a'/2^{n-n'}) \times 2^{n}$$

d'où les opérations à effectuer :

- séparation des exposants des deux nombres ;
- division par $2^{n-n'}$, c'est-à-dire décalage à droite de n-n' positions, du nombre qui n'a pas le plus grand exposant ;
- addition (avec reports éventuels) ;
- contrôle de dépassement de capacité ;
- normalisation ;
- réunion de l'exposant avec arrondi binaire.

La durée de l'addition est 1,05 ms.

2°/ Soustraction

On se ramène à l'addition par un changement de signe après séparation de l'exposant.

La durée de la soustraction est 1,25 ms.

3°/ Multiplication

Principe : $a \times 2^n \times a' \times 2^{n'} = aa' \times 2^{n+n'}$.

Le produit aa' fait intervenir 9 produits partiels dont 3 sont négligeables. Ces produits partiels étant en double longueur, alors que le produit est en triple longueur, il faut bien faire attention dans le cas de facteurs négatifs. Voici les principales étapes, sans insister autrement sur les détails:

- séparation des exposants ;
- calcul et sommation des 6 produits partiels ;
- calcul du nouvel exposant ;
- réunion de l'exposant avec arrondi binaire.

La durée de la multiplication est 1,95 ms.

4°/ Division

Principe $\frac{a \times 2^n}{a' \times 2^{n'}} = \frac{a}{a'} \times 2^{n-n'}$.

Malheureusement, le quotient ne peut s'exprimer simplement en fonction de "quotients partiels".

Toutefois, les 3 mots C, c et γ représentant a et les 3 mots D, d et δ représentant a' ont des poids proportionnels à 1, 2^{-16}, 2^{-32}, on peut donc appliquer une technique de développement limité et utiliser les opérations câblées (multiplication et division). Les difficultés proviennent des risques de dépassement de capacité et aussi des signes. Mais les opérations câblées sont algébriques, on ne doit pas rencontrer de difficulté de mise au point insurmontable.

En posant $m = 2^{16}$, on a :

$$Q = \frac{a}{a'} = \frac{C + c/m + \gamma/m^2}{D + d/m + \delta/m^2} = \frac{Cm^2 + cm + \gamma}{Dm^2 + dm + \delta}$$

En posant successivement :

$$\begin{aligned} Cm + c &= Dq + r \\ rm + \gamma &= Dq' + r' \\ dm + \delta &= Dq'' + r'' \\ r'm - qr'' &= Dq''' + \rho \end{aligned}$$

on établit, avec une erreur inférieure à m^{-3} :

$$Q = \frac{1}{m^3} \left\{ qm^2 + (q'm - qq'') + q''' - \frac{q'm - qq''}{m^2} q'' \right\}$$

Tel est le développement limité utilisé, calculé en virgule fixe et suivi par une normalisation et une réunion du nouvel exposant.

La durée de la division est 1,95 ms.

5°/ Racine carrée

Principe : $\sqrt{a \times 2^n} = \sqrt{a' \times 2^{2p}} = \sqrt{a'} \times 2^p$

avec $\begin{cases} a' = a \\ 2p = n \end{cases}$ si n est pair

$\begin{cases} a' = a/2 \\ 2p = n + 1 \end{cases}$ si n est impair

On voit donc que $0{,}25 \leqslant a' < 1$

D'autre part, si $\alpha = \sqrt{a'}\ (1 + \varepsilon)$ est une valeur approchée de la racine carrée,

$\frac{1}{2}\left(\alpha + \frac{a'}{\alpha}\right) \approx \sqrt{a'}\ \left(1 + \frac{\varepsilon^2}{2}\right)$ est une bien meilleure approximation.

C'est la méthode de Newton bien connue.

Dans notre cas, le plus économique semble être d'appliquer deux fois la méthode de Newton. Si ε est l'erreur relative de la valeur de départ, $\frac{\varepsilon^4}{8}$ sera l'erreur après 2 applications de la méthode de Newton.

Pour avoir $\frac{\varepsilon^4}{8} < 10^{-12}$ il faut $\varepsilon < \sqrt[4]{8} \times 10^{-3} \approx 1{,}7 \times 10^{-3}$.

Dans chacun des deux sous-intervalles $0{,}25 \leqslant a' < 0{,}5$ et $0{,}5 \leqslant a' < 1$ une fonction homographique donne une précision bien suffisante moyennant 2 additions et une division, un polynôme du second degré donne une précision moins bonne mais encore meilleure que 10^{-3}, donc suffisante, moyennant 2 additions et 2 multiplications. La division (câblée) dure plus longtemps que deux multiplications ; j'ai donc adopté le polynôme du second degré. Son calcul est fait en simple longueur. La 1re application de la méthode de Newton donne un résultat en double longueur. La 2e application donne le résultat cherché. De plus, on remplace les moyennes par des sommes, grâce à un cadrage approprié et ces sommes sont simplifiées du fait que le mot de droite de l'un des deux termes est nul. Le résultat est obtenu automatiquement normalisé.

La durée du calcul de racine carrée est 2,05 ms.

Le sous-programme I.B.M., 500 fois moins précis exige 10,4 ms.

6°/ Logarithme népérien

Principe : $\mathrm{Ln}\ (\underline{a} \times 2^n) = \mathrm{Ln}\ \underline{a} + \underline{n}\ \mathrm{Ln}\ 2$

On a toujours $0{,}5 \leqslant \underline{a} < 1$

On déduit l'intervalle de variation de $\underline{a}$ et on le centre (géométriquement) par rapport à 1 en multipliant $\underline{a}$ par l'un des coefficients

$2^{7/8}\ \ 2^{5/8}\ \ 2^{3/8}\ \ 2^{1/8}$ selon sa position par rapport aux nombres

$2^{-1}\ \ 2^{-3/4}\ \ 2^{-1/2}\ \ 2^{-1/4}\ \ 1.$

En désignant par 2^c celui des coefficients ci-dessus qui nous intéresse, la formule devient

$$\mathrm{Ln}(\underline{a} \times 2^n) = \mathrm{Ln}(2^c \times \underline{a}) + (n - c)\ \mathrm{Ln}\ 2.$$

On passe alors à la variable z définie par

$$2^c \times \underline{a} = \frac{1 + z}{1 - z}$$

On constate que l'intervalle de variation de z est environ $-0{,}043 \leqslant z \leqslant 0{,}043$. On utilise un développement de $\mathrm{Ln}\ \frac{1 + z}{1 - z}$ (qui ne contient que les puissances impaires de z, donc rapidement convergent) en fraction continue. Pour obtenir les 12 chiffres significatifs désirés j'utilise la 4e réduite qui s'écrit :

$$\mathrm{Ln}\ \frac{1 + z}{1 - z} = \cfrac{2z}{1 - \cfrac{z^2}{3 - \cfrac{4z^2}{5 - \cfrac{9z^2}{7}}}}$$

Le terme $\frac{9z^2}{7}$ ne joue de rôle appréciable que vers les extrémités de l'intervalle de variation de z. J'ai pu le remplacer par une valeur constante correspondant à $z \approx 0{,}041$, c'est-à-dire en fait remplacer 4/5 par $4/5 + 23 \times 2^{-16}$ sans perdre de précision.

Remarque : le repérage de l'intervalle contenant $\underline{a}$ se fait en simple précision. Pour la fraction continue, on commence en précision simple puis double, puis triple ; n - c est calculé comme différence de nombres entiers en prenant comme unité 1/8. Les autres opérations utilisent la précision maximale.

La durée du calcul du logarithme népérien est 11,9 ms.

7°/ Exponentielle

Principe $e^x = 2^{x/\mathrm{Ln}\,2}$

Posons E = partie entière de $\frac{x}{\mathrm{Ln}\,2}$

$$F = \frac{x}{\mathrm{Ln}2} - E \qquad 0 \leqslant F < 1$$

$$F' = F\mathrm{Ln}2 = x - \mathrm{Ln}2 \times E \qquad 0 \leqslant F' < \mathrm{Ln}2$$

ou $x = F' + \mathrm{Ln}2 \times E$

On a donc: $e^x = 2^{F'/\mathrm{Ln}2} \times 2^E = 2^E \times e^{F'}$

Considérons alors les 4 intervalles définis par les bornes

$$0 \; ; \; \frac{1}{4}\mathrm{Ln}2 \; ; \; \frac{1}{2}\mathrm{Ln}2 \; ; \; \frac{3}{4}\mathrm{Ln}2 \; ; \; \mathrm{Ln}2$$

auquels nous associons les valeurs i = 0 i = 1 i = 2 i = 3

Posons $z = F' - (i + \frac{1}{2}) \times \frac{1}{4}\mathrm{Ln}2$ alors $-\frac{1}{8}\mathrm{Ln}2 \leqslant z < \frac{1}{8}\mathrm{Ln}2$

soit $-0{,}086 \leqslant z < 0{,}086$

On a finalement

$$e^x = 2^E \times 2^{\frac{2i+1}{8}} \times e^z.$$

Une fois E et i déterminés, il faut calculer F' puis z avec toute la précision possible. Le terme 2^E est très simplement représenté sous la forme habituelle $\underline{a} \times 2^n$ avec $\underline{a} = 0{,}5$ $n = E + 1$; les 4 valeurs de

$2^{(2i+1)/8}$, soit $2^{1/8}$ $2^{3/8}$ $2^{5/8}$ $2^{7/8}$ sont conservées en mémoire, d'ailleurs on les utilise déjà pour le logarithme. Il reste à calculer e^z.

La relation $e^z = \frac{1}{e^{-z}}$ conduit à utiliser un développement de la forme $\frac{P(z)}{P(-z)}$.

On pourrait penser prendre pour P(z) le développement de $e^{z/2}$ en série de Mac-Laurin ; mais il y a des polynômes beaucoup plus économiques.

On obtient la précision souhaitée avec :

$$P(z) = 120 + 60z + 12z^2 + z^3 \quad \text{pour} \quad |z| < \frac{\mathrm{Ln}2}{8} .$$

Si on exprime le quotient des polynômes sous une forme analogue à une fraction continue, on remplace des multiplications par un nombre moindre de divisions. On peut donc hésiter sur la méthode. Mais la fraction continue contient des termes du type c/z dont l'ordre de grandeur varie énormément de sorte que l'on est pratiquement obligé de travailler en virgule flottante.

Au contraire, les polynômes ont un ordre de grandeur invariable, on peut donc travailler en virgule fixe.

Evidemment on calcule

$$P(z) = ((z + 12)z + 60)\, z + 120$$
$$-P(-z) = ((z - 12)z + 60)\, z - 120$$

et on fait le quotient en virgule fixe.

La durée du calcul de l'exponentielle est 11,5 ms.

8°/ Arc tgte

On partage le domaine de variation de l'argument x $(-\infty, +\infty)$ en un nombre impair d'intervalles. J'ai choisi 9. Les limites en sont donc :

$$-\infty \quad tg\left(-\frac{7\pi}{18}\right) \quad tg\left(-\frac{5\pi}{18}\right) \quad tg\left(-\frac{3\pi}{18}\right) \quad tg\left(-\frac{\pi}{18}\right)$$
$$tg\left(\frac{\pi}{18}\right) \quad tg\left(\frac{3\pi}{18}\right) \quad tg\left(\frac{5\pi}{18}\right) \quad tg\left(\frac{7\pi}{18}\right) \quad +\infty .$$

On cherche d'abord si x est compris dans l'intervalle central. Si oui on passe au calcul. Si non, si x est négatif on change le signe, ce qui ramène au cas x positif. On repère alors dans lequel des 4 intervalles possibles x se trouve (en simple précision bien sûr). Désignons par α l'arc correspondant au milieu de cet intervalle et par y l'arc cherché.

$$tg\,(y-\alpha) = \frac{tgy - tg\alpha}{1 + tgy.tg\alpha} = \frac{x - tg\alpha}{1 + x\,tg\alpha} = \frac{1}{tg\alpha} - \frac{1 + \frac{1}{tg^2\alpha}}{x + \frac{1}{tg\alpha}} = u .$$

Cette formule est économique en opérations. Malheureusement, dans l'intervalle

$$+\frac{\pi}{18} \quad \text{à} \quad +\frac{3\pi}{18} ,$$

le premier terme est de l'ordre de 3, la différence des deux termes ne dépassant pas $\pm$ 0,176. On perd donc de la précision.

J'ai finalement adopté le changement de variable

$$tg\,(y - \alpha) = \frac{\frac{x}{tg\alpha} - 1}{\frac{1}{tg\alpha} + x} = u$$

qui comporte une multiplication de plus que le précédent mais on gagne en précision et il suffit de mémoriser les 4 valeurs de $1/tg\alpha$.

Les valeurs utilisées sont :

$$1/tg\,\frac{\pi}{9} = 2,7474774194539$$
$$1/tg\,\frac{2\pi}{9} = 1,1917535925925$$

$$1/\mathrm{tg}\,\frac{\pi}{3} = 0{,}5773502691917$$

$$1/\mathrm{tg}\,\frac{4\pi}{9} = 0{,}1763269807089\ .$$

On est donc ramené à :

$$z = y - \alpha = \mathrm{Arc\ tg}\ u.$$

Ainsi $|u| \leqslant \mathrm{tg}\,\frac{\pi}{18} \approx 0{,}176.$

J'ai choisi de calculer z par un développement polynômial (polynôme de Tchebichef plus économique que la série de Mac-Laurin).

$$z = \mathrm{Arc\ tg}\ u = ((((((a_{13}u^2+a_{11})u^2+a_9)u^2+a_7)u^2+a_5)u^2+a_3)u^2+a_1)u$$

avec
$a_1 = 1{,}000\ 000\ 000\ 000\ 0$
$a_3 = -0{,}333\ 333\ 333\ 330\ 6$
$a_5 = 0{,}199\ 999\ 999\ 354\ 0$
$a_7 = -0{,}142\ 857\ 043\ 604\ 2$
$a_9 = 0{,}111\ 103\ 259\ 178\ 2$
$a_{11} = -0{,}090\ 575\ 046\ 111\ 6$
$a_{13} = 0{,}069\ 610\ 657\ 817\ 5$

Le calcul est fait en virgule fixe, sauf le changement de variable initial, la multiplication par u et le changement de variable final $y = z + \alpha$.

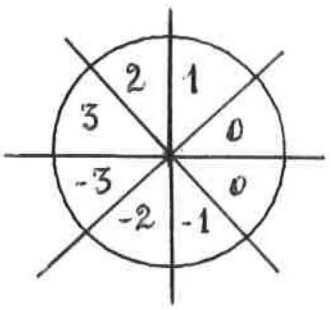
La durée du calcul est 9,5 ms pour l'intervalle central (pas de changement de variable), 17,3 ms pour les autres intervalles.

9°/ Sinus et cosinus

J'utilise un développement valable de $-\frac{\pi}{4}$ à $+\frac{\pi}{4}$ ce qui permet d'avoir un organigramme relativement simple.

L'angle x, exprimé en radians est converti en tours par l'opération $z = x/2\pi$.

L'angle z' exprimé en huitièmes de tour est obtenu du même coup puisqu'il a la même représentation binaire. Sa partie entière est cadrée entre - 4 et + 4 par des opérations binaires simples. La valeur - 4 peut être atteinte exceptionnellement. Elle est traitée spécialement. On passe des domaines i = 2, 3, - 3, - 2 aux domaines 1, 0, 0, - 1 par le changement de variable $3\ \frac{i}{|i|} - z'$. Cela correspond à un changement de signe pour le cosinus mais non pour le sinus.



On passe des domaines i = 1 ou - 1 au domaine 0 par le changement de variable $\frac{i}{|i|} - z'$. Cela correspond à une permutation de sinus et cosinus et une multiplication par $\frac{i}{|i|}$.

On lance alors le calcul. Les développements utilisés sont des développements en polynômes de Tchébichef que l'on écrit :

$$\cos x = ((((a_{10}z^2+a_8)z^2+a_6)z^2+a_4)z^2+a_2)z^2+a_0$$

$$\frac{\sin x}{x} = ((((a_{11}z^2+a_9)z^2+a_7)z^2+a_5)z^2+a_3)z^2+a_1 .$$

Les sous-programmes de précision 10^{-18} m'ont permis de calculer ces coefficients, qui apparaissent comme des sommes d'intégrales calculables. Voici leurs valeurs approximatives en numération décimale.

a_0 =	1,000 000 000 000	a_1 =	6,283 185 307 184
a_2 =	- 19,739 208 801 941	a_3 =	- 41,341 702 240 170
a_4 =	64,939 393 832 581	a_5 =	81,605 249 152 286
a_6 =	- 85,456 765 173 469	a_7 =	- 76,705 829 970 771
a_8 =	60,238 201 227 505	a_9 =	42,055 286 971 852
a_{10} =	- 26,058 582 414 058	a_{11} =	- 14,910 094 204 330

Tout cela est fait en virgule fixe, à partir du calcul de z. Une difficulté de cadrage se présente pour les angles $\pm \frac{\pi}{4} = \pm$ 1 huitième de tour qui appartiennent aux domaines 1 ou - 1 mais y restent lorsqu'on prend le complément. Ces cas sont aisément reconnus et traités à part.

Il reste à effectuer un changement de signe éventuel, passer à la représentation virgule flottante, et dans le cas du sinus, multiplier par z'/8.

Durée du calcul 9,5 ms.

10°/ <u>Tangente hyperbolique et puissance fractionnaire</u>

Ces deux sous-programmes, nécessaires pour un fonctionnement correct en Fortran sont en fait très brefs puisqu'ils utilisent les sous-programmes précédemment décrits.

en effet $\mathrm{Th}\ x = \frac{e^{2x} - 1}{e^{2x} + 1}$

Le calcul dure 16 ms.

Si $x < 2^{-13}$ il est plus précis de poser Thx = x. La durée est alors négligeable.

Le calcul de A^B se ramène immédiatement au calcul d'un logarithme et d'une exponentielle puisque

$$A^B = \exp(B.\mathrm{Ln}A)$$

6 janvier 1970