数理解析研究所講究録
215巻 1974年 66-74

マイクロプログラムによる
初等関数の計算法とその評価

京大 工　　小柳 滋
萩原 宏

§1. 序

最近マイクロプログラミング方式の計算機が普及し、ソフトウェアをファームウェア化して効率を向上させることが可能となった。初等関数近似を計算機でおこなう場合、多項式近似を用いるのが一般的であるが、マイクロプログラムを用いて高速化するときは乗除算を数回用いなければならないのであまり有効でない。この論文ではマイクロプログラムに適した初等関数の計算法として、CORDIC、STL、PDMについて簡単に説明し、これらの計算法の一般化、評価について論じる。

§2. CORDIC

CORDICは Coordinate Rotation DIgital Computer の略であり、1956年 Volder が発表した計算法である[1]。この計算法の特徴は多くの初等関数を統一された計算法で生成する点にある。$\overrightarrow{OP}=(x,y)$ に対してR, Aを次のように定義する。

$R=\sqrt{x^2+my^2}$, $A=m^{-\frac{1}{2}}\tan^{-1}(\sqrt{m}\,y/x)$

$\overrightarrow{OP_i}=(x_i, y_i)$ から $\overrightarrow{OP_{i+1}}=(x_{i+1}, y_{i+1})$ を次のように作る。

$x_{i+1}=x_i+my_i\delta_i$, $y_{i+1}=y_i-x_i\delta_i$ (1)

但し $\alpha_i=m^{-\frac{1}{2}}\tan^{-1}(\sqrt{m}\,\delta_i)$

これは $m=1$ のとき右図のような $\overrightarrow{OP_i}$ の回転を意味する。

また，回転角度を保持するために z という変数を導入する。

$z_{i+1}=z_i+\alpha_i$ (2)

このとき $A_{i+1}=A_i-\alpha_i$, $R_{i+1}=R_i\sqrt{1+m\delta_i^2}\equiv R_i\cdot K_i$ となる。

この計算を $i=0$ より n 回くり返すと次の式が得られる。

$R_n=R_0\cdot K$ (但し $K=\prod_{i=0}^{n-1}K_i$), $A_n=A_0-\alpha$ (但し $\alpha=\sum_{i=0}^{n-1}\alpha_i$)

$x_n=K\{x_0\cos(\sqrt{m}\alpha)+y_0\sqrt{m}\sin(\sqrt{m}\alpha)\}$

$y_n=K\{y_0\cos(\sqrt{m}\alpha)-\frac{x_0}{\sqrt{m}}\sin(\sqrt{m}\alpha)\}$

$z_n=z_0+\alpha$

$\delta_i=2^{-F_i}$ (F_i：整数)となるように α_i を決めると各段階での計算(1)，(2)は加減算，シフト，データ転送のみでおこなえる。計算モードとして $y_n\to 0$ となるように計算するVモードと，$z_n\to 0$ となるようにするRモードを考え，初期値 x_0，y_0，z_0 と m の値を決めると次の初等関数が生成される。

	Rモード	Vモード
$m=1$	sin, cos	$\tan^{-1}$
$m=0$	乗算	除算
$m=-1$	sinh, cosh	$\tanh^{-1}$, $\sqrt{\ }$

§3. STL

適当な定数を用いて表を作り、それを逐次参照して近似する方法であり、1965年 Specker [4] が発表した。この方法は $\ln$, exp に適しているが、sin, cos, $\tan^{-1}$ も生成でき、sin, cos, $\tan^{-1}$ に関してはCORDICと同じ方法になる。ここでは $\ln$ についてのみ説明しよう。

$$\ln x = \ln\left[\prod_{i=1}^{n} a_i x\right] - \sum_{i=1}^{n} \ln a_i \qquad (3)$$

であるから、与えられた x に対し $\prod_{i=1}^{n} a_i x \fallingdotseq 1$ となるように定数 a_i を選ぶ。すなわち

$$\begin{cases} a_i = 1 + 2^{-i} & : \text{if} \quad (1+2^{-i}) \prod_{k=0}^{i-1} a_k x < 1 \\ a_i = 1 & : \text{otherwise} \end{cases}$$

$\prod_{i=1}^{n} a_i x \fallingdotseq 1$ の時、$\ln x$ は(3)式より次のように近似できる。

$$\ln x \fallingdotseq \prod_{i=1}^{n} a_i x - 1 - \Sigma \ln a_i, \quad \text{error} < \frac{1}{2}\left(\prod_{i=1}^{n} a_i x - 1\right)^2$$

この計算は加減算、シフト、データ転送のみでおこなえる。

§4. PDM

計算機で乗除算は加減算とシフトをくり返して実行される。この方法を応用して初等関数を生成する計算法が Meggitt により1962年に発表された[5]。これは Pseudo Division and pseudo Multiplication process と呼ばれ、略してPDMとする。PDMでは $\ln$, $\tan^{-1}$, $\sqrt{\ }$, exp, tan が生成できる。ここでは $\tan^{-1}$ の場合についてのみ説明しよう。

$$(x+iy)\prod_{j=1}^{n}(1-i2^{-j})^{q_j}=R \quad (\text{real}) \qquad (4)$$

となるように q_j を選ぶ。但し q_j は 0 or 1 とする。両辺の対数をとると

$$\ln(x+iy)=\ln R-\sum_{j=1}^{n}q_j\ln(1-i2^{-j})$$

この両辺の虚数部をとると

$$\tan^{-1}(y/x)=\sum_{j=1}^{n}q_j\tan^{-1}2^{-j} \qquad (5)$$

となる。従って与えられた x, y に対して (4) 式をみたすように q_j を選び、その q_j を用いて (5) 式より $\tan^{-1}(y/x)$ を計算すればよい。q_j を求めるのを才1段階、$\tan^{-1}(y/x)$ を計算するのを才2段階としよう。q_{j-1} まで求まっている時 q_j の求め方を以下に示す。

$x_j+iy_j=(x+iy)\prod_{k=1}^{j-1}(1-i2^{-k})^{q_k}$ とおく。

(4) 式より y_j が 0 に近づくように q_j を決めればよい。

$q_j=0$ ならば　$x_{j+1}+iy_{j+1}=x_j+iy_j$

$q_j=1$ ならば　$x_{j+1}+iy_{j+1}=(x_j+iy_j)(1-i2^{-j})$

$$=(x_j+y_j\cdot 2^{-j})+i(y_j-x_j\cdot 2^{-j})$$

従って $w_j=y_j-x_j\cdot 2^{-j}$ とおき、$w_j\geq 0$ ならば $q_j=1$, $w_j<0$ ならば $q_j=0$ としてこれをくり返せば、y_n は 0 に近づく。

この才1段階の計算は除算に似ているので Pseudo Division process、才2段階は乗算に似ているので Pseudo Multiplication process と呼ぶ。

§5. 計算法の一般化

以上に述べたCORDIC, STL, PDMは定数を用い, 加減算とシフトをくり返して初等関数を生成している点で類似している。ここでは別の見方からこれらの3方法を一般化する。

初等関数を次の2つの型に分類する。

乗算型	除算型
乗算	除算, 累乗根
exp	ln
sin, cos, tan	$\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$
sinh, cosh, tanh	$\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$

CORDICのRモードは乗算型, Vモードは除算型に対応する。次にこの2つの型の初等関数の性質と近似法について述べる。

(1) 乗算型

乗算型の初等関数$f(x)$の性質として、$f(a+b)$が$f(a)$と$f(b)$を用いて表わせる。

Ex. $$\exp(a+b) = \exp a \cdot \exp b$$

$$\sin(a+b) = \sin a \cdot \sqrt{1-\sin^2 b} + \sqrt{1-\sin^2 a} \cdot \sin b$$

この性質を用いて乗算型の初等関数が近似できる。$x_{i+1} = x_i + \alpha_i$とすると$f(x_{i+1})$が$f(x_i)$と$f(\alpha_i)$を用いて計算できる。$i=0$からn回これをくり返すと$f(x_0 + \sum_{i=0}^{n-1} \alpha_i)$が$f(x_0)$と$f(\alpha_i)$を用いて計算できる。そこで$x_0 = 0$とし, $x \fallingdotseq \Sigma \alpha_i$となるように定数$\alpha_i$を決めると$f(x)$が計算できる。定数$\alpha_i$は計算が簡単になるように選ぶ。以下, 各関数別に述べる。

乗算の場合 $\ln(x_i+\alpha_i)=\ln x_i+\ln\alpha_i$ であり、これをくり返すと $\ln(\Sigma\alpha_i)=\Sigma\ln\alpha_i$ であるので $\alpha_i=2^{-i}$ or 0 とすれば加算とシフトのみで実行できる。これは乗数を2のベキ乗の和に展開する普通の乗算方法である。

exp の場合 $\exp(\Sigma\alpha_i)=\Pi\exp\alpha_i$ であるから $\exp\alpha_i=1+2^{-i}$ or 1, すなわち $\alpha_i=\ln(1+2^{-i})$ or 0 とすると

$$\exp(x_i+\alpha_i)=\exp x_i+2^{-i}\cdot\exp x_i \qquad \text{if } \alpha_i=\ln(1+2^{-i})$$
$$=\exp x_i \qquad \text{if } \alpha_i=0$$

となり、加算とシフトのみで実行できる。これはSTLの近似法と同じである。

sin, cos の場合は同時に求めることができる。この場合、$\sin\alpha_i$, $\cos\alpha_i$ の両方の値を簡単にするため $\tan\alpha_i=\pm2^{-i}$、すなわち $\alpha_i=\pm\tan^{-1}2^{-i}$ と選ぶと、$\sin\alpha_i=\frac{\pm2^{-i}}{\sqrt{1+2^{-2i}}}$, $\cos\alpha_i=\frac{1}{\sqrt{1+2^{-2i}}}$ となり、

$$\sin(x_i+\alpha_i)=\frac{1}{\sqrt{1+2^{-2i}}}(\sin x_i\pm2^{-i}\cos x_i)$$
$$\cos(x_i+\alpha_i)=\frac{1}{\sqrt{1+2^{-2i}}}(\cos x_i\mp2^{-i}\sin x_i)$$

$\frac{1}{\sqrt{1+2^{-2i}}}$ は α_i の正負によらないので $\prod_{i=1}^{n-1}\sqrt{1+2^{-2i}}=K$ は定数となる。従って $x\fallingdotseq\Sigma\alpha_i$ と近似することにより $K\sin x$ と $K\cos x$ が加減算とシフトを用いて同時に求められる。この方法はCORDIC、STLと同じである。

tan の場合は上記のようにsin, cos が同時に求まるので除算をおこなえばよい。この時Kは定数でなくてよいので、

$\alpha_i = \tan^{-1} 2^{-i}$ or 0とすれば、これはPDMと同じになる。

sinh, cosh, tanh も同様にすればよい。

(2) 除算型

除算型の初等関数$f(x)$の性質として$f(a)+f(b)=f(c)$とする時、cがaとbを用いて表わせる。但し除算、累乗根を除く。

Ex. $\ln a + \ln b = \ln a \cdot b$

$$\tan^{-1} a + \tan^{-1} b = \tan^{-1} \frac{a+b}{1-ab}$$

この性質を利用して$f(x)$が近似できる。$f(x_i)+f(\alpha_i)=f(x_{i+1})$としてこれを$i=0$からn回くり返すと$f(x_0)+\sum_{i=0}^{n-1} f(\alpha_i) = f(x_n)$となり、$x_n$が$x_0$と$\alpha_i$を用いて計算できる。そこで$f(x_n)\to 0$となるように定数$\alpha_i$を選べば$f(x) = -\sum_{i=0}^{n-1} f(\alpha_i)$で近似することができる。以下、各関数別に述べる。

$\ln$の場合 $\ln x + \sum_i \ln \alpha_i = \ln (\prod_i \alpha_i x)$ であるから、$\prod_i \alpha_i x \to 1$ となるように定数α_iを選べばよい。$\alpha_i = 1+2^{-i}$ or 1とすると $x_{i+1} = x_i + 2^{-i}\cdot x_i$ or x_i となり、加算とシフトをくり返すことにより $\ln x = -\Sigma \ln \alpha_i$ で求められる。これはSTLと同じである。

$\tan^{-1}$の場合、$\tan^{-1} x_i + \tan^{-1} \alpha_i = \tan^{-1} \frac{x_i+\alpha_i}{1-x_i\alpha_i}$ であるのでこのまま計算するとくり返し毎に除算が必要となる。そこで$x_i = \frac{v_i}{u_i}$とすると $x_{i+1} = \frac{v_{i+1}}{u_{i+1}} = \frac{\frac{v_i}{u_i}+\alpha_i}{1-\frac{v_i}{u_i}\alpha_i} = \frac{v_i+u_i\alpha_i}{u_i - v_i\alpha_i}$ となり、

$v_{i+1} = v_i + u_i\alpha_i$, $u_{i+1} = u_i - v_i\alpha_i$ となる。従って $\alpha_i = 2^{-i}$ or 0

とおくと (u_i, v_i) から (u_{i+1}, v_{i+1}) が加減算とシフトだけで計算できる。そして $f(x_n) \to 0$ するには $v_n \to 0$ とすればよい。これはPDMと同じであり、$\alpha_i = \pm 2^{-i}$ とすればCORDIC, STLと同じである。

$\sin^{-1}$ の場合 $\sin^{-1}x_i + \sin^{-1}\alpha_i = \sin^{-1}(x_i\sqrt{1-\alpha_i^2} + \alpha_i\sqrt{1-x_i^2})$ であるので $\alpha_i = \frac{\pm 2^{-i}}{\sqrt{1+2^{-2i}}}$ とすると $\sqrt{1-\alpha_i^2} = \frac{1}{\sqrt{1+2^{-2i}}}$ となり

$x_{i+1} = \frac{1}{\sqrt{1+2^{-2i}}}(x_i \pm 2^{-i}\sqrt{1-x_i^2})$ となる。更に $\sqrt{1-x_i^2} = y_i$ とおくと

$$x_{i+1} = \frac{1}{\sqrt{1+2^{-2i}}}(x_i \pm 2^{-i}y_i)$$

$$y_{i+1} = \frac{1}{\sqrt{1+2^{-2i}}}(y_i \mp 2^{-i}x_i)$$

となり、$x_n \to 0$ とすることにより $\sin^{-1}x = -\sum_i \sin^{-1}\frac{\pm 2^{-i}}{\sqrt{1+2^{-2i}}}$
$= \mp \sum_i \tan^{-1}2^{-i}$ で求められる。

$\tanh^{-1}$, $\cos^{-1}$, $\sinh^{-1}$, $\cosh^{-1}$ も同様にできる。

§6. 評価

ここではCORDIC, STL, PDMの各方法で生成される初等関数の平均実行時間を比較する。まずHITAC 8350での実験結果を表1に示す。

Algorithm	Function	Main	Data	Shift	計
CORDIC	sin,cos	225	51	334	610
	$\tan^{-1}$	250	51	334	635
	sinh,cosh	243	59	360	662
	$\tanh^{-1}$	270	59	360	689
STL	ln	107	22	36	165
	exp	199	59	60	318
PDM	ln	208	38	63	309
	$\tan^{-1}$	231	34	44	309
	tan	306	61	76	510

表 1

Algorithm		Main	Data	Shift	計
CORDIC	(m=1)	100	25	50	175
	(m=-1)	104	26	52	182
STL	(ln)	44	11	11	66
	(exp)	69	23	12	104
PDM	(ln)	96	12	12	120
	(tan⁻¹)	96	12	24	132
	(tan)	163	25	25	213

表　2

またHITAC 8350には、多重シフトができない、定数が8bitしか生成できない、任意のレジスタ間の演算ができない等の点のため実行時間が遅くなっているので、これらの点を改良した仮想計算機を想定して平均実行時間を求めると表2のようになる。表1と表2を比べると明らかに表2の方が2~4倍高速になっている。結局マイクロプログラム化の効果はハードウェアにかなり依存するものと思われる。

なお最後に御指導、御協力いただいた渡辺勝正助教授に深く感謝致します。

[参考文献]

[1]J.E.Volder; TheCORDIC Trigonometric Computing Technique; IRE Trans.on E.C. 1959 pp330-334
[2]J.S.Walther; A unified algorithm for elementary functions; s.J.C.C. 1971 pp379-385
[3]M.D.Perle; CORDIC Technique Reduces Trigonometric Function Look-Up; Computer design 1971 pp72-78
[4]W.H.Specker; A Class of Algorithms for lnx,expx,sinx,cosx, tanx and cotx; iEEE Trans.EC 1965 pp85-86
[5]J.E.Meggitt;Pseudo division and pseudo multiplication processes; IBM J.Res.Develop.1962 pp210-226
[6] 小柳　滋「マイクロプログラムによる初等関数の計算法とその評価」京大工学部数理工学科 修士論文 1974