数理解析研究所講究録
第 359 巻 1979 年 1-14

IFIP W.G. 2.5 の報告
— 特に新浮動小数点標準体系について —

京都大学 数理解析研究所 一松 信

I. IFIP W.G. 2.5 の大会について

1. IFIP（国際情報処理連合）の技術委員会(TC)は現在TC-10 まであり，プログラミングに関するTC-2には，委員会(W.G.) が 2.7 まである．言語関係の 2.1, 2.2 は日本での関心も高いが，数値計算に関する 2.5 は，1975年に数学的ソフトウェアに改組拡充されて以後，毎年委員会を開催してきたにもかかわらず，日本から誰も出席していなかった．もちろんそれには，手が廻りきれないというお家の事情があったらしいが，1977年の Toronto での IFIP 総会の折に，日本からの出席を強く要請され，1978 年12月11日(月)～15日(金)にオーストリヤ，Wien 郊外の Baden で開催される拡大大会（Working Conference）に筆者が出席し，パネル討論の折に日本での現状を報告した．

この準備のため，1978 年10月には短期協同の研究集会を開催し，数多くの貴重な資料をいただいたが，けっきょく報

告できたのは、東大大型センターのサブルーチン使用統計のみであった。正直いって筆者の了解に若干のくい違いがあり、非常に有益ではあったが、いま一つ十分意をつくさない点があった。── 具体的には、本研究所の研究集会や講究録に発表された興味深い業績が、その後どの雑誌に正式に発表されているかといった追跡調査の不備であった。── 現在この種の非公式情報の比重は増しているが、やはり外国の諸学者をなっとくさせるには、国際共通語で書かれた公式の論文を引用しなければ通用しないことを思い知らされた。

2. 以下にこの大会でのプログラムと、若干の感想を記す。なお報告集が North-Holland 社から刊行されている。

11日（月） 午前、開会式（経過の説明；事務上の注意）；続いて

General Aspects of Performance Evaluation I.

T.E.Hull, Correctness of numerical software.

W.S.Brown, Some fundamentals of performance evaluation for numerical software.

T.J.Dekker, Correctness proof and machine arithmetic.

午後、同上 II

J.Lyness, Performance profile and software evaluation.

T.L.Jordan, A performance evaluation of linear algebra software in parallel architectures.

L.D.Fosdick, Detecting errors in software.

同日夜は懇親のためのカクテル・パーティー

12日(火)午前. Performance Evaluation in Linear Algebra.

I. Moichanov, Performance evaluation of linear algebra software.

B.T.Smith, Can automated theorem provers be used to evaluate linear algebra software?

I.S.Duff & J.K.Reid, Performance evaluation of codes for sparse matrix problems.

午後. Reliability and Warranty of Numerical Software.

B.Niblett, Legal aspects of numerical software

E.L.Battiste, Reliable-Warrantable code

C.F.H.Tapper, Legal remedies for misrepresentation of software

同日夜はWienの芸術の夜(オペラ,音楽会,博物館など).

13日(水)(午前) Performance Evaluation in Ordinary Differential Equations

H.J.Stetter, Performance evaluation of O.D.E. software through modelling.

P.J.van der Houwen & J.G.Verwer, Comparison of algorithms for systems of O.D.E.'s originating from Parabolic I.B.V.P. in 2 dimensions

W.H.Enright, Using a testing package for the automatic assessment of numerical metbods for O.D.E.'s.

同日午後は遠足，夜は晩餐会のため，会議は休.

14日(木) 午前. Performance Evaluation in Optimization and Non-linear Equations

W.Murray & P.E.Gill, Performance evaluation for optimization software.

K.Schittkowski, Effective development of optimization codes (N.Molseeyev 不参のため変更)

J.J.More, Development and testing quality optimization software .

午後 Open Sessions: Panel 1. The use of mathematical software outside the mathematical community.

Panel 2. Nonlinear programming software .

夜 New Floating Standard system; by A.Bossanit, R.P.Brend & W.Kahan (Informal Session)

15日(金) General Aspects of Performance Evaluation III.

J.R.Rice, Methodology for the algorithm selection problem

J.A.Nelder, Experimental design and statistical evaluation

B.Ford,G.S.Hodgson & D.K.Sayers, Evaluation of numerical software intended for many machines — Is it possible?

午前のみ；続いて閉会式．午後は役員会，前夜の討論の続き，及びWien市観光（希望者のみ，有料）など．

3. いくつかの感想（メモより）

1°（第1日）数値計算プログラムの正当性とは何か？

(i) 算法の正当性の検証は，やればできそうである．実際，Choleski分解や平方根（Newton法）の算法の厳格な形式的体系上の証明が例示された．しかし実用上では，「算法」と「プログラム」とは，正しく区別しなければならない．

(ii) 与えられた算法を正しく実現した（はずの）プログラムが，現実の計算機でどういう結果を与えるかは，「予測不可能」と思ったほうがよい近似であるらしい．計算機の四則演算体系を十分注意して設計すれば，多少の望みがあるらしい．

2°プログラムの性能とは何か？

(i) とかく速度が重視されるが，少くとも結果の信頼性，たくましさ（robustness）といった尺度を数量化し，総合判断をする必要がある．完全主義者には不満だろうが，実用上の成功率といった尺度も，正しく与えられれば有用である．

(ii) 性能比較のための客観的な国際的標準問題を確立する必要があるらしい．不適切な例による比較しかしていない論文は拒否すべきであるようだ．

3°（第2日）線型計算について．

一口に疎行列（sparse matrix；中国では稀陣）といっても，完全な帯行列，ブロック三重対角型，一面に非零要素

がちらばめられた行列，…… などでは性格がかなり異なる．何らかの（必要なら人間の協力による）非零要素のパターン認識が考えられなければならない．

4°（第2日午後）ソフトウェアの品質保証について．

この問題は，これまであまり日本では論じられなかったが，ソフトウェアが有料化されるにつれて，当然生ずる問題である，なんらかの法的規制が必要になるかもしれない．ただ通例の欠陥商品とかなりの違いがあり，今回も現状報告や問題提起に留った感がある．

5°（第3日）微分方程式について

(i) $y' = f(x, y)$ に対し，$F(x, y, y) = f(x, y)$ となる $F(x, y, z)$ をとり，たとえば次の形のような混合型差分：

$$(y_{n+1} - y_n)/h = F(x_n, y_n, y_{n+1})$$

による方法が詳しく論ぜられた．

(ii) 常微分方程式の解を普通の函数と考え，特定の値 x_0 での真値と計算値の差を誤差と考える，というのは一面的に過ぎるらしい．解は正しくは曲線であり，真の解と近似解との距離や，ある値に達する x の値の差，といった「誤差」も必要であるらしい．

6°（第4日） 最適化について

(i) 依然として難問であり，決定的な算法がない．また試

験用問題の次元数(変数の数)が低すぎる傾向がある.

(ii) 残差が小さいのがよい解なのか, 真の値に近いのがよい解なのかという根本的な疑問(山下真一郎氏が30年前から繰り返えし論じている)が未だに論争の種である.

7°(第5日) しめくくり.

既製の数学とは別の立場の対象の分類が必要らしい. たとえばRiceは今回は扱われなかった偏微分方程式について, 8種への分類試案を提案した.

8°(楽屋裏の雑談)

(i) 計算機科学者の「IBM離れ」は, 世界的な現象である.

(ii) 西ヨーロッパの人々も, 外国(アメリカ)の計算機導入に対する無形の圧力(?)をぼやいていた.

(iii)「掲示は誰も読まない」と考えたほうがよい近似であるらしい. 大事なことは少くとも2回アナウンスすること.

上記のうち「数理科学」以外の, たとえば計算機政策に関する話題が多いことをお許し願いたい. パネル討論と木曜夜の非公式部会(実はこれが最も重要だった)については別に述べる. 依然として, 数学者好みのきれいな理論と, 現実の計算及びその道具との間のギャップを感ずるし, また日本での計算機科学者の孤立ないし周辺のアンバランスを痛感させられた.

4. パネル討論．　2つ行われたが，その2は事実上午前の続きであった．その1は当初「各国の現状」という話であったが前記の題になった．メンバーは次の通り（アルファベット順）

B. Einarson（スエーデン），A.M. Erisman（米；座長），S. Hitotumatu（日），L.F. Shampine（米），J.H. Wilkinson（英），N.N. Yanenko（ソ連）

大体この順に発言したが，Yanenko は所用のため，自分の話を最初にして退場し，ソ連の現状をきくことができなかった．実質的に Einarson が，「優れた数学的ソフトウェアがなぜ使われないか？」を論じたのが基調講演になった．彼は使用者の無知・無理解，更新の不備（マニュアルとの不一致），互換性など6の原因をあげた．私も東大大型機センターの利用統計を引用して，教育の重要性を説いたつもりである．（本巻の二宮市三，唐木幸比古両氏の論文参照）．そのあとかなり教育論議が行われた．しかし大勢としては，「一般使用者は，数値解析の論文など読みはしないし，だからこそ良い算法に基く数学的ソフトウェアが必要なのだ．」というバカチョン方式(?)が多数をしめた感じである．

この部分は各パネリストの原稿を Erisman がまとめ，正式の報告の前にプレプリントの形で限定配布されている．

II. 新標準浮動小数点体系

1. 現在の大半(国産の場合はほとんど)の計算機はIBM 360の流れをくむ16進バイト方式を採用している。数値計算の目的にはこれは甚だ始末が悪い。歴史的には(木村泉氏が以前に報告していたが)360(SPREAD)計画の折に、技術計算に対する配慮が不十分であり、「必要なら倍長演算がある」で反対を強引に押し切ったしわよせといえそうである。

集積回路の発展その他により、かつての「規模の経済」は神話となり、特殊目的にはそれに向いた単能計算機のほうが有利になってきた現在、数値計算向きの専用計算機の再設計を考えるべき時期に達したようである。計算量の理論の専門家達の間にも、単に能率のよい算法を提案するだけでなく、その算法を有効に活用できる計算機の設計(算法の金物化)まで考えなければ無意味という意見が強い。

そういう案は、平野菅保氏などが十年以上前から繰り返し主張しておられたが、今回Kahanらの提案をきいて、びっくりした。—提案の内容自体にではなく、新標準を提案し、試作機を作っている点にである。もちろんその案にもいろいろ批判や注文があろうが、ともかく注目すべき提案なので、その概要を紹介したい。

2. 基本構造

s	e	f
0	E	E+F

純2進，絶対値，"げたばき・けち" 表現．

すなわち上の図のデータ構造で，eとfの間に小数点をおき

$$X = (-1)^{s} \cdot 2^{e-B} \times (1.f) \qquad (1 \leqq e < 2B+1)$$

	E	B	F	E+F+1(全長)
単長型	8	127	23	32
単長拡大型	11	1023	≧32	≧44
倍長型	11	1023	52	64
倍長拡大型	15	16383	≧64	≧80
4倍長型	15	16383	112	128

注．1° 上記すべての型を用意するわけではない．拡大型はアキュムレータなどで保持する型であり，主記憶中は32, 64, 128 ビットの標準型になる．"げたばき"とは，実際の指数部が e−B と bias されていること；"けち"とは正規化数の頭の1を陽に表示しないことを意味する．"けち"表現は2進法特有の方法であり，この点でも純2進方式は優れている．

2° 補数表示も検討されたが，さしあたっては絶対値表示となった．指数部を上位にしたのは，あふれの影響をさけるためと，大小比較がビットの辞書式順序で可能なためらしい．

3° e＝0 は不正規化（de-normalized）数を表す．e＝0, f＝0 は ±0 を表す．これにより，僅かの下への

あふれは救える．―― この考えは古くからあり，近年は「労多くして功少し」として捨てられている．これが一つの問題点であるが，Kahan によれば「功少いかもしれないが労もそう多くない」として採用したようである．

4° $e = 2B+1$ は特別扱いをする．このとき $f = 0$ を $\pm\infty$ とし，$f \neq 0$ は非数（Not A Number；略して NAN）とする．この f は多くの場合，それを生じた番地のポインタなどの情報である．

四則演算も細かく規定されている．$1/(\pm 0)$ は $\pm\infty$ になり，$(+\infty)+(-\infty)$ は不定（NAN）になる．ただし $0 \times \mathrm{NAN}$ を NAN のままにするのか，0 にするのかは未定らしい．

上下のあふれはある程度拡大型（指数部も広い）で防がれる．たとえば $\sqrt{x^2+y^2}$ で，x, y の指数部が範囲の半分をこえていても，最終結果が範囲内ならこれで救える：

[哲学？] 金物の不備を，$|x| \geq |y|$ なら $|x|\sqrt{1+(y/x)^2}$，$|x| < |y|$ なら $|y|\sqrt{1+(x/y)^2}$ などとソフトウェアで苦労して救うような余計な苦労はやめて，なるべく普通の数式をそのまま計算できる金物を作ろう．少くとも苦労は金物，ソフトウェア，使用者の三者で等分に分ちあおう！

――倍長数の指数部を単長数のより広くしてあるのが，大変に喜ばしい．

3. その他の特長

1° 丸めは，区間解析の便をはかって４種用意する．

RN	Round to nearest	１捨０入 *)
RZ	Round to zero	絶対値切捨て
RP	Round to Plus ∞	正なら切上げ，負なら切捨て
RM	Round to Minus ∞	正なら切捨て，負なら切上げ **)

*) 中央のときは末位が０になるように丸める．現在のJIS規格の４捨５入はそうなっている．この方式の利点が例示されている．

**) ０はRMでは－０，他では＋０になる．

なお±０，±∞の区別をしない"射影モード"も用意する．

2° 全体的に次の哲学が流れている：

この世に完全なものはない；すべて費用とのバランスである．その範囲で，これまでのものより信頼性が高く，柔軟性に富み，ソフトウェアの作りやすいものをねらう．他方既存の（現行の）機械用のプログラムとの互換性を重視する．

じっさい純粋に数値計算専用なら，CRAY-1のように１語64ビット方式が優れている．単長32ビットとし，少々無理をしてまで単長，その拡大型を考えたのも，互換性のためであろう．また実用上さしつかえない範囲で標準を確立することをねらっている．

3° $s=\sum a_i b_i$ の計算で，$a_i \cdot b_i$ を中間ですべて倍長（少くとも単長拡大型）で保持し，最後の答 S を単長に丸めれば，精度が上ることは，Wilkinson の理論でも，また多くの実験でも明白であるが，現在の多くの計算機では，この実行に特別なプログラムを要し，いちじるしく速度を低下させている．Kahan の論文の後半には，この種のよく現れる計算の「部品」の精度を保持するための工夫と実例が数多く載っている．

4° Kahan らは，「あふれ」をめったに生じない「異常現象」とは考えず，数値計算ではしょっちゅう生ずる現象であり，少し位は何とか自動的に救おうと考えている．もちろん救えなくなれば警告を発する．（この限界については検討事項になっている．）

5 検討事項．既に述べたもの（補数表示，射影モード，0×NAN の結果，など）を含めて 9 項目があげられている．そのうち二三を引用する．

(i) $a>0$ のとき，$\sqrt{-a}$ を $-\sqrt{a}$ とするのが原案だが，NAN とすべきではないか？

(ii) 現在の案では不正規化数による乗除は NAN としているが，可能な場合（不正規化数に普通の大きさの数を掛けたり，小さい数を不正規化数で割ったりする）には，許容すべ

きではないか？

(iii) 倍長整数を倍長整数で割って商と剰余を正しく求める命令までが本当に必要か？

日本に帰ってから，二三の方々に伺ったところ，いろいろの御注意をいただいた．たとえば

(iv) 10進機械は？ ―― 1000進にすれば，ビットの損は少い．ただ"けち"表現はできない―それも必要があろう．

(v) 単長（32ビット）が必要か？ ―― 互換性を無視すればたぶん不要だろう。もっとも単長とその拡大型のみという「簡易版」も目的によっては有用かもしれない．

きく所によれば，DEC社とINTEL社で，この体系を実現する試作機を製作中という．Kahanからは，「さっそく日本のメーカーがまねをするだろう」という厳しい皮肉をちょうだいしてきた！

[参] J. T. Coonen, Specifications for a proposed standard for floating point arithmetic, 1978 Dec. 6. Revised Memorandum No. UCB/ERL M 78/72.