浮動小数点演算の単位丸め誤差のテスト*

牛 島 和 夫**

Abstract

Most scientists and engineers using computers through higher languages such as FORTRAN may obtain operating information about machines at their disposal only from language manuals. When they are going to transfer programs written in higher languages from one computer to another of a different manufacturer, such manuals do not give them sufficient information for the purpose of reprogramming. In order to get pieces of information around the unit rounding errors in floating point operations, this paper shows a test program in FORTRAN and discusses a number of methods to analyze results of the program from the users view point.

1. はじめに

計算機を高級プログラム言語のみを用いて利用する大部分の利用者にとって個々の計算機の動作に関する情報は，各メーカの発行するその言語の入門書，あるいは解説書によるのがほとんどである．これらの言語マニュアルは，計算機に関するある程度の情報と，言語の外部仕様に応じた使用法を与えることが主で，言語と特定の計算機が組み合わさった動的な情報については，かなり不十分といってよい．そこで，ユーザとしてはやむをえず実験と称して高級言語を用いて自分の目的に応じたテストプログラムを作って，その高級言語を介した計算機システムに関する動的な情報を得ている．

たとえば，数値計算上重要な役割を演ずる浮動小数点演算の丸め誤差の処理を巡る情報についても，言語マニュアルには精確な記載はほとんどない．処理系Aで作られた数値計算のプログラムを処理系Bにうつしかえる作業を行なう際に，もとのプログラムが知らず知らずのうちに処理系Aの方式のくせを利用したものになっていると，後の処理系のうつしかえの作業は非常に困難になることがある．そこでここでは，特に浮動小数点の丸め誤差の処理を巡って，高級言語のみを介して計算機を利用する利用者の立場から，浮動小数点演算の丸め誤差の処理に関する動的な情報を得るための一つのプログラムを示し，このプログラムをいくつかの処理系にかけた結果を示しながら，そこから情報を得る過程を述べる．

2. 単位丸め誤差の定義

浮動小数点演算の単位丸め誤差は，高級言語を介した処理系ごとに異なる可能性があるが，この誤差をForsythe ら[1] は 1.0 に対する相対誤差として論理式

$$1.0+\mathrm{EPS}=1.0 \qquad (1)$$

を満たすような EPS の最大の値と定義している．まず，2進法の計算機について考察を進める．いま，EPS を，次のような集合に属する数に限定する．

$$\{\pm 2^{-i}\} \quad (i>0) \qquad (2)$$

浮動小数点数の仮数部の長さを t 桁（符号部を除く）とすると(1)式を満足するこの集合の最大値は

0捨1入演算で $\mathrm{EPS}=2^{-t-1}(i=t+1)$ (3)

切捨て演算で $\mathrm{EPS}=2^{-t}(i=t)$ (4)

となることが予期される．これらはそれぞれの方式の単位丸め誤差の 1/2 の大きさである．ところで，FORTRAN でこれらの値を知るプログラムは，直観的に考えれば，論理 IF 文または算術 IF 文を用いて

IF(1.0+2**-I.EQ.1.0)GO TO2 (5)

IF((1.0+2**-I)-1.0)1,2,3 (6)

について，i を大きくしていって最初に文番号2に脱出したときの i の値が(3)または(4)を与えると考えられる．ここで 2^{-i} は便宜的な記述である．しかし処理系ごとに異なる丸め誤差を巡る処理方式のために，必ずしも直観的に期待する結果が得られるとは限らない．したがって，逆に，このようなプログラムを実行させてみれば，その結果から，丸め誤差の処理を巡るそれぞれの処理系に関する動的な情報が得られることが期待される．

* A Test Program for a Unit Rounding Error in Floating Point Operations, by Kazuo USHIJIMA (Faculty of Engineering, Kyushu University)

** 九州大学工学部

3. 丸め誤差に影響する要因

丸め誤差のふるまいに影響する要因を，よりハードウェアに近いものから，よりソフトウェアに近いものに列挙してみる．これらは(5)または(6)の実行にも関連している．

A1. 浮動小数点演算で仮数部をとり扱うレジスタ(以下 Acc と略す)の長さが，必ずしも t 桁(符号部を除く)とは限らない．t 桁をこえる場合がある．

A2. ベキ乗以外の算術演算子の作用の結果の精度は，実数型になる場合と倍精度実数型になる場合がある．これは A1 の Acc の長さと密接な関連がある．言語マニュアルの関連部分を抜き出すと**表1**のようになる．ここで，実数型を R，倍精度実数型を D とする．

表 1　算術演算子の作用の結果

(a)

第1演算数 ＼ 第2演算数	+−*/	R	D
+−*/	R	R	D
R	R	R	D
D	D	D	D

(b)

第1演算数 ＼ 第2演算数	+−*/	R	D
+−*/	R	R	D
R	R	D	D
D	D	D	D

JIS FORTRAN では，表1(a)を規定としている(JIS[2] p. 9)．

A3. 数値の表示方式は，絶対値表示と補数表示に大別され，負数の場合と正数の場合では，丸められた値が代表する区間がそれぞれの方式によって異なる(表2参照，後述)．

A4. 丸め誤差の処理には，4捨5入(2進法では0捨1入)と切り捨ての2方式があり，4捨5入の採否はコンパイラ作成者に任される場合がある．

A5. 組込み関数のうち，丸め誤差の処理に積極的に関与しうる単精度化関数SNGL，および倍精度化関数 DBLE の動作について言語マニュアルでは，単精度化する，倍精度化するとしか説明していないが，Acc 上で単精度化または倍精度化が行なわれるので，これらの操作がまず浮動小数点数の仮数部に対して行なわれ，特に SNGL の場合，処理の対象となる桁は，A1 との関連から単精度仮数部の長さと必ずしも一致せず，そのうえ処理系が違えば，その桁で4捨5入が行なわれたり切り捨てが行なわれたりする．

A6. A レジスタが複数個ある場合には，演算の中間結果をAレジスタにおきざりにできるので，中間結果を格納する際に生ずる丸め誤差の発生機構に影響がありうる．

A7. 最適化の程度によるもの．コンパイル時の定数の処理，共通式のくくり出し，たとえば(5)式の文の関係式の両辺から定数 1.0 がコンパイル時に消去されてしまっては，実行時に(5)の文は絶対に true になりえない．

A8. 定数の表現の一意性の問題．ソースプログラムに直接記述した定数，データとして読み込んだ定

```
      SUBROUTINE IFCHEK(X,Y,IBEKI)
      COMMON BASIS(100),LINE
      DOUBLE PRECISION DEPS
      DIMENSION   IND(20)
      A=X
      B=Y
      IB=IABS(IBEKI)
      EPS=SIGN(BASIS(IB),FLOAT(IBEKI))
      DEPS=DBLE(EPS)
      DO 1 I=1,20
    1 IND(I)=1
      IF(A) 20,2,20
    2 CONTINUE
      IF(1.0+EPS.EQ.1.0)          IND(1)=0
      IF(EPS+1.0.EQ.1.0)          IND(2)=0
      IF(1.0.EQ.1.0+EPS)          IND(3)=0
      IF(1.0.EQ.EPS+1.0)          IND(4)=0
      IF(SNGL(1.0+DEPS).EQ.1.0)   IND(5)=0
      IF(1.0.EQ.SNGL(1.0+DEPS))   IND(6)=0
      IF(1.0+EPS.EQ.1.0+EPS)      IND(7)=0
      INDEX=8
 1008 IF(1.0+EPS-1.0)             12,13,14
 1009 IF(EPS+1.0-1.0)             12,13,14
 1010 IF(1.0-(1.0+EPS))           12,13,14
 1011 IF(1.0-(EPS+1.0))           12,13,14
 1012 IF(SNGL(1.0+DEPS)-1.0)      12,13,14
 1013 IF(1.0-SNGL(1.0+DEPS))      12,13,14
 1014 IF(1.0+EPS-(1.0+EPS))       12,13,14
   12 IND(INDEX)=-1
      GO TO 15
   13 IND(INDEX)=0
      GO TO 15
   14 IND(INDEX)=1
   15 INDEX=INDEX+1
      KEY=INDEX-8
      GO TO (1009,1010,1011,1012,1013,1014,50),KEY
   20 CONTINUE
      IF(A+EPS.EQ.B)              IND(1)=0
      IF(EPS+A.EQ.B)              IND(2)=0
      IF(B.EQ.A+EPS)              IND(3)=0
      IF(B.EQ.EPS+A)              IND(4)=0
      IF(SNGL(A+DEPS).EQ.B)       IND(5)=0
      IF(B.EQ.SNGL(A+DEPS))       IND(6)=0
      IF(A+EPS.EQ.B+EPS)          IND(7)=0
      INDEX=8
 2008 IF(A+EPS-B)                 22,23,24
 2009 IF(EPS+A-B)                 22,23,24
 2010 IF(B-(A+EPS))               22,23,24
 2011 IF(B-(EPS+A))               22,23,24
 2012 IF(SNGL(A+DEPS)-B)          22,23,24
 2013 IF(B-SNGL(A+DEPS))          22,23,24
 2014 IF(A+EPS-(B+EPS))           22,23,24
   22 IND(INDEX)=-1
      GO TO 25
   23 IND(INDEX)=0
      GO TO 25
   24 IND(INDEX)=1
   25 INDEX=INDEX+1
      KEY=INDEX-8
      GO TO (2009,2010,2011,2012,2013,2014,50), KEY
C
C*****OUTPUT*****
C
   50 CONTINUE
      IS=ISIGN(1,IBEKI)
      JB=IB
      IF(A) 51,52,51
   52 WRITE(6,100)IS,JB,(IND(I),I=1,14)
      RETURN
   51 WRITE(6,101)A,IS,JB,B,(IND(I),I=1,14)
      RETURN
  100 FORMAT(10X  ,I4,5H*2**(,I3,3H)  ,5X,  2X,14I3)
  101 FORMAT(F10.1,I4,5H*2**(,I3,3H) =,F5.1,2X,14I3)
      END
```

図 1

表 2　±1.0 近傍の丸め誤差のふるまい

方式	代表値（格納された値）	絶対値表示 代表される区間	区間の幅	単位丸め誤差	補数表示 代表される区間	区間の幅	単位丸め誤差
0捨1入	$-1-\Delta^0$	$(-1-\Delta^0-\Delta^{-1},\ -1-\Delta^{-1}]$	Δ^0	Δ^{-1}	$[-1-\Delta^0-\Delta^{-1},\ -1-\Delta^{-1})$	Δ^0	Δ^{-1}
	-1	$(-1-\Delta^{-1},\ -1+\Delta^{-2}]$	$\Delta^{-1}+\Delta^{-2}$	Δ^{-1}	$[-1-\Delta^{-1},\ -1+\Delta^{-2})$	$\Delta^{-1}+\Delta^{-2}$	Δ^{-1}
	$-1+\Delta^{-1}$	$(-1+\Delta^{-2},\ -1+\Delta^{-1}+\Delta^{-2}]$	Δ^{-1}	Δ^{-2}	$[-1+\Delta^{-2},\ -1+\Delta^{-1}+\Delta^{-2})$	Δ^{-1}	Δ^{-2}
	$-1+\Delta^0$	$(-1+\Delta^{-1}+\Delta^{-2},\ -1+\Delta^0+\Delta^{-2}]$	Δ^{-1}	Δ^{-2}	$[-1+\Delta^{-1}+\Delta^{-2},\ -1+\Delta^0+\Delta^{-2})$	Δ^{-1}	Δ^{-2}
	$1-\Delta^0$	$[1-\Delta^0-\Delta^{-2},\ 1-\Delta^{-1}-\Delta^{-2})$	Δ^{-1}	Δ^{-2}			
	$1-\Delta^{-1}$	$[1-\Delta^{-1}-\Delta^{-2},\ 1-\Delta^{-2})$	Δ^{-1}	Δ^{-2}	左に同じ		
	1	$[1-\Delta^{-2},\ 1+\Delta^{-1})$	$\Delta^{-1}+\Delta^{-2}$	Δ^{-1}			
	$1+\Delta^0$	$[1+\Delta^{-1},\ 1+\Delta^0+\Delta^{-1})$	Δ^0	Δ^{-1}			
切捨て	$-1-\Delta^0$	$(-1-2\Delta^0,\ -1-\Delta^0]$	Δ^0	Δ^0	$[-1-\Delta^0,\ 1)$	Δ^0	Δ^0
	-1	$(-1-\Delta^0,\ -1]$	Δ^0	Δ^0	$[-1,\ -1+\Delta^{-1})$	Δ^0	Δ^0
	$-1+\Delta^{-1}$	$(-1,\ -1+\Delta^{-1}]$	Δ^{-1}	Δ^{-1}	$[-1+\Delta^{-1},\ -1+\Delta^0)$	Δ^{-1}	Δ^{-1}
	$-1+\Delta^0$	$(-1+\Delta^{-1},\ -1+\Delta^0]$	Δ^{-1}	Δ^{-1}	$[-1+\Delta^0,\ -1+\Delta^0+\Delta^{-1})$	Δ^{-1}	Δ^{-1}
	$1-\Delta^0$	$[1-\Delta^0,\ 1-\Delta^{-1})$	Δ^{-1}	Δ^{-1}			
	$1-\Delta^{-1}$	$[1-\Delta^{-1},\ 1)$	Δ^{-1}	Δ^{-1}	左に同じ		
	1	$[1,\ 1+\Delta^0)$	Δ^0	Δ^0			
	$1+\Delta^0$	$[1+\Delta^0,\ 1+2\Delta^0)$	Δ^0	Δ^0			

(または) は区間の端の点を含まないことを意味する.
[または] は区間の端の点を含むことを意味する.
$\Delta^0=2^{-t+1},\ \Delta^{-1}=2^{-t},\ \Delta^{-2}=2^{-t+1}$

数，10進2進変換における変換法の違いが原因で，同じ値になるはずの 10 進数が，内部表現で必ずしも同じにならない場合がある.

このほか，(5) だけに関連する問題として

A9. 関係式のコンパイル方法の多様性. たとえば，(a) 算術式に直して，条件付飛越し命令で処理する. (b) 関係式の両辺を別々に評価して比較命令で処理する. この場合，値が別々に評価されるために，中間結果の退避が行なわれ丸めが発生する.

4. 論理 IF と 算術 IF の実行

前節に述べた丸め誤差を巡る情報を得るために(5)と(6)，およびそれらのいくつかの書き替えを作ってその動作を調べてみた. プログラムの一部を**図 1** に示す. ここで A7 定数の消去の効果が及ばないように

IF(A+EPS. EQ. B)　　(7)

についても加えた. ただし，AとBには同じ値を与え EPS は (2) の集合に属する値とする. すなわち，次の3つの場合について実行させることになる.

Case Ⅰ　　1.0+EPS=1.0

Case Ⅱ　　A+EPS=B(A=B=1.0)

Case Ⅲ　　A+EPS=B(A=B=−1.0)

に対して

$EPS=\pm 2^{-i}(i=t-1, t, t+1\cdots\cdots)$

とする.

また逆に A7 共通式のくくり出しの有無を知るために，数学的には両辺が同じ値をとるべき関係式

IF(1.0+EPS. EQ. 1.0+EPS)
IF(A+EPS. EQ. B+EPS)　　(8)

が参考のために加えてある.

さて，Case Ⅱ と Case Ⅲ の結果を比較すれば，A3 に関する情報が得られる. すなわち，絶対値表示の場合は，数直線に対して丸めは0に対して対称性をもつのに対して，補数表示の場合は，同一方向性をもっているので，負の場合 (Case Ⅲ) に，それぞれの代表値によって代表される区間が，それぞれの表示方式で差が出てくる. 実行の結果を検討する際の便利のために，±1.0 の近傍の丸め誤差のふるまいを**表 2** に示した. ここで

$$\left.\begin{aligned}\Delta^0&=2^{-t+1}\\ \Delta^{-1}&=2^{-t}\\ \Delta^{-2}&=2^{-t-1}\end{aligned}\right\}\qquad(9)$$

表 3　Acc の長さを考慮した算術演算子の作用の結果

(a)

第1演算数 ＼ 第2演算数	+−*/	R	R′	D	D′
+−*/	R	R′	R′	D′	D′
R	R′	R′	R′	D′	D′
R′	R′	R′	R′	D′	D′
D	D′	D′	D′	D′	D′
D′	D′	D′	D′	D′	D′

(b)

第1演算数 ＼ 第2演算数	+−*/	R	D	D′
+−*/	R	D′	D′	D′
D	D′	D′	D′	D′
D′	D′	D′	D′	D′

表 4　単精度の場合の結果

処理系	Case 番号	EPS	論理IF 1	2	3	4	5	6	7	算術IF 8	9	10	11	12	13	14
H	I・II	1*2** (−22)=⊿⁰	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−23)	1	1	1	1	0	0	0	1	1	−1	−1	0	0	0
		1*2** (−24)	1	1	1	1	0	0	0	1	1	−1	−1	0	0	0
		1*2** (−25)	1	1	1	1	0	0	0	1	1	−1	−1	0	0	0
		1*2** (−26)	1	1	1	1	0	0	0	1	1	−1	−1	0	0	0
		−1*2** (−22)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−23)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−24)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−25)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−26)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
	III	1*2** (−22)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−23)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−24)	1	1	1	1	0	0	0	1	1	−1	−1	0	0	0
		1*2** (−25)	1	1	1	1	0	0	0	1	1	−1	−1	0	0	0
		1*2** (−26)	1	1	1	1	0	0	0	1	1	−1	−1	0	0	0
		−1*2** (−22)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−23)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−24)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−25)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−26)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
N	I・II	1*2** (−34)=⊿⁰	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−35)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−36)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−37)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−38)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−34)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−35)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−36)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−37)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−38)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
	III	1*2** (−34)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−35)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−36)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−37)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−38)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−34)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−35)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−36)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−37)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−38)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
C D	I・II	1*2** (−25)=⊿⁰	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−26)	1	1	1	1	1	1	1	1	1	−1	−1	1	−1	−1
		1*2** (−27)	1	1	0	0	1	0	1	1	1	0	0	1	0	1
		1*2** (−28)	1	1	0	0	1	0	1	1	1	0	0	1	0	1
		1*2** (−29)	1	1	0	0	1	0	1	1	1	0	0	1	0	1
		−1*2** (−25)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−26)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−27)	1	1	0	0	1	0	1	−1	−1	0	0	−1	0	−1
		−1*2** (−28)	1	1	0	0	1	0	1	−1	−1	0	0	−1	0	−1
		−1*2** (−29)	1	1	0	0	1	0	1	−1	−1	0	0	−1	0	−1
C D	III	1*2** (−25)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−26)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−27)	1	1	1	1	1	1	1	1	1	−1	−1	1	−1	−1
		1*2** (−28)	1	1	0	0	1	0	1	1	1	0	0	1	0	1
		1*2** (−29)	1	1	0	0	1	0	1	1	1	0	0	1	0	1
		−1*2** (−25)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−26)	1	1	0	0	1	0	1	−1	−1	0	0	−1	0	−1
		−1*2** (−27)	1	1	0	0	1	0	1	−1	−1	0	0	−1	0	−1
		−1*2** (−28)	1	1	0	0	1	0	1	−1	−1	0	0	−1	0	−1
		−1*2** (−29)	1	1	0	0	1	0	1	−1	−1	0	0	−1	0	−1
D	I	1*2** (−25)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−26)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−27)	1	1	0	0	1	0	0	1	1	0	0	1	0	0
		1*2** (−28)	1	1	0	0	1	0	0	1	1	0	0	1	0	0
		1*2** (−29)	1	1	0	0	1	0	0	1	1	0	0	1	0	0
		−1*2** (−25)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−26)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−27)	1	1	0	0	1	0	0	−1	−1	0	0	−1	0	0
		−1*2** (−28)	1	1	0	0	1	0	0	−1	−1	0	0	−1	0	0
		−1*2** (−29)	1	1	0	0	1	0	0	−1	−1	0	0	−1	0	0

で表わした．ところで，表2に示した代表値の意味は，一般に Acc 上の値が丸められて記憶装置に格納された値である．しかし，A1 から Acc の長さが仮数部の長さに等しく t 桁の場合(Rと略す)，t 桁をこえる場合 (R′ と略す) とでは，演算のふるまいが異なる．すなわち

Rの場合：演算の結果がすでに丸められて，Acc 上で代表値のいずれかの値をとっている．したがって，これを記憶装置に格納しても Acc 上と同じ値である．

R′ の場合：演算の結果，代表される区間内の値が R′ の表現しうる桁だけ Acc 上に残る．Acc 上の値に指数部をつけて記憶装置に格納する際に丸めが行なわれて，表2の代表値のいずれかをとる．したがって R′ の場合は，表1は厳密にいえば**表 3** のように書き替えられる．ここで D′ は倍精度の場合 Acc がその仮数部より長いことを意味する．

図1のプログラムには，さらに A5 SNGL の動作を調べる文

```
IF(SNGL(1.0+DEPS).EQ.1.0)          (10)
```

などを加えた．ここで

```
DBPS=DBLE(EPS)                     (11)
```

であえ．

終わりに A8 の定数の一意性の問題は，このようなテストに影響を与えるところが大であり，それだけで別のテストが構成できる[3]．そこで図1のテストプログラムでは，1.0 または −1.0 はすべてこの記法で，ソースプログラムに直接記述したものだけに限る．また（2）の集合に属するデータは，IBASE=2 として

```
BASE=IBASE
BASIS(1)=1.0/BASE
DO 1 I=1,99
1 BASIS(I+1)=BASIS(I)/BASE         (12)
```

で作る．これによって，2進法の計算機では

$$BASIS(I)=2^{-I}$$

は，仮数部の最上桁だけが1で，他は0のデータとして t 桁で正確に表わすことができる．

5. 結果の解釈

図1のプログラムを **C, D, H, N** の処理系で実行した結果を**表 4** に示す．1～7の論理 IF に対して論理式の値が true のとき 0, false のとき1が出力されている．また 8～14 の算術 IF（これらは文番号 1008～1014 または 2008～2014 に対応している）に対して算術式の値が負，0，正に応じて −1, 0, 1 が出力されている．表4の結果の解釈によって得られる FORTRAN を仲介とした各計算機システムの丸め誤差を巡る情報のおもなものを**表 5** にまとめた．各処理系についてこのような解釈をくだした過程の概略を以下に述べる．

[C および D] Acc に余裕があって R′ であるとすれば 1, 2(8, 9) の演算が 1.0+EPS−1.0 の順序で行なわれ，中間結果を退避することがないため，EPS の値が Acc 上に残って true または0になりえない．一方，3, 4 (10, 11) は (1.0+EPS) がいったん中間結果としてさきに格納され，そのとき0捨1入が生ずるとして解釈できる．すなわち表2から，$1+\Delta^{-1}\Rightarrow 1+\Delta^{0}$，$1-\Delta^{-1}\Rightarrow 1-\Delta^{-1}$，また $|EPS|\leq\Delta^{-2}$ について，1+EPS⇒1 となる．また Case Ⅲ では，0捨1入と補数表示を仮定すれば，$-1+\Delta^{-2}\Rightarrow -1+\Delta^{-1}$ $-1+\Delta^{-3}\Rightarrow -1$，$-1-\Delta^{-1}\Rightarrow -1$，$-1-\Delta^{-2}\Rightarrow -1$ によって説明できる．5 (12) は，1, 2 (8, 9) と結果が同一であることから SNGL の動作が R′ に対して行なわれることが推定できる．すなわち，SNGL された値は Acc 上に残り 1.0 を減じても0にならない．それに反し 6 (13) は単精度化された値がいったん退避されるため 3, 4 (10, 11) の場合と同一になる．7

表 5 結果から得られる情報

処理系 / 項目	H	N	C	D
1 語 の 長 さ*	32	48	36	36
単精度の仮数部* の長さ(t 桁)	23	35	26	26
A1 Acc の長さ	R′ (t 桁をこえるという意味)	R	R′	R′
A2*	R·R=D	R·R=R	R·R=R	R·R=R
A3	補 数 表 示	同 左	同 左	同 左
A4 丸めの方式	単精度については不明	切 り 捨 て	0 捨 1 入	同 左
A5 単精度化関数 (SNGL)	Dを切り捨ててRにする	同 左	Dを切り捨ててR′にする	同 左
A6 Aレジスタの個数	複 数 個*	マニュアルに記述なし	1個 (中間結果の退避方法から)	同 左
A7 共通式のくくり出し	R·R=Dのため不明	切捨て演算のため不明	な し	あ り

* 事前に言語マニュアルから得られる情報．

(14) は，Case Ⅰ，Ⅱ の比較から，**D** では共通式 1.0 +EPS がくくり出されていると解釈できる．また特に，**D** Case Ⅱ EPS=Δ^{-1} の場合は，Acc 上の $1+\Delta^{-1}$ が退避されて $1+\Delta^{0}$ に変化するため

$$1+\Delta^{-1}-(1+\Delta^{0})<0$$

となると解釈できる．

[N] Acc が R であり，そこで切捨て演算が行なわれるとすると Case Ⅰ，Ⅱ，1, 2, 3, 4 (8, 9, 10, 11) EPS >0 の結果は説明できる．すなわち $0<\mathrm{EPS}\leq\Delta^{-1}$ については，1+EPS⇒1 となって，Acc 上には EPS の情報は残らない．また $-\Delta^{-1}\leq\mathrm{EPS}<0$ については $1+\mathrm{EPS}\Rightarrow 1-\Delta^{-1}$ となって指数部が 1.0 とは異なってしまうため結果はすべて非 0 になる．Case Ⅲ の結果は補数表示と切り捨てとすれば解釈できる．たとえば，$-1-\Delta^{-2}\Rightarrow -1-\Delta^{0}$ など．5 (12) は，1, 2 (8, 9)，6 (13) は，3, 4 (10, 11) に同じであることから SNGL は t 桁に対して切り捨てが行なわれる．7(14) は，切り捨てられ格納された 1.0+EPS と Acc 上の 1.0+EPS が，Acc が R であるために同じ値になり，A7 共通式のくくり出しの有無は結論できない．

[H] C, D, N とちがって R·R=D のために 1, 2, 3, 4 (8, 9, 10, 11) の演算の結果は倍精度型になってしまうので，t を単精度仮数部の長さにとっている限り 0 にはなりえない．したがって，単精度の丸め誤差に関する情報は得られない．SNGL は単精度化の際に切り捨てを行なっている．もし 0 捨 1 入であれば Case Ⅰ，Ⅱ 5, 6 (12, 13) EPS=Δ^{-1} の結果が非 0 にならなければならない．また補数表示は，5, 6(12, 13) の Case Ⅰ と Ⅲ の比較からいえる．

6. 倍精度演算の場合

表 4 は単精度に関する結果だけであったが，とくに **H** については，R·R=D のために結論を保留しなければならない項目がいくつかある．このテストを倍精度について同じように行なえば，D·D=D (D·D=Q とはならないところに注目．ここで Q はたとえば 4 倍精度実数とする) は，ちょうど他の処理系の R·R=R に対応するから，他の処理系で得られたような情報が得られる可能性がある．また R·R=R の処理系についても D·D=D の場合には，倍精度の Acc は，単精度の場合とちがって，もうすでに長さに余裕がないという場合もありうるので，丸め誤差の取扱い方法に，単精度の場合と異なったふるまいをする可能性がある．そこで同じ計算を倍精度についても実行させる．ただし単精度の場合の 5, 6 (12, 13) に対応する文は倍精度では意味がないので，R·D=D を利用して

$$\mathrm{IF}(1.0+\mathrm{DEPS.\ EQ.\ 1.0}) \qquad (13)$$

などとさしかえる．ここで

DEPS=DBLE(EPS)

である．今度は，倍精度実数の仮数部の長さを t 桁 (ビット) として，3 つの Case について $i=t-1, t, t+1, \cdots\cdots$ について計算した．結果の一部を**表 6** に示す．倍精度の単位丸め誤差の立場から，(9) 式の t を今後は，倍精度仮数部の長さと読みかえると，**H** の結果は，**N** の単精度の場合と同じふるまいをしていることがわかるので表 5 の **N** に関する記述を R を D に書き替えただけでそのまま利用できる．

C および **D** は，$|\mathrm{EPS}|>2^{-69}$ については単精度と同じ結果なので，ここでも Acc は D′ であることが知れる．なお $\mathrm{EPS}\leq 2^{-70}$ については 2 語長が 72 ビットであることから Acc の限界と解釈できる．

N は，EPS<0 について，単精度の場合とは全く様相を異にしているが，表 5 A3, A4 を考慮すれば倍精度の場合は，Acc 最後の桁に回路上の余裕がないことを示唆している．逆にいえば，単精度の場合，Acc の長さを R と結論したが，実際には丸めの判定のための回路の余裕が存在していることが知れる．

7. 追　　補

(1) ***p* 進法への適用**　近年 16 進法の計算機が多数使用されるようになったが

$$p=2^{a}(a=2, 3, 4, \cdots) \qquad (14)$$

の場合は 2 節で設定した (2) 式の集合を

$$\left\{\pm p^{-i},\ \pm\frac{1}{2}p^{-i}\right\} \qquad (15)$$

すなわち

$$\{\pm 2^{-ai},\ \pm 2^{-ai-1}\} \qquad (16)$$

とおきかえる．この集合は，(2) の部分集合であり (14) の場合には，(2) の集合について

$$i=-a(t-1),\ -a(t-1)-1,\ at,\ -at-1, \cdots$$

なる i について図 1 のプログラムをそのまま利用できる．(12) 式の BASIS(・) の値は，$p=16$ の場合は仮数部の最上桁が 1000, 0100, 0010, 0001 のような内部表現になり 16 進 t 桁で完全に表現できる．一方，$p=10$ では (15) は (2) の部分集合とはならないので，IBASE=10 として，(12) 式によりあらためて BASIS(・) を求め，$i=t-1, t, t+1, \cdots$ について，

表 6 倍精度の場合の結果

処理系	Case 番号	EPS	論理IF 1	2	3	4	5	6	7	算術IF 8	9	10	11	12	13	14
H	I・II	1*2** (−54)=⊿⁰	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−55)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−56)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−54)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−55)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−56)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
	III	1*2** (−54)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−55)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−56)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−54)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−55)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−56)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
N	I・II	1*2** (−69)=⊿⁰	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−70)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1*2** (−71)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−69)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−70)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−71)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	III	1*2** (−69)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−70)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−71)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−69)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−70)	1	0	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−71)	0	0	0	1	0	0	0	0	0	0	0	0	0	0
C D	I・II	1*2** (−60)=⊿⁰	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−61)	1	1	1	1	1	1	1	1	1	−1	−1	1	−1	−1
		1*2** (−62)	1	1	0	0	0	0	1	1	1	0	0	0	0	1
		1*2** (−69)	1	1	0	0	0	0	1	1	1	0	0	0	0	1
		1*2** (−70)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−60)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−61)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−62)	1	1	0	0	0	0	1	−1	−1	0	0	0	0	−1
		−1*2** (−69)	1	1	0	0	0	0	1	−1	−1	0	0	0	0	−1
		−1*2** (−70)	1	1	0	0	0	0	1	−1	−1	0	0	0	0	−1
C・D	III	1*2** (−60)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−61)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−62)	1	1	1	1	1	1	1	1	1	−1	−1	1	−1	−1
		1*2** (−63)	1	1	0	0	0	0	1	1	1	0	0	0	0	1
		1*2** (−70)	1	1	0	0	0	0	1	1	1	0	0	0	0	1
		1*2** (−71)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		−1*2** (−60)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−61)	1	1	0	0	0	0	1	−1	−1	0	0	0	0	−1
		−1*2** (−70)	1	1	0	0	0	0	1	−1	−1	0	0	0	0	−1
		−1*2** (−71)	1	1	0	0	0	0	1	−1	−1	0	0	0	0	−1
D	I	1*2** (−60)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−61)	1	1	1	1	1	1	0	1	1	−1	−1	1	−1	0
		1*2** (−62)	1	1	0	0	0	0	0	1	1	0	0	0	0	0
		−1*2** (−60)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−61)	1	1	1	1	1	1	0	−1	−1	1	1	−1	1	0
		−1*2** (−62)	1	1	0	0	0	0	0	−1	−1	0	0	0	0	0

EPS=BASIS(i) および BASIS(i)/2.0 とするなどの変更を要する.

(2) **octal 出力について** たとえば A3 を知るためにはO変換（あるいはB変換）を用いて出力させればただちにわかるので，わざわざ図1のようなプログラムを作る必要はない．しかし，O変換などはすべての処理系が備えているわけではない．また JIS でもこのような機械ごとに異なる欄記述子は規定からはずしてある (JIS[2] p. 25)．さらに O 変換が使えたとしても，出力できる値はいったん Acc から退避された変数の内容であって，Acc 上でのふるまい，Acc の余裕などについては，O変換による出力からでは必ずしも知りえない．Acc の余裕の有無は，収束判定の数値アルゴリズムなどに影響することがあり，プログラムのうつしかえには深刻な問題である．

(3) **目的プログラムの最適化と丸め誤差** 目的プログラムの最適化については，実行速度の短縮，目的プログラムの占有容量の短縮など，主として能率面が前面で論じられやすいが，最適化，特にここでふれた共通式のくくり出しによる丸め誤差の発生点の違いなどについては，それほどきびしく指摘されていない．このプログラムでは，付加的にその影響を調べたにすぎなかったが，最適化の影響だけを調べる目的で別の検査を計画する必要がある．

(4) **組込み関数** 組込み関数は機械語の code が，そのまま目的プログラムの中に組み入れられるだけに，関数によっては，表4の SNGL でみたように機械に依存しやすいものがある．そのために JIS の規

格票でもたとえば，SNGL や DBLE の機能について単精度化する，倍精度化するという簡単な説明しかしていない（JIS[2] p. 32）．たとえば，丸め誤差が関連する複素数に関連した組込み関数などについても，別な検査を計画する必要があろう．

（5） **浮動小数点定数の一意性**　4節で言及したように，このテストは恒川ら[3]によって行なわれたものなどを利用する．

8. むすび

プログラムのうつしかえの作業の観点から浮動小数点丸め誤差を論じた．言語マニュアルにここで論じたような情報をどこまで記載すべきかという議論は，いろいろな立場があって，非常にむずかしい．ある特定の処理系だけを使って仕事をしている利用者にとって，その処理系がもっている特徴(他の処理系との差)を意識することは非常にむずかしく，普通には無意識的にそれらの特徴を使ってプログラムを作っている．知的財産としてのプログラムを他の処理系にうつしかえる必要が生じたときに，もとの処理系の特徴がどう利用されているか，それが明確にされていれば，うつしかえの作業は，プログラムが FORTRAN などの高級言語で書かれている限り非常に容易である　ここにいう特徴とは，丸め誤差の問題ばかりにとどまらない．

最後にこの計算のために，九大，東大，阪大の各大型計算機センターの各計算機（FACOM 230-60 FORTRAN-C および D, HITAC 5020E FORTRAN HARP, NEAC 2200-500 FORTRAN-L）を使用した．本文中に言及した言語マニュアルは上記の各処理系のものである[4),5),6)]．

参考文献

1) G. E. Forsythe & C. B. Moler: Computer Solution of Linear Algebraic Systems, 148 pp., Prentice Hall (1967).
2) 電子計算機プログラム用言語 FORTRAN（水準 7000）JIS C 6201—1967，日本規格協会 62 pp. (1962).
3) 恒川純吉，他：フォートランテストプログラム，第 11 回プログラミングシンポジウム報告集，C80，情報処理学会プログラミングシンポジウム委員会 (1970).
4) FACOM 230-60 FORTRAN 解説編（Ⅱ）．富士通，171 pp. (1968).
5) HITAC 5020, 5020E/F FORTRAN (HARP) 日立製作所，99 pp. (1969),
6) NEAC シリーズ 2200 オペレーティングシステム MOD Ⅲ/MOD Ⅳ FORTRAN コンパイラ説明書，日本電気，218 pp. (1969).

（昭和 45 年 9 月 4 日受付）