BiPM/68-2

P. CARRE

Note complémentaire sur quelques sous-programmes de précision étendue dite "améliorée".

Complément à "Travaux du mois de février 1968"

Cette note complète le "Complément" à "Travaux du mois de janvier 1968" et lui apporte quelques rectifications.

III. 2° . division A la dernière ligne de ce paragraphe, remplacer 2,4 ms par 2,1 ms.

. racine carrée

La durée du calcul est 3,2 ms. Le programme IBM, 500 fois moins précis, exige 10,4 ms d'après la notice.

. logarithme népérien

La formule donnée en haut de la page 7 est fausse. Il faut lire

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

Cette erreur de l'ouvrage "méthodes mathématiques pour calculateurs arithmétiques" m'a retardé d'un ou deux jours. 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 remplacer 5 par 5 - 0,0022 ou plus exactement par $5 - 9 \times \cancel{10}^{-12}$ 2 sans perdre de précision.

Le calcul est effectué en 18 ms.

Le programme IBM ne demande que 8 ms (je l'ai effectivement vérifié). Ce rapport 2,25 est un peu grand à mon gré).

J'espérais ne pas dépasser 2, qui serait conforme à la proportionnalité au cube du nombre de chiffres exacts fournis par le sous-programme.

L'astuce déployée par IBM pour ce sous-programme semble donc bien supérieure à celle déployée pour la racine carrée.

. exponentielle

Je n'ai pas cherché beaucoup la rapidité pour ce sous-programme dont l'exécution demande 36 ms.

Afin de réduire la durée de mise au point, j'ai programmé tous les calculs en virgule flottante et avec la précision maximale. On peut se demander si l'utilisation de la fraction continue qui exige 4 additions et 3 divisions est vraiment plus rapide que le calcul direct des deux polynômes P(z) et P(-z) et de leur quotient, ce qui exigerait 2 (2 multiplications + 3 additions) + 1 division = 1 division + 4 multiplications + 6 additions. La différence ne pourrait être que la durée de 2 divisions - 4 multiplications - 2 additions. Même en tenant compte du fait que notre machine a une division (câblée) plutôt lente, ce qui se répercute sur la division en virgule flottante, on peut voir que l'on ne pourrait pas gagner ainsi plus de 1,4 ms. De toute façon, je n'envisage de chercher une amélioration que si cela se révèle vraiment utile, ce qui me semble extrêmement peu probable. Il va sans dire que la précision souhaitée est obtenue.

A titre indicatif, le programme IBM dure 4,4 ms, d'après la notice.

IV. Conclusion

En principe il faudrait faire encore les sous-programmes de lignes trigonométriques. Toutefois, j'y renonce pour le moment. Lorsqu'on fera des transformées de Fourier et que la précision actuellement disponible sera insuffisante je reconsidèrerai peut-être la question.

Enfin, il y a encore pas mal de travail pour que les sous-programmes qui existent soient utilisables automatiquement en Fortran. J'aimerais savoir s'il est utile de l'entreprendre.

27.2.68