数理解析研究所講究録
第172巻 1973年 119-137

誤差評価の可能な多重精度演算について

阪大 工　大中 幸三郎
安井 裕

§1. まえがき

数値計算によって得られた解の確度の評価は，自動計算機による数値計算上，重要な事柄の一つである。高速に処理するための工夫も勿論，興味深い事柄ではあるが，数値解析上の重要な関心は，打切り誤差，丸め誤差，入力データの誤差の伝播の解析的な決定にもある。しかしながら，どのような計算についても，意味のある計算の過程として，その誤差の振舞いを知ろうとするときには，計算過程の個々について何らかの形で直接的に誤差の影響をつかんだ演算を行わなければならない。この目的のためには，丸めの制御，interval arithmetic，significance arithmetic 等の方法があり，たとえば文献(1),(2)等の発表がある。また，単なる多重精度演算プログラムであれば文献(3),(4)にもあり，計算センターのライブラリーになっているものもある。

ここに我々は，通常，新しい数値計算アルゴリズムやライブラリーの開発，またそれらのチェックに際して，しばしば精度，確度の確認，諸係数の算出等のために用いられる多重精度演算プログラムを作成すると共に，さらにこれらを基礎とし，誤差評価が可能となる機能として，丸めの制御，誤差の計算，interval arithmetic 等を行なう一連の基本演算プログラムを作ったので，ここに述べる。このプログラムは，利用に際して全国共同利用大型計算機センター間について，互換性を保つ事を充分考えて，FORTRAN(注)で作られている。それらの特徴と計算所要時間を例とともにここに示す。

(注) DFLOATを用いた以外は，JIS水準7000にふくまれる。

§2．数の丸めについて

現在，計算機のハードウエアにおける実数型の数の丸めはおおむね，切り捨て又は四捨五入である。また，ソフトウエアである言語のコンパイラーによって作られる目的プログラムの丸めの制御は自由になっていない。すでに作られた多重精度演算プログラムにおいても，同様である事が多い。そこで我々は，丸めを切り捨てから切り上げまで制御できる様にプログラムを作った。丸めが制御できる事は，同一のプログラムを切り捨て，切り上げ，四捨五入で実行してみて，その

結果の等しい部分を解と考えるという利用法もある。

我々の行なっている丸めの処理の方法は，簡単なパターン認識によって行なっている。すなわち，最下位から上位に向って"9"がどこまで続くかをさがし，はじめて"9"でない数字がみつかれば，その部分に1を加え，それより下位を零にする方法をとっている。この方法によれば，丸めにともなうcarryを求めて，carryがあればより上位のcarryの計算を行う方法よりも，演算時間が短かくなる。

§3. 演算中における可変精度の有効性について

多重精度演算プログラムで演算中に，operandの仮数部の長さを可変にする事によって，演算時間を短縮する事ができる。この例として，$y=\frac{1}{x}-a$ にNewton法を適用して，逆数を求める場合を考えてみよう。この場合の反復式は次の(1)式で与えられる。

$$x_{n+1}=x_n\times(2-a\times x_n) \tag{1}$$

$$\lim_{n\to\infty} x_n = x=\frac{1}{a} \tag{2}$$

(2)式をもちいて(1)式を変形すれば

$$1-\frac{x_{n+1}}{x}=\left(1-\frac{x_n}{x}\right)^2 \tag{3}$$

したがって，x_nがp桁の精度をもっていれば

$$1-\frac{x_n}{x}\approx 10^{-p}$$

となるので，(3)式をもちいると次の様になる

$$1-\frac{x_{n+1}}{x}\approx 10^{-2p}$$

よって，x_{n+1}の精度は２ｐ桁となることがわかる。すなわちこの場合は，２ｐ桁より多少多い程度の桁数で演算を行なえばよく，同じように考えると，x_{n+2}を求めるには４ｐ桁程度の演算を行なえばよい。なお，初期値は倍長演算で求められ，その有効桁数もある程度わかる。反復の停止則も毎回判定しなくても，始めから定める事ができる。多重精度演算における大小判定は，各要素毎にくらべなければならないので，時間がかかり，大小判定をしなくてもよい事は非常に有効である。

§４．除算ルーチンとその加速法

多重精度演算では，除算がもっとも複雑であり，時間がかかる。ここで用いている方法はdivide and correct methodである。この方法で問題となるのは，各ステップにおいて，いくらが立つかを推定する事である。この推定値を求める方法は，文献(4)では分母，分子の各々最高位の配列要素の内容で推定している。ところがこの推定値は誤差が多い。そこで推定値の信頼性を増すために，実数型の数を利用した。すなわち分母，分子とも高位の三要素の内容を実数化して，有効

数字を長くし，推定値を求めた。この様にして推定値を求めれば，その誤差は高々±1になるので，真値を求めるループも高々1回ですみ，加速される事になる。

§5. 機械語に近いもちい方をする多重精度演算プログラム（M型）

(1) 数は一次元整数配列で表わされる。πを例にとって下図に示す。

	1	2	3	4	5	6	7	8	9
MX	1	0	3	14159	26535	89793	23846	26433	83280

$[MX] = 1 \times 3.14159265358979323846264338328 0 \times (10^5)^0$

MX(1)： 符号 （1:正，0:0，-1:負）
MX(2)： 指数部
MX(3)： 整数部
MX(4)： 小数部
⋮ ⋮
） 仮数部 [一要素内にN桁はいる。ただし $(10^N)^2$ がoverflowしない事]

図 1

(2) 第1 operand を accumulator 的考え方としているけれども，次の(3)との関連や利用上の理由から，サブルーチンの引数としておもてに出した。

(3) 演算中に可変長で演算を行なうのに有利である。

(4) 加減算において，桁落ちを示すindicatorをつけた。

§6 引数として配列名だけでもちいうる多重精度演算プログラム （S型）

(1) 記憶容量を節約するために，小数部のみ一要素内に2N桁入れた。小数部以外をM型と同一にしたのは，M↔S変換を容易にするためである。

(2) operandは使い易さを考えて3つとし，サブルーチンの引数もこの3つのみとした。したがって，サブルーチン内の配列は，使用者が最大寸法を宣言しなければならない。

(3) 演算時間の節約のために，working areaに転送する事をできるだけ少なくした。さらに，operandが次の5つの場合のいずれも可能とした。

X←[Y]㊉[Z], X←[Y]㊉[Y], X←[X]㊉[Y], X←[Y]㊉[X], X←[X]㊉[X]

(4) 乗除算のときは，M型に展開してから演算を行うが，加減算の場合は，展開しなくてもよい場合がある。それは，指数部の差が偶数の時である。この時は，多重精度の加減算は2N桁の数の加減算に分解できる。各々のoperandが2N桁の数であるとき，これら二数の和は一般に2N+1桁になる。しかし，$10^{2N}+1$ はoverflowするので，加算の場合にはそのままではできない。したがって，A+Bは次の様に変形して行なう。

$$C = A + B = (A - 10^{2N}) + B \quad \text{ただし } A, B < 10^{2N}$$

この時の答えは次の通りである。

$\begin{cases} C \geq 0 \text{ のとき：その桁 } C, \text{ 1つ上位へのくり上がり } 1 \\ C < 0 \text{ のとき：その桁 } C+10^{2N}, \text{ 1つ上位へのくり上がり } 0 \end{cases}$

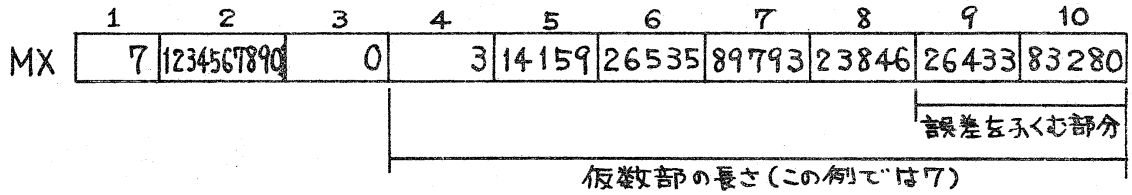
この様に演算を行うと，展開する場合よりも演算時間が短かくなる。

(5) M型の(4)に同じ。

§7. 誤差の計算をともなった多重精度演算プログラム

（E型）

(1) M型に誤差を示す項をつけ加えたもので，配列の各要素の用途は下図の通りである。



$$[MX] = (3.14159265358979323846264338328 0 \pm 0.00000000000000000001234567890) \times (10^5)^0$$

MX(1)： 符号×仮数部の長さ
MX(2)： 誤差　$0 \leq [MX(2)] < 10^{2N}$, 最下位二要素に対応する。
MX(3)： 指数部
MX(4)： 整数部
MX(5)： 小数部　} 仮数部
⋮　⋮

図 2

第一要素に仮数部の長さを示す情報を与えたのは，誤差が伝播して有効桁が短かくなる場合，誤差のないものの乗算で有効桁が長くなる場合に，仮数部の長さが自動的に変化するためである。また，第一要素の内容の定義からわかるように，零に対して誤差は認めない。すなわち，零には絶対的な零しか存在しないとして取扱う。

(2) E型においてもっとも困難なのは，下式におけるδ_Zの計算である。

$$(X \pm \delta_X) \textcircled{OP} (Y \pm \delta_Y) = Z \pm \delta_Z \qquad (4)$$

式(4)において実際の演算に際して誤差をもつから，δ_Zの近似値δ_zを求めることになる。x, y, δ_x, δ_yが$|X|, |Y|, \delta_X, \delta_Y$を倍長精度実数化したものとすれば，

$$\delta_z = \begin{cases} \delta_x + \delta_y & \text{加減算} \\ x\delta_y + y\delta_x + \delta_x\delta_y & \text{乗　算} \\ \dfrac{\delta_x}{y} + \dfrac{1}{y}\left\{\dfrac{\delta_y}{y} + \left(\dfrac{\delta_y}{y}\right)^2\right\}(x+\delta_x) & \text{除　算} \end{cases} \qquad (5)$$

ところが，(5)式そのままでは第二要素の数としての表現，倍長精度実数の指数部の制限等で実際のプログラムに適用できない。すなわち我々は，第二要素および仮数部の上位四要素を各operandについて倍長精度実数化し，誤差の仮数部を求め，整数型に直して格納する。その指数部は，第三要素および仮数部の長さを整数型のままで計算する。

乗除算の場合は，各operandの仮数部と指数部は各々分離して演算できるので，誤差の計算も加減算にくらべて容易である。図3に乗算の場合の誤差計算の流れ図を示す。

この流れ図において(*)印の部分で，倍長精度実数→整数の変換を行っている。このステートメントで倍長精度実数の演算誤差は充分に保証できる。なぜなら，$x, y, \delta x, \delta y$は正数であり，(5)式に減算がないので桁落ちの心配がない事と，ハードウエアにおける倍長精度実数の仮数部の桁数は，単精度整数の桁数よりも数桁多いのがふつうである事によるものである。

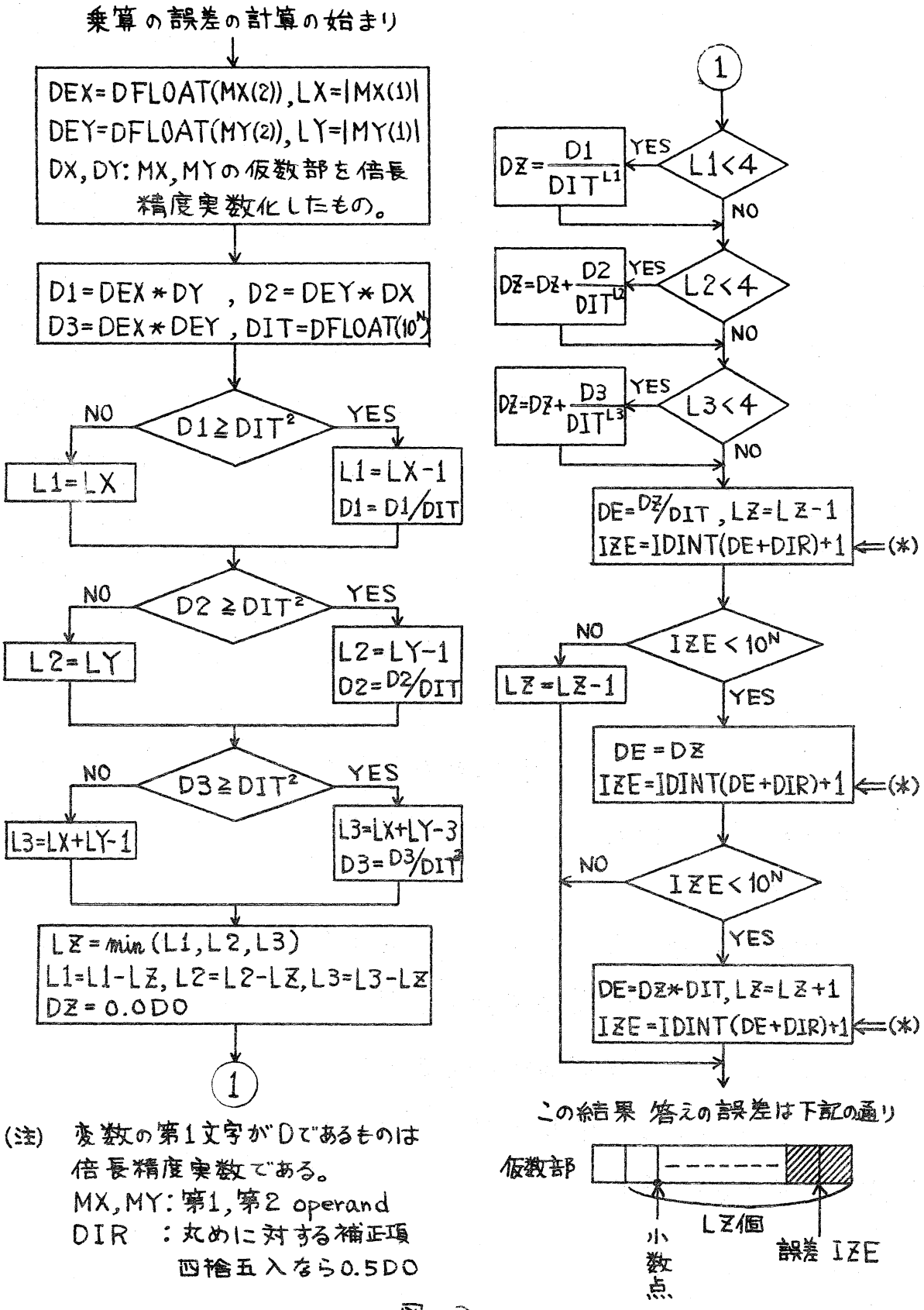
(3) 仮数部の長さはどのような場合にも4より小さくしない。これは(5)式において，除算の誤差評価を行う式にもとづいている。したがって仮数部の長さが4より小さくなる場合の処理は，preset parameterで次の三つの場合を指定する事ができる。

(i) 仮数部の長さを4とし，誤差を最大にして計算を続行する。（最大誤差$=10^{2N}-1$）

(ii) (i)と同一だがwarningを出す。

(iii) errorを印字し，計算を止める。

(4) S型の(2)に同じ。



(注) 変数の第1文字がDであるものは倍長精度実数である。
MX, MY: 第1, 第2 operand
DIR : 丸めに対する補正項
四捨五入なら0.5D0

図 3

§8. Interval arithmeticを行なう多重精度演算プログラム（I型）

(1) 一個のinterval numberの上，下限を各々，S型の表現であらわしたもので，一つの二次元配列として取扱う。

(2) interval arithmeticを行なっている。

$$[A_1,A_2]\,\textcircled{OP}\,[B_1,B_2]=[C_1,C_2]$$

$$\begin{cases} C_2=\max([A_1\,\textcircled{OP}\,B_1]_U,[A_1\,\textcircled{OP}\,B_2]_U,[A_2\,\textcircled{OP}\,B_1]_U,[A_2\,\textcircled{OP}\,B_2]_U) \\ C_1=\min([A_1\,\textcircled{OP}\,B_1]_L,[A_1\,\textcircled{OP}\,B_2]_L,[A_2\,\textcircled{OP}\,B_1]_L,[A_2\,\textcircled{OP}\,B_2]_L) \end{cases} \tag{6}$$

ただし

$$[X]_U\begin{cases} X\text{の丸めは切り上げ} & X\geqq 0 \\ X\text{の丸めは切り捨て} & X<0 \end{cases}$$

$$[X]_L\begin{cases} X\text{の丸めは切り捨て} & X\geqq 0 \\ X\text{の丸めは切り上げ} & X<0 \end{cases}$$

一般的には，上記の(6)式でよいが，四則演算各々の場合を以下に示す。

加算　$C_1=[A_1+B_1]_L$，　$C_2=[A_2+B_2]_U$

減算　$C_1=[A_1-B_2]_L$，　$C_2=[A_2-B_1]_U$

乗算

上段 C_2 下段 C_1	$A_2\leqq 0$	$A_1\times A_2<0$	$A_1\geqq 0$
$B_2\leqq 0$	$[A_1\times B_1]_U$ $[A_2\times B_2]_L$	$[A_1\times B_1]_U$ $[A_2\times B_1]_L$	$[A_1\times B_2]_U$ $[A_2\times B_1]_L$
$B_1\times B_2<0$	$[A_1\times B_1]_U$ $[A_1\times B_2]_L$	$\max([A_1\times B_1]_U,[A_2\times B_2]_U)$ $\min([A_1\times B_2]_L,[A_2\times B_1]_L)$	$[A_2\times B_2]_U$ $[A_2\times B_1]_L$
$B_1\geqq 0$	$[A_2\times B_1]_U$ $[A_1\times B_2]_L$	$[A_2\times B_2]_U$ $[A_1\times B_2]_L$	$[A_2\times B_2]_U$ $[A_1\times B_1]_L$

表 1

除算

上段 C_2 下段 C_1	$A_2 \leqq 0$	$A_1 \times A_2 < 0$	$A_1 \geqq 0$
$B_2 < 0$	$[A_2 \div B_1]_U$ $[A_1 \div B_2]_L$	$[A_2 \div B_2]_U$ $[A_1 \div B_2]_L$	$[A_2 \div B_2]_U$ $[A_1 \div B_1]_L$
$B_1 \times B_2 \leqq 0$	不　能	不　定	不　能
$B_1 > 0$	$[A_1 \div B_1]_U$ $[A_2 \div B_2]_L$	$[A_1 \div B_1]_U$ $[A_2 \div B_1]_L$	$[A_1 \div B_2]_U$ $[A_2 \div B_1]_L$

表　2

(3)　S型の(2)に同じ

§9．演算時間と使用例

例題(1)：演算時間の測定（M,S,E,I型）

条件　N=5, 55桁相当の仮数部の長さをもつ。

結果　表3

例題(2)：πの計算（M,S型）

計算法　文献(5)

$$\frac{\pi}{4} = 4\arctan\frac{1}{5} - \arctan\frac{1}{239} \qquad \text{ただし } \arctan x = \sum_{n=1}^{\infty} \frac{(-1)^{n-1} x^{2n-1}}{2n-1}$$

結果　表4

例題(3)：平方根の計算（I型）

解法　文献(4)

$$x_{n+1} = 1 + \frac{a-1}{x_n + 1}, \quad \lim_{n\to\infty} x_n = \sqrt{a}$$

条件　$a=2$, $x_0=[1,2]$

結果　表5

		NEAC 2200 series		FACOM 230 - 60	
		model 700	model 500	FORTRAN C	FORTRAN D
M型	ADD	～1	3～6	2～3	2
	SUB	～1	3～6	2～3	2
	MULT1	6	60～63	12～14	10～12
	DIVI1	7～9	70～80	14～18	10～16
	MULT2	～1	5～6	2	1～2
	DIVI2	1	5～6	2	1～2
	CVSHO	1	4～6	1～2	1
	CVLNG	1	8 9	1	2
S型	ADDS	～1	2～6	1～3	1～2
	SUBS	～1	2～6	1～3	1～2
	MULT1S	5～6	58～60	12～14	10～12
	DIVI1S	6～8	75～80	18～20	14～16
	MULT2S	～1	5～7	2	1～2
	DIVI2S	～1	5～7	2	2
	CVSHOS	～1	5	1～2	1
	CVLNGS	～1	8～9	1	2
E型	ADDE	1～2	13～15	2～3	2
	SUBE	1～2	13～15	2～3	2
	MULT1E	5～6	50～65	13～16	8～11
	DIVI1E	6～10	85～95	20～23	15～17
	MULT2E	1	8～10	2	1～2
	DIVI2E	1	8～10	2 3	2
I型	ADDI	3～4	25～32	7～9	5～6
	SUBI	3～4	25～32	7～9	5～6
	MULT1I	17～18,35	160～173,335	30～34,62	23～25,45
	DIVI1I	13～17	150～170	31～45	27～30
	MULT2I	3	23～27	5～9	4～5
	DIVI2I	3	23～26	5～7	4～5

表 3　　(単位: msec)

		NEAC 2200 series		FACOM 230 - 60		HARP
		model 700	model 500	FORTRAN C	FORTRAN D	5020 E
M型	(1)	0.369	3.666	0.902	0.604	1
	(2)	2.239	22.949	5.148	3.424	5
	(3)	15.762	163.235	34.667	23.785	37
S型	(1)	0.340	3.531	0.877	0.706	1
	(2)	2.010	21.611	5.004	3.808	6
	(3)	13.947	151.540	33.631	25.234	40

(単位: sec)

(HARP5020E　　: N=4　(1) 44桁, (2) 84桁, (3) 164桁
(HARP5020E以外 : N=5　(1) 55桁, (2) 105桁, (3) 205桁

表 4

例題 (4) ：連立一次方程式の解（S,E,I型）

解法　掃出法　文献(6)　pp 238－246

問題　AX=B　A：12次Hilbert 行列

B (右辺)	X (解)
549947480	27720
505269308	360360
467420948	360360
434913308	360360
406674622	720720
381906956	12252240
360002020	12252240
340487160	232792560
322989141	232792560
307209091	232792560
292904731	232792560
279877507	5354228880

結果　S型　表6(切り上げ), 表7(四捨五入), 表8(切り捨て)

E型　表9

I型　表10

§10. まとめ

§9の例題(4)の結果をみると，解の広がりは，S型＜E型＜I型の順になっている。これは当然の結果であり，I型の解はその区間内に必ず真値が存在する。解の存在範囲がわかる事は，非常に大きな意味をもつものと考える。この例のごとく，直接的解法に対しては，E,I型は有効であり，誤差評価の目的をはたす事ができる。

反復法において，I型ではアルゴリズムそのものの誤差を含めて，解の存在範囲を決定しうる場合がある。しかし，平

方根の計算を $x_{n+1}=\frac{x_n^2+a}{2x_n}$ で行なう時の様に，その実行時のアルゴリズムによっては，intervalからはみ出したり，広がったりする場合もある。一方，E型においては，アルゴリズムそのものの誤差を求める事は本質的に不可能であるが，各ステップにおける演算誤差を求める事ができる。しかしながら，反復法では前回の誤差が波及しないので，§3に述べた様に，各ループ毎にoperandの長さを制御して，計算時間を短かくする事ができる。

これらM,S,E,I型4つの基本演算は，各々異なった機能をもっており，その特色に充分な注意をはらってもちいれば，まえがきに述べた目的に対して有効であるものと思われる。

参考文献

(1) R.E. Moore, "The automatic analysis and control of error in digital computing based on the use of interval numbers.", In L.B. Rall (Ed.), Error in Digital computation, Vol.1, 61-130, John Wiley and Sons, New York, (1965).

(2) C.V. Ramamoothy and M. Tsuchiya, "Microprogrammed significance arithmetic; A perspective and feasibility study.", AFIPS, Vol. 40, 653-673, (1972).

(3) I.D. Hill, "Algolithm 34 – Procedures for the basic arithmetic operations in multiple length working.", Computer J., Vol. 11, No 2 232-235, (1968).

(4) I.D. Hill, "Note on algolithm 34 – Procedures for the basic arithmetic operations in multiple length working.", Computer J., Vol. 12, No 1, 102-103, (1969).

(5) G.W. Reitwriesner, "An ENIAC Determination of π and e to more than 2000 Decimal Place.", MTAC, Vol. Ⅲ, No 29, 11-15, (1950).

(6) 雨宮綾夫, 田口武夫 編, "数値解析とFORTRAN.", 丸善, (1969).

(7) 大中幸三郎, 安井裕, "FORTRAN による多重精度演算用基本ルーチンについて.", 情報処理学会関西支部プログラミング言語研究会資料集, 25-49, 昭和46年10月.

```
I        50

LXI(J,1)          1          0          1  4142135623  7309504880  1688724209  6980785722  3296684316
LXI(J,2)          1          0          1  4142135623  7309504880  1688724209  6980785671  1078391074

I        51

LXI(J,1)          1          0          1  4142135623  7309504880  1688724209  6980785701  1128920360
LXI(J,2)          1          0          1  4142135623  7309504880  1688724209  6980785692  3246155030

I        52

LXI(J,1)          1          0          1  4142135623  7309504880  1688724209  6980785697  4726687062
LXI(J,2)          1          0          1  4142135623  7309504880  1688724209  6980785695  9648388328

I        53

LXI(J,1)          1          0          1  4142135623  7309504880  1688724209  6980785696  8481051229
LXI(J,2)          1          0          1  4142135623  7309504880  1688724209  6980785696  5894024160
```

表 5

```
*** TEST OF MELISW ***

 1          9     159953670     0    27720      0      0      0      0      0      0      0   3997
 2          9        201795     1        3  60359  99999  99999  99999  99999  99999  99999  99995
 3          9       6317956     1        3  60360      0      0      0      0      0      0    169
 4          9      85722015     1        3  60359  99999  99999  99999  99999  99999  99999  97659
 5          9     625972897     1        7  20720      0      0      0      0      0      0  17438
 6          9    2740495180     1      122  52239  99999  99999  99999  99999  99999  99999  22346
 7          9    7610371251     1      122  52240      0      0      0      0      0      2  18837
 8          8        137335     1     2327  92559  99999  99999  99999  99999  99999  99996
 9          8        161329     1     2327  92560      0      0      0      0      0      5
10          8        118350     1     2327  92559  99999  99999  99999  99999  99999  99997
11          9    4927609440     1     2327  92560      0      0      0      0      0      1  46014
12          9     888928464     1    53542  28879  99999  99999  99999  99999  99999  99999  73540

      CPTIME=       7910
```

表 9

ND OFF 0

LAN(I,1)=	1	LAN(I,2)=	0	27719	9999999999	9999999999	9999999999	9999998761	1003508954
LAN(I,1)=	1	LAN(I,2)=	1	3	6036000000	0	0	1	7045644813
LAN(I,1)=	1	LAN(I,2)=	1	3	6035999999	9999999999	9999999999	9999999942	9806974690
LAN(I,1)=	1	LAN(I,2)=	1	3	6036000000	0	0	815	2327644432
LAN(I,1)=	1	LAN(I,2)=	1	7	2071999999	9999999999	9999999999	9999993787	8434745427
LAN(I,1)=	1	LAN(I,2)=	1	122	5224000000	0	0	28176	2257276841
LAN(I,1)=	1	LAN(I,2)=	1	122	5223999999	9999999999	9999999999	9999919379	529125378 7
LAN(I,1)=	1	LAN(I,2)=	1	2327	9256000000	0	0	149262	123999098
LAN(I,1)=	1	LAN(I,2)=	1	2327	9255999999	9999999999	9999999999	9999821584	8319960672
LAN(I,1)=	1	LAN(I,2)=	1	2327	9256000000	0	0	132885	6954478409
LAN(I,1)=	1	LAN(I,2)=	1	2327	9255999999	9999999999	9999999999	9999943928	7329922470
LAN(I,1)=	1	LAN(I,2)=	1	53542	2888000000	0	0	10235	2383546660

CPTIME= 5904

表 6

ND OFF 5

LAN(I,1)=	1	LAN(I,2)=	0	27720	0	0	0	3825	778315372
LAN(I,1)=	1	LAN(I,2)=	1	3	6035999999	9999999999	9999999999	9999999994	9919212914
LAN(I,1)=	1	LAN(I,2)=	1	3	6036000000	0	0	161	5485987626
LAN(I,1)=	1	LAN(I,2)=	1	3	6035999999	9999999999	9999999999	9999997753	5885842558
LAN(I,1)=	1	LAN(I,2)=	1	7	2072000000	0	0	16745	6315444615
LAN(I,1)=	1	LAN(I,2)=	1	122	5223999999	9999999999	9999999999	9999925390	2683706551
LAN(I,1)=	1	LAN(I,2)=	1	122	5224000000	0	0	210350	724912489
LAN(I,1)=	1	LAN(I,2)=	1	2327	9255999999	9999999999	9999999999	9999615365	8464142122
LAN(I,1)=	1	LAN(I,2)=	1	2327	9256000000	0	0	454904	1948362469
LAN(I,1)=	1	LAN(I,2)=	1	2327	9255999999	9999999999	9999999999	9999664275	1795727008
LAN(I,1)=	1	LAN(I,2)=	1	2327	9256000000	0	0	140530	1627888217
LAN(I,1)=	1	LAN(I,2)=	1	53542	2887999999	9999999999	9999999999	9999974527	7904702841

CPTIME= 5883

表 7

ND OFF 10

LAN(I,1)=	1	LAN(I,2)=	0	27720	0	0	0	7559	800639221
LAN(I,1)=	1	LAN(I,2)=	1	3	6035999999	9999999999	9999999999	9999999990	3855301865
LAN(I,1)=	1	LAN(I,2)=	1	3	6036000000	0	0	303	2431829879
LAN(I,1)=	1	LAN(I,2)=	1	3	6035999999	9999999999	9999999999	9999995858	5257895935
LAN(I,1)=	1	LAN(I,2)=	1	7	2072000000	0	0	30419	4258248690
LAN(I,1)=	1	LAN(I,2)=	1	122	5223999999	9999999999	9999999999	9999866130	7493012262
LAN(I,1)=	1	LAN(I,2)=	1	122	5224000000	0	0	373485	9815004745
LAN(I,1)=	1	LAN(I,2)=	1	2327	9255999999	9999999999	9999999999	9999323201	4257172182
LAN(I,1)=	1	LAN(I,2)=	1	2327	9256000000	0	0	794187	1822485984
LAN(I,1)=	1	LAN(I,2)=	1	2327	9255999999	9999999999	9999999999	9999417905	9951523814
LAN(I,1)=	1	LAN(I,2)=	1	2327	9256000000	0	0	242174	8236285615
LAN(I,1)=	1	LAN(I,2)=	1	53542	2887999999	9999999999	9999999999	9999956342	1340318861

CPTIME= 5886

表 8

*** TEST OF MILISW ***

DIMENSION 12

1	*** UPPER BOUND ***	1	0	27720	0	0	0	180857502	9504715882
	*** LOWER BOUND ***	1	0	27719	9999999999	9999999999	9999999999	9819136776	4033802802
2	*** UPPER BOUND ***	1	1	3	6036000000	0	0	228167	1900458533
	*** LOWER BOUND ***	1	1	3	6035999999	9999999999	9999999999	9999771840	2210681785
3	*** UPPER BOUND ***	1	1	3	6036000000	0	0	7143332	2137587201
	*** LOWER BOUND ***	1	1	3	6035999999	9999999999	9999999999	9992856430	5590155078
4	*** UPPER BOUND ***	1	1	3	6036000000	0	0	96922280	504248997
	*** LOWER BOUND ***	1	1	3	6035999999	9999999999	9999999999	9903080999	5031564270
5	*** UPPER BOUND ***	1	1	7	2072000000	0	0	707725865	6002258555
	*** LOWER BOUND ***	1	1	7	2071999999	9999999999	9999999999	9292249798	2901465415
6	*** UPPER BOUND ***	1	1	122	5224000000	0	0	3098470633	776735281
	*** LOWER BOUND ***	1	1	122	5223999999	9999999999	9999999999	6901637402	9581642096
7	*** UPPER BOUND ***	1	1	122	5224000000	0	0	8604057762	1644920291
	*** LOWER BOUND ***	1	1	122	5223999999	9999999999	9999999999	1395638545	1897663523
8	*** UPPER BOUND ***	1	1	2327	9256000000	0	1	5527052412	8912712477
	*** LOWER BOUND ***	1	1	2327	9255999999	9999999999	9999999998	4473501544	7758029184
9	*** UPPER BOUND ***	1	1	2327	9256000000	0	1	8239197867	7228131210
	*** LOWER BOUND ***	1	1	2327	9255999999	9999999999	9999999998	1760148312	9778718644
0	*** UPPER BOUND ***	1	1	2327	9256000000	0	1	3380671059	546529037
	*** LOWER BOUND ***	1	1	2327	9255999999	9999999999	9999999998	6619810627	1791420175
1	*** UPPER BOUND ***	1	1	2327	9256000000	0	0	5571098905	6245472406
	*** LOWER BOUND ***	1	1	2327	9255999999	9999999999	9999999999	4428699768	2275834535
2	*** UPPER BOUND ***	1	1	53542	2888000000	0	0	1005056227	2759516321
	*** LOWER BOUND ***	1	1	53542	2887999999	9999999999	9999999999	8994980217	3570965814

CPTIME= 30922

表 10