

《論文》

電子式卓上計算機用組み込み関数の一演算方式*

浅田 篤** 吉田 幸弘**

Abstract

In the case when a few class of transcendental functions are to be calculated by and electronic desk top calculator, they are usually implemented by some fixed programs provided in the calculator.

The present paper considers an arithmetical method for transcendental functions which is to be employed in an electronic desk top calculator. The procedure consists in a modified application of Euler's method in numerical integration. By providing in a calculator the equipments with those functions excuting basic operations in the procedure, we can synthesize the one with much more reduced hard wares than those which are equipped with fixed programs.

1. まえがき

電子式卓上計算機（以下電卓と呼ぶ）が開発されて以来プログラム式電卓より高度な科学用電卓の中には、関数計算機能をもったものもあるが、一般的には加減算、乗算、除算およびプログラム機能によって構成されている。

通常プログラム式電卓を用いて関数計算を行う場合に使われる演算方法は各関数に対するマクローリン展開¹⁾や、最良近似法²⁾である。一方組み込み関数として用意している科学用電卓の場合には、マクローリン展開かまたはその他の方法、たとえば擬似乗除算法³⁾、CORDIC法^{4),5)}、STL法⁶⁾などがあり、小柳一渡⁷⁾一萩原⁷⁾がこれ等の算法について比較検討している。これ等の(3)~(6)の手法は演算の速度、精度を上げるという目的から見れば妥当である。しかしながら、あらかじめ計算された定数を用いる必要があり、電卓の設計のように10進法を基本とする場合には定数を記憶しておくのに多くのビット数を要し、ハード構成上多くの電子回路を必要とする点では問題がある。

最近のようにLSIを使用して電卓を設計するという立場では、ハードの規模にあまり制約されない高級電卓は別として、出来るだけ少ないチップ数で関数計算の実用的な数値の範囲で求められる演算方法が要求さ

れる。

本論で述べる方法は、この目的のために考えられたものであり、指数関数、三角関数等各関数値を実用的に求める一演算方式である。これは数値積分法におけるオイラー法が基本になっているが、奇数回目、偶数回目のインクリメントで近似法を変え補正効果を持たせながら数値積分を逐次進めていくものである。

逆関数に対してはニュートン・ラプソン法の1次近似としてインクリメントの累積値を用い1回目の使用で補正値を求めている。この演算法を実行するに当たり、レジスター長の決定、誤差を調べるためシミュレーションを行い、また実際に装置を設計し実用性を確かめている。

従来はソフト面を中心にした演算法の開発に重点が置かれハードに直結したものではなかったので組み込み関数をハードで用意しておく場合の目的には合致していない。この点から考えて、本方式はインクリメント法によるデジタル関数発生装置、計算尺の電子化、電卓用組み込み関数等広く応用できるものである。

2. 演算法の基本

$f(x)$ を x に関する任意関数として、この関数のある x_0 に関して、

$$\left. \begin{aligned} f(x+x_0) &= \sum_{m=0}^{\infty} \frac{f^{(m)}(x_0)}{m!} x^m \\ f^{(m)}(x_0) &\triangleq \left. \frac{d^m f(x)}{dx^m} \right|_{x=x_0} \end{aligned} \right\} (1)*$$

* Electronic Desk Top Calculator Oriented Arithmetical Method of Transcendental Functions, by Atsushi ASADA & Yukihiko YOSHIDA (Business Machine Group, Industrial Instruments Division, Sharp Corporation)

** シャープ(株)産業機器事業本部事務機事業部

* \triangleq は '定義により等しい' ことを表す

いま、いくつかの x の値 $x=x_0, x_0+h, \dots, x_0+nh$ (h : インCREMENT) に関して関数値が要求されるとき、 $x_n \triangleq x_0+nh$ ($n=0, 1, 2, \dots, N$) として各 $x_{n+1} \equiv x_n+h$ に対して $f(x_n+h)$ を次式のように展開する。

$$f(x_n+h) = \sum_{m=0}^{\infty} \frac{f^{(m)}(x_n)}{m!} h^m \quad (2)$$

ここで、 $f_n = f(x_n)$ とおけば、 f_{n+1} は次のように与えられる。

$$f_{n+1} = \sum_{m=0}^{\infty} \frac{f_n^{(m)} h^m}{m!} \quad (3)$$

以下で(3)式において第3項までとって近似をする場合を考える。このとき各 f_{n+1} は

$$f_{n+1} = f_n + hf'_n + \frac{1}{2}h^2f''_n \quad (4)$$

となるが、この(4)式はINCREMENTによる一つの数値積分である。本文では、この近似式をそのまま用いないで、以下で示すような近似法をとることにする。Fig. 1 から

$$\int_{x_{2n-2}}^{x_{2n-1}} f'(x) dx \approx h(f'_{2n-2} + hf''_{2n-2}) \quad (5)$$

$$\int_{x_{2n-1}}^{x_{2n}} f'(x) dx \approx hf'_{2n-1} \quad (6)$$

であるから

$$\left. \begin{aligned} f_{2n-1} &= f_{2n-2} + h(f'_{2n-2} + hf''_{2n-2}) \\ f_{2n} &= f_{2n-1} + hf'_{2n-1} \end{aligned} \right\} \quad (7)$$

または

$$\left. \begin{aligned} f_{2n-1} &= f_{2n-2} + f'_{2n-2} \\ f_{2n} &= f_{2n-1} + h(f'_{2n-1} + hf''_{2n-1}) \end{aligned} \right\} \quad (8)$$

が成立する。

そこで、各 f_i について上式で表わされるような奇

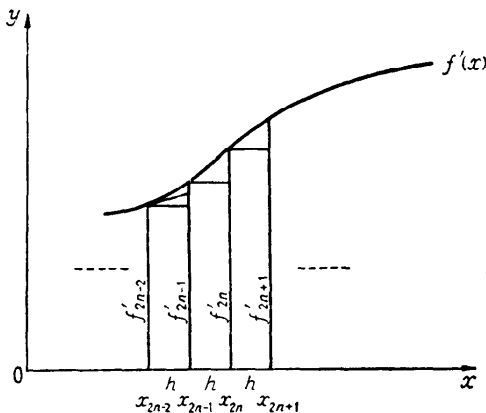


図-1 (5), (6)に対する説明図
Fig. 1 Explaining example for (5) and (6)

数番目の対する f_i と偶数番目の i に対する f_i に関する近似式を用いて、数値積分を行う場合に注目すれば、(4)式を用いて行う場合よりも計算手数が少なく済み、しかも精度は同程度である。

さて、与えられた y に対して、 $x=f^{-1}(y)$ を求めることを考える。いま (7), (8) を用いてこれを求める場合、たとえば単調関数のとき次の手続きが考えられる。

(操作 I)

$f(x_i) < y$ (f : 単調増加関数) または $f(x_i) > y$ (f : 単調減少関数) とするような x_i ($0 \leq i \leq N$) を選んで $\xi = x_i + h$ ($= x_{i+1}$) として次へ移る。

(操作 II)

$f(\xi) > y$ (f : 単調増加関数) または $f(\xi) < y$ (f : 単調減少関数) であればそのときの ξ ($\triangleq x_N$) は $x=f^{-1}(y)$ に対する近似解である。 $f(\xi) < y$ (f : 単調増加関数) または $f(\xi) > y$ (f : 単調減少関数) であれば $\xi \leftarrow \xi + h$ として操作 II を繰返す。

この手続きで求められる $x \equiv f^{-1}(y)$ の精度は明らかに h の程度である。

そこで、以下のようにこれを1次近似としてニュートン・ラプソン法で補正することを考える。すなわち、うえで求められた近似解 $\xi = x_N$ に対して、

$$J \triangleq \begin{cases} 2N-2; \text{うへの近似解 } \xi \text{ が最後に } f_{2n} \text{ を用いて算出されたとき} \\ 2N-1; \text{そおでないとき} \end{cases}$$

とおいたとき

$$x = Nh - \frac{f_{J+1} - y}{f'_J} \quad (f: \text{単調増加関数}) \quad (9)$$

$$x = Nh + \frac{f_{J+1} - y}{f'_J} \quad (f: \text{単調減少関数}) \quad (10)$$

として、この x を $f^{-1}(y)$ の近似解とする。このとき、ニュートン・ラプソン法の近似精度から h^2 の程度まで精度を高めることが出来る。

3. 各関数計算への応用

ここでは、2章で述べた演算法を用いて関数計算を行う場合の例について説明する。

3.1 指数関数 ($y=e^x$)

この場合には $y' = y'' = y$ であるから各 y_i は次式のように与えられる。

$$\left. \begin{aligned} y_{2n-1} &= y_{2n-2} + h(y_{2n-2} + hy_{2n-2}) \\ y_{2n} &= y_{2n-1} + hy_{2n-1} \\ y_J &\equiv e^x \end{aligned} \right\} \quad (11)$$

3.2 三角関数 ($y = \sin x, z = \cos x$)

この場合には

$y' = z, y'' = -y, z' = -y, z'' = -z$ であるから
各 y_i, z_i は次式で与えられる。

$$\left. \begin{aligned} y_{2n-1} &= y_{2n-2} + h(x_{2n-2} - h y_{2n-2}) \\ y_{2n} &= y_{2n-1} + h x_{2n-1} \\ z_{2n-1} &= z_{2n-2} - h(y_{2n-2} + h z_{2n-2}) \\ z_{2n} &= z_{2n-1} - h y_{2n-1} \\ y_j &\cong \sin x \\ z_j &\cong \cos x \end{aligned} \right\} \quad (12)$$

3.3 双曲線関数 ($y = \sinh x, z = \cosh x$)

この場合には、 $y' = z, y'' = y, z' = y, z'' = z$ であるから、各 y_i, z_i は次式のように与えられる。

$$\left. \begin{aligned} y_{2n-1} &= y_{2n-2} + h(x_{2n-2} + h y_{2n-2}) \\ y_{2n} &= y_{2n-1} + h z_{2n-1} \\ z_{2n-1} &= z_{2n-2} + h(y_{2n-2} + h z_{2n-2}) \\ z_{2n} &= z_{2n-1} + h y_{2n-1} \\ y_j &\cong \sinh x \\ z_j &\cong \cosh x \end{aligned} \right\} \quad (13)$$

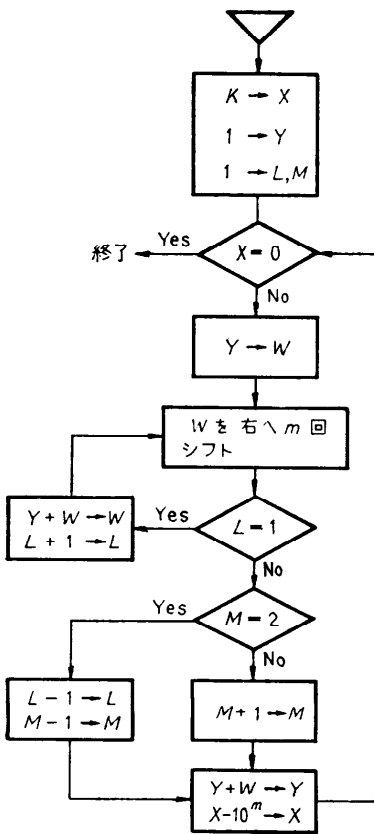


図-2 e^x 算出のフローチャート
Fig. 2 Flow chart for e^x

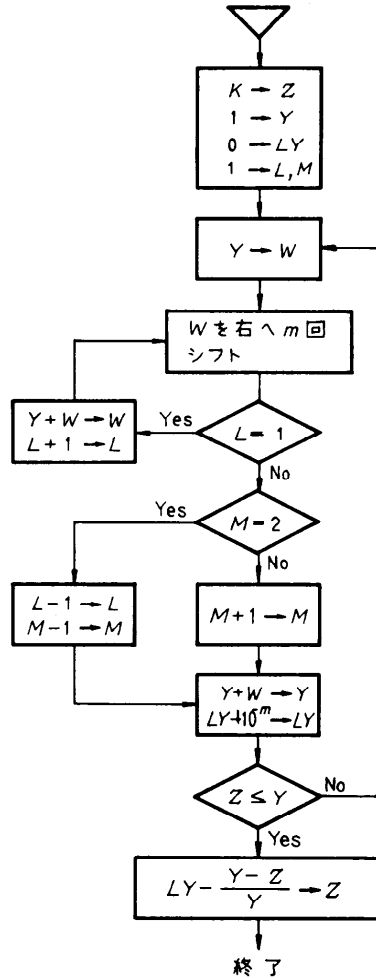


図-3 $\ln x (x \geq 1)$ 算出のフローチャート
Fig. 3 Flow chart for $\ln x (x \geq 1)$

3.4 演算の実行

以上の考察に基づいて演算を実行したが、以下ではその概要および結果について述べる。

Fig. 2~Fig. 5 はそれぞれ $y = e^x, y = \ln x, y = \sin x, y = \sin^{-1} x$ の算出手順のフローチャートを表わす。

これらのうち、特に $y = e^x$ および $y = \ln x$ について着目し、その概要を述べよう。各フローにおいて、インクリメント $h = 10^{-m}$ とし、 K は x に対するある一定値であり、 X, Y, Z, W_1, W_2 はレジスタで L, M は偶数番目と奇数番目で演算式を使いわけするためにフロー上のブランチング用として便宜的に使用する。

Fig. 2 の e^x に関して考えると、 K をまずレジスタ

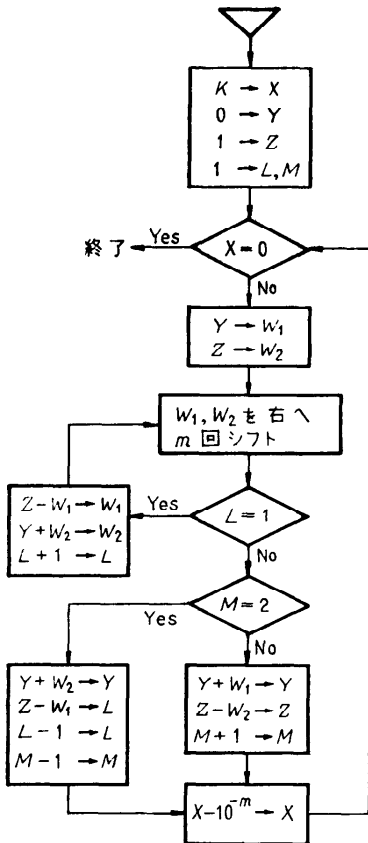


図-4 $\sin x, \cos x$ 算出のフローチャート
Fig. 4 Flow chart for $\sin x$ and $\cos x$

X に移し、レジスタ Y には $X=0$ のときの初期 1 が記憶される。 $X=0$ は (11) 式の演算が終了したか値どうかを判定するためのものである。もし $X=0$ でなければレジスタ Y の内容をレジスタ W に移し、 W の内容を m 回右にシフトする。もし (11) 式の前式を実行するのに Y と W も加算し W に入れる。その W を m 回右にシフトし、 Y と W を加算しインクrementを X から減じ $X=0$ の判定を行い、もし $X \neq 0$ なら Y を W に移し、(11) の後式を実行するのに W を m 回右にシフトし、 Y と W を加える。このような繰返しの結果 $X=0$ になれば計算を終了する。

同ように、 $\ln x$ を求める場合について説明する。 Z には x に対する一定値 K がまず記憶される。 Y には各インクrementに対し e^Y が求まっており繰返し計算の中で $Z \leq Y$ が満たされれば、 $N \times 10^{-m} \geq \ln K$ なる

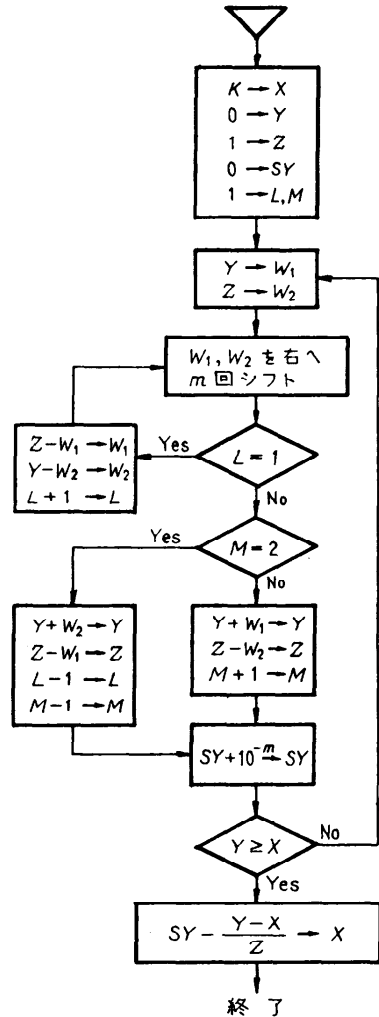


図-5 $\sin^{-1} x$ 算出のフローチャート
Fig. 5 Flow chart for $\sin^{-1} x$

インクrementの累積が LY に求められる。その後 (9) に示された補正式に従い実行すれば $\ln K$ が算出されることになる。 $\sin x, \sin^{-1} x$ も同様な説明がなされるので省略する。

4. シミュレーション

以上述べた演算法を実用化する前にどの程度のレジスタ長が必要であるかを決定することが必要となる。同時に各関数値に対する精度を調べておくうえでも重要なことである。このためにシミュレーションを行ったので以下にその概略を示す。

4.1 シミュレーションの手順の概要

レジスターの桁数と精度との関係を知るためには関数値を求めるのに用いる各レジスターの動きを正確にシミュレーションしておかなければならない。

どの程度の桁数のレジスターを選ぶことが適当であるかを求めるために、可変長のレジスターを用いてシミュレーションを行った。

レジスター長は、得られる精度に関するファクターであり、インCREMENTを $k=10^{-m}$ に選べば h^2 の精度を得るためのレジスター長としては $4m$ 桁程度と評価できる。この妥当性をシミュレーションにて確認する方法として、 $(4m-1)$, $4m$, $(4m+1)$ の3つのレジスター長を選ぶことが出来る。シミュレーションではレジスター長を可変とするためデータとして入力し得るようにしている。各レジスターの内容は1桁単位の10進配列として処理し、添字付き変数、 $X(M)$ ……、 $X(I)$, ……、 $X(1)$ を1つのレジスターとして構成する。

上述の各関数に対するフローチャートを忠実に実行するため、転送、右シフト、判定等基本命令や Add, Sub, Div の演算命令もハードと等価な機能をもつようにプログラムし、演算順序を制御している。このようにして桁数と繰返し回数によって関数フローを実行し、関数値を記憶しているレジスターや小数点位置を出力し、IBM-360 の組み込み関数値と比較してその妥当性を確認した。

なお、実設計への応用を考えて、 $m=3$ としてシミュレーションを行なっている。このような考え方で2章で述べた関数すべてについてシミュレーションを行い、レジスター長として $4m$ にとれば h^2 程度の精度が得られることを確認した。

4.2 シミュレーションによる誤差傾向

精度の評価には、絶対誤差を用いる場合がある。絶対誤差はある関数 $f(x)$ の x に対する真値 T と近似値 A との差の絶対値で定義される。一方相対誤差は真値の絶対値と絶対誤差との比で定義される故、ある x に対し、 T が0となるような場合には意味を持たない。

たとえば $y=\sin x$ のとき

$$\left| \frac{\Delta y}{y} \right| = \left| \frac{\Delta x \cos x}{\sin x} \right| \tag{14}$$

となり、 $x=n\pi$ の近傍では、 $\sin \approx 0$ で $|\Delta y/y|$ は非常に大きくなりレジスター長が有限である限り、相対誤差による一定の精度を保証することは非常に困難で

ある。同ようなことは $\ln x$ の $x=1$ の近傍においても生じ、これ等は関数の特有な性質によるものといえる。そこで、ここでは12桁レジスターの装置設計で固定小数点方式を採用して、絶対誤差に近い有効桁数により評価を行なった。定量的には最大誤差の範囲で評価されるが、装置設計ではこれだけでは不十分で実際の誤差の傾向を調べるの方が重要である。このためにわれわれは実設計への応用上 $m=3$ として実用的立場からシミュレーションにより調べた。

Table 1 には、有効桁数が6となるような x の範囲がまとめられている。

Fig. 6 は、 e^x , $\sin x$, $\tan^{-1}x$ についてそれぞれ本算方式において生じる誤差の傾向を示している。特に e^x の場合は、本方法の特徴である累積値に対する過修正、不足修正の傾向がよく出ている。

また、 $\tan x$ の場合の誤差傾向は Fig. 7 のように

表-1 有効桁数が保証される x の範囲

Table. 1 6 significant digits can be obtained as follows

関数	x の範囲	備考
$\sin x, \cos x, \tan x$	$ x < 10$	$\tan((2n+1)\pi/2 \pm dx)$ は除く
$\sin^{-1}x$	$0.9999 \geq x \geq 0$	$x=1-dx$ の近傍除く
$\cos^{-1}x$	$0.9999 \geq x \geq 0$	$x=1-dx$ の近傍除く
$\tan^{-1}x$	$0 \leq x \leq 10^9$	$x=dx$ の近傍除く
e^x	$ x \leq 13.6$	
$\log_e x, \log_{10} x$	$0 < x \leq 10^7$	
$\sinh x, \cosh x, \tanh x$	$ x \leq 10$	

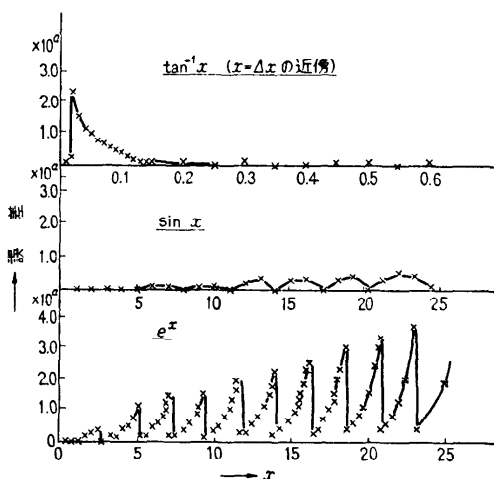


図-6 $e^x, \sin x, \tan^{-1}x$ に対する誤差傾向 (a は最上位桁より7桁目の指数を表わす)

Fig. 6 Error tendency for $e^x, \sin x$, and $\tan^{-1}x$

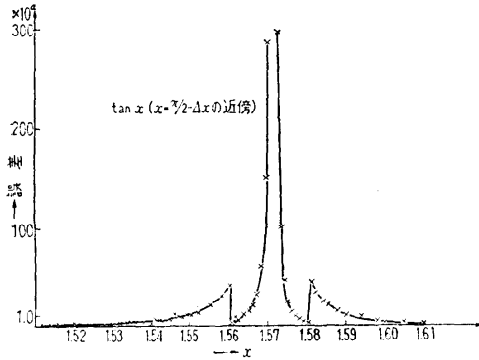


図-7 tan x に対する誤差傾向 (αは最上位桁より7桁目の指数を表わす)
Fig. 7 Error tendency for tan x

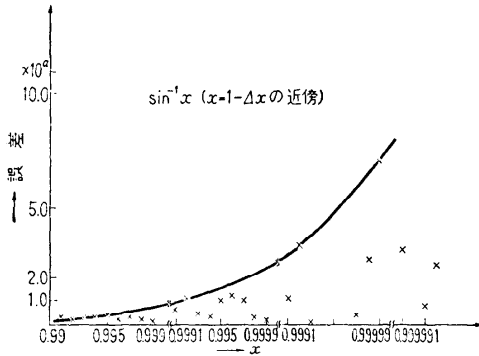


図-8 sin⁻¹x に対する誤差傾向 (αは最上位桁より7目桁の指数を表わす)
Fig. 8 Error tendency for sin⁻¹x

なるが、ここで $x = \pi/2$ の近傍で誤差が改良されているのは、 $x = \alpha X$ とすると、

$$\left| \frac{\Delta y}{y} \right| = \left| \frac{2\alpha \Delta X}{\sin 2\alpha X} \right| \quad (15)$$

であり、 $\alpha \cong 1$ となるような α として、実験的に 0.99999987 を選んだためである。一方 $\sin^{-1}x$ の場合、Fig. 8 に示すように誤差は離散的であるが、これは補正式による補正効果の影響によっている。なお $\ln x$ については、 $0 < x < 10^7$ の範囲内においては、Table 1が示すようにこの方式適用に問題が起らない。

以下では、本手法を用いた場合の誤差を指数関数、三角関数について定量的に評価してみる。その導出は紙数の関係で割愛し結果のみを述べる。

(a) 指数関数の場合: 各 y_i は、 $\gamma \triangleq (1+h) \times (1+h+h^2)$ に対して

$$\left. \begin{aligned} y_{2n} &= e^{\gamma^{2n}} \\ y_{2n-1} &= (1+h+h^2)e^{(n-1)\gamma} \end{aligned} \right\} \quad (16)$$

で与えられ、これらに対する(相対)誤差式はそれぞれ

$$1 - e^{-(\gamma^{2n}-2h)}, \quad 1 - (1+h+h^2)e^{-(n-1)\gamma}$$

となる。

たとえば、 $h=10^{-3}$ にとれば相対誤差を 10^{-6} 以内にするためには、 $2nh < 8$ となる。

(b) 三角関数の場合: $\theta \triangleq \tan^{-1} \frac{h(h^2-2)}{1-2h^2}$,

$\eta \triangleq \sqrt{1+h^6}$ に対して各 y_i, z_i は

$$\left. \begin{aligned} y_{2n} &= -\eta^n \sin n\theta \\ y_{2n-1} &= \eta^{n-1} \{ (h^2-1) \cos(n-1)\theta + h \cos(n-1)\theta \} \\ z_{2n} &= \eta^n \cos n\theta \\ z_{2n-1} &= \eta^{n-1} \{ (1-h^2) \cos(n-1)\theta + h \sin(n-1)\theta \} \end{aligned} \right\} \quad (17)$$

で与えられ、これらに対する(相対)誤差式は、それぞれ

$$\begin{aligned} &1 + \eta^n \sin n\theta / \sin 2nh \\ &1 - \eta^{n-1} \{ (h^2-1) \sin(n-1)\theta + h \cos(n-1)\theta \} / \sin(2n-1)\theta \\ &1 - \eta^n \cos n\theta / \cos 2nh \\ &1 - \eta^{n-1} \{ (1-h^2) \cos(n-1)\theta + h \sin(n-1)\theta \} / \cos(2n-1)\theta \end{aligned}$$

である。たとえば、 $0 \leq x \leq 2\pi$ の範囲で調べると、 $\sin x$ の場合、 $h=10^{-3}$ にとれば相対誤差を 10^{-6} 以内にするためには、 $2nh < 2\pi$ ($\pi, 2\pi$ の近傍を除く) となり、 $\cos x$ の場合も $2nh < 2\pi$ ($\pi/2, 3\pi/2$ の近傍を除く) となる。

5. 結び

本文で考察したこの演算法は、主として電卓用組み込み関数用として考えられたものであり、簡単な方法で関数値を求めることを目的としている。

すでに考察したように、インクリメントによって逐次数値積分を進めていくために、本方法の定量的な誤差の解析することはむずかしい。したがって、装置設計をする場合には、定量的な解析だけでは不十分であるので、われわれはレジスター長の決定や誤差の傾向を調べるためにシミュレーションを行った。実設計への応用を考えて $m=3$ に限定して詳細に調べ実用性を確めた。

従来は、組み込み関数を電卓で計算する場合、ソフト面を中心にした関数計算法であったが、本論で考察した方式に基づいて、Fig. 2~5で示されるフローチ

ャートの各段階の基本命令を遂行するような装置をハード的に構成して組み入れた場合には、全体に要する装置部品をより少なくすることが可能となる。

実際の計算には、よく知られているようにスケールリング処理を行い、その上で本方法を応用することが必要であるが、本論では演算法の基本そのものの内容とシミュレーションに重点を置いたため、この点については割愛した。シミュレーション結果も膨大なデータであり本方法の説明に必要と思われるデータに限り付加した。なお $\ln x = \int_1^x \frac{dt}{t}$ を数値積分（シンプソン公式）で求めて e^x を逆に $\ln x$ からニュートン法で求めるという方法も考えられることを付記しておく。

最後に組み込み関数用一演算法の実用化に当り、種々御指導いただいた大阪大学工学部尾崎教授、白川助教授、事務機一技 鷲塚部長、シミュレーションや装置設計に当られた事務機5技 猪崎、前川、同一技 竹田、中川各君に深謝致します。

参 考 文 献

1) A. Ralston & H. S. Wilf: Mathematical Met-

hod for Digital Computers, Vol. 1, p. 9, John Wiley & Sons, Inc. 1960.

- 2) 一松 信: 数値解析, p. 97, 税務経理協会, 東京 (1971)
- 3) J. E. Meggitt: Pseudo Division and Pseudo Multiplication Process, p. 210, IBM Journal April 1962
- 4) J. S. Walther: A unified Algorithm for Elementary Functions, Vol. 38, p. 379, AFIPS. Conference Proceedings, SJCC 1971
- 5) J. E. Volder: The CORDIC Trigonometric Computing Technique, p. 330, IRE Trans. E. C. September, 1959
- 6) W. H. Specker: A Class of Algorithms for $\ln x$, $\exp x$, $\sin x$, $\cos x$, $\tan^{-1}x$, and $\cot^{-1}x$ p. 85, IEEE Trans. E. C. February, 1965
- 7) 小柳, 渡辺, 萩原: マイクロ・プログラムによる初等関数近似, p. 171, 昭和48年情報処理学会第14回大会

(昭和49年5月29日受付)

(昭和49年7月26日再受付)