5L-1 初等関数の精度低下防止法

井阪 秀高 (神戸日本電気ソフトウェア㈱)

1．はじめに

初等関数の計算において、多倍長演算を使わずに、できるだけ精度を保って計算する注意事項をまとめる。さらに、対数ガンマ関数、べき乗関数、$\sqrt{(x^2+y^2)}$ 等については、精度を保つ計算方法を提案する。

2．基本的方法

精度を保つ基本的考え方をまとめると次のようになる。
①近似解yに微小量Δyを加算して解を更新する。
②途中でビット落ちした値(目標:1/16~0.5)を経由しない。
③減算で近似区間内に還元するときは、入力引数から正確な上位ビット分の減算した後、下位ビット分の減算をする。
④x≠0でf(x)=0になる関数はxを中心に最良近似式を作る。
⑤解の指数部，仮数部別々に計算し、最後に合成する。

また、各関数計算時に使う近似式に対し、16進浮動小数点内部表現計算として共通な方法を以下にまとめる。
⑴多項式の場合
x*(1+(y*(a_1+…)‥)は、x+{x*y*(a_1+…)‥)}として計算。
x*{a_0+(y*(a_1+…)‥}で{}内の仮数部がビット落ちするときは、各係数を定数で割り{}内を0.5~1.0になるようにしてから多項式計算をし、最後にその定数を乗じる。
⑵x＊｛(多項式Ⅰ)／(多項式Ⅱ)｝の場合
{}内の仮数部がビット落ちするときは、{(多項式Ⅰ)*x}/(多項式Ⅱ)と計算するか、近似区間で解がxに近いときx+x*(多項式Ⅲ)/(多項式Ⅱ)と式を変形して計算するか、|x|<0.5のとき (x+x)*[(多項式Ⅰ)/{2*(多項式Ⅱ)}]と計算する。但し、このとき多項式Ⅰ、多項式Ⅱ、多項式Ⅲともxのとりうる値の範囲内で仮数部ができるだけビット落ちしないように多項式係数に定数を乗じて近似式を作る。

3．三角関数 (sin(x),cos(x),tan(x))

⑴ y=|x|＊2／πとする。
ｓｉｎ(ｘ)：n=｛(yの整数部)＋1｝∧'F‥FE'
ｃｏｓ(ｘ)：n=(yの整数部)∨'0‥01'
ここで、(y－n)で[-1,1]区間に還元したり(|x|－n＊π／2)で[-π/2,π/2]区間に還元すると精度が落ちるので、次のように分割減算し[-π/2,π/2]区間に還元する。
r=(|x|－n＊a)－n＊b，cos(x)計算時：r=－r
n：整数部，r：小数部＊π／2
a：π／2の仮数部上位半ビット分で下位半ビットを0にしたもの
b：π／2－a (aの下位半ビット以下のビット列)
ｓｉｎ(ｒ)を計算し、nの最下位から2ビット目が1のとき正負逆転する。
ｓｉｎはxが負のときさらに正負逆転する。
⑵ y=|x|＊4／πとする。
このときも⑴と同様に[-π/4,π/4]区間に還元する。
n=｛(yの整数部)＋1｝∧'F‥FE'
r=(|x|－n＊a)－n＊b
n：整数部，r：小数部＊π／4
a：π／4の仮数部上位半ビット分で下位半ビットを0にしたもの
b：π／4－a (aの下位半ビット以下のビット列)
nの最下位から2ビット目をi
nの最下位から3ビット目をjとする。
ｓｉｎ(ｘ)
i=0：ｓｉｎ(ｒ)、1：ｃｏｓ(ｒ) を計算
j≠(xの符号ビット)：正負逆
ｃｏｓ(ｘ)
i=0：ｃｏｓ(ｒ)、1：ｓｉｎ(ｒ) を計算
i≠j：正負逆
ｔａｎ(ｘ)
i=0：ｔａｎ(ｒ)、1：－1／ｔａｎ(ｒ)を計算
x<0：正負逆

4．x≠0でf(x)=0になる関数

⑴ ｌｏｇ(Γ(ｘ))
x≒1，2でf(x)≒0になり桁落ちしやすい。
x≒1のときy=x－1として
$\log(\Gamma(x)) \fallingdotseq -\gamma * y + \sum_{n=2}^{\infty}\{\zeta(n)*(-y)^n/n\}$
x≒2のときy=x－2として
$\log(\Gamma(x)) \fallingdotseq (1-\gamma)*y + \sum_{n=2}^{\infty}\{(\zeta(n)-1)*(-y)^n/n\}$
を最良近似式化した多項式で求める。
(γ：Euler定数，ζ(n)：Riemannのゼータ関数)
⑵ ａｃｏｓｈ(ｘ)
x≒1 (x≧1)のときf(x)≒0になり桁落ちしやすい。
$z=\sqrt{(x-1)}$，Y=－(x－1)として
$\mathrm{acosh}(x) \fallingdotseq z + z*[(\sqrt{2}-1)+\sqrt{2}*\sum_{n=1}^{\infty}[(2n-1)!!*(Y)^n/\{(2n)!!*(2n+1)*2^n\}]]$
を最良近似式化した多項式で求める。
⑶ ａｃｏｓ(ｘ)
x≒1 (x≦1)のときf(x)≒0になり桁落ちしやすい。

The Algolithms for Keeping Precision of Elementary Functions

Hidetaka Isaka
NEC Software Kobe,Ltd.

$z=\sqrt{(1-x)}$，$Y=1-x$として
$\mathrm{acos}(x)\fallingdotseq z+z*[(\sqrt{2}-1)+\sqrt{2}*\sum_{n=1}^{\infty}[(2n-1)!!*(Y)^n/\{(2n)!!*(2n+1)*2^n\}]]$
を最良近似式化した多項式で求める。

⑷ $\log_2(x)$

$x\fallingdotseq 1$のときｆ（ｘ）$\fallingdotseq 0$になり桁落ちしやすい。
$1/\sqrt{2}<x<\sqrt{2}$のとき$y=(x-1)/(x+1)$として
ｙに対する最良近似式ｆ（ｙ）を用いる。
または、$x=>2^e*m,(0.5\leqq m<1)$なるｍを取り出し
$m<1/\sqrt{2}$のとき：$b=1$，$z=2*m$
$m\geqq 1/\sqrt{2}$のとき：$b=0$，$z=m$
$y=(z-1)/(z+1)$として
$\log_2(x)=(e-b)+f(y)$

5．公式どうり計算すると桁落ちする

⑴ x^r

ｅｘｐ（ｙ＊ｌｏｇ（ｘ））と計算するとｘが大きいとき桁落ちするので、解の指数部と仮数部を別々に計算する。
ｐ進内部表現とする。
$x=>p^e*m$　（ｅ:指数部，ｍ:仮数部$(1/p\leqq m<1)$,を分離）
if（$m<1/\sqrt{p}$）then　$m=m*p$，$e=e-1$
$y=>i+d$　（ｉ:整数部，ｄ:小数部，を分離）
$g=\log_p(m)*y+e*d$
$g=>j+b$　（ｊ:整数部，ｂ:小数部$(-1\leqq b<0)$，を分離）
$x^r=p^{(e*i+j)}*p^b$（ｅ＊ｉ＋ｊ:指数部，p^b:仮数部）

⑵ $\sqrt{(x^2+y^2)}$

x^2，y^2をそのまま計算するとオーバフロしたり精度が低下することがある。
$|y|\leqq|x|$として$|x|/n*\sqrt{\{n^2+n^2*(y/x)^2\}}$，$(n=1,0.5,\cdots)$としたりｘの指数部の値をｘ，ｙの指数部から減じて$\sqrt{(x^2+y^2)}$を計算し，減じた値を解の指数部に加算するのも精度が低下することがある。そこで、近似解$|x|$に微小量Δxを加えて解を更新する。
$|y|\leqq|x|$とする。
$r=(y/x)^2$
$t=\overset{\infty}{\Phi}\left\lfloor\frac{r}{2}\right\rfloor$，（$0\leqq r\leqq 1$）を最良近似式化した
$t=r*f(r)/g(r)$によりｔを計算する。
$\sqrt{(x^2+y^2)}\fallingdotseq|x|+|x|*t$
また、精度を上げるためには、
$\sqrt{(x^2+y^2)}\fallingdotseq|x|+|x|*(t*t+r)/(t+t+2)$とする。
応用として$\sqrt{x}$，$(0.5\leqq x\leqq 1)$は、$r=2*(x-0.5)$としてｔを求め、$p=1/\sqrt{2}$，$\sqrt{x}=p+p*t$とする。
$\mathrm{asinh}(x),(x\geqq 1)$は、$r=1/x^2$としてｔを求め、
$\mathrm{asinh}(x)=\log(2x+x*t)$，$(x*r=1/x)$とする。

⑶ $\sqrt{(x^2-y^2)}$，（$|y|\leqq|x|$）

$\sqrt{(x^2+y^2)}$と同様に$r=(y/x)^2$とし
$t=\overset{\infty}{\Phi}\left\lfloor\frac{-r}{2}\right\rfloor$，$(0\leqq r\leqq a,a<1)$を最良近似式化した
$t=r*h(r)/i(r)$によりｔを計算する。
$\sqrt{(x^2-y^2)}\fallingdotseq z=|x|+|x|*t$
また、精度を上げるためには、
$\sqrt{(x^2-y^2)}\fallingdotseq|x|+|x|*(t*t-r)/(t+t+2)$とする
$\mathrm{acosh}(x),(x\geqq 1/\sqrt{a})$は、$\mathrm{asinh}(x)$と同様に計算できる

6．その他の関数

⑴ e^x-1

$x\fallingdotseq 0$で桁落ちするので、最良近似式とデータテーブルを利用する。また、近似区間への還元値（ｚ）を得るときも精度を保つため上位、下位ビット分けて減算する。
$y=x*\log_2(e)$
$|y|<1$のとき
$|y|$の小数点以下ｎビットを取り出しｉとする。
$i=(i+1)/2$（整数演算），$x<0$のとき$i=-i$
$p=i/2^{n-1}$，$z=(x-p*a)-p*b$
$e^x-1=u(i)+t(i)*(e^z-1)$
ａ：$\log_e(2)$の仮数部上位半ビット分で下位半ビットを０にしたもの
ｂ：$\log_e(2)$-ａ（ａの下位半ビット以下のビット列）
ｕ（ｉ）：$2^{i/(2**(n-1))}-1$のデータテーブル
ｔ（ｉ）：$2^{i/(2**(n-1))}$　のデータテーブル
e^z-1　：ｚに対する最良近似式の値（$|z|\leqq 1/2^n$）
$|y|\geqq 1$のとき
$y=>n+d$（ｎ：整数部，ｄ：小数部$(-1<d\leqq 0)$,を分離）
$|d|$の小数点下ｎビットをｉとする。
$p=n-i/2^n$，$z=(x-p*a)-p*b$
$e^x-1=2^n*(v(i)*e^z)-1$　(n:整数部,()内:仮数部)
ｖ（ｉ）：$2^{-1/(2**n)}$のデータテーブル
e^z　：ｚに対する最良近似式の値（$-1/2^n<z\leqq 0$）

⑵ ａｔａｎ（ｘ）

$x\geqq 0$の範囲で計算し、$x<0$のとき解を負にする。
$x\geqq\tan((2n-1)*\pi/4n)$（ｎ：区間数－１）のとき
$\mathrm{atan}(x)=\pi/2-\mathrm{atan}(1/x)$
ｅｌｓｅ
$\tan((2i-1)*\pi/4n)<x\leqq\tan((2i+1)*\pi/4n)$
$(i=0,\cdots,n-1)$，なるｉを見つける。
$\mathrm{atan}(x)=i*\pi/2n+\mathrm{atan}\{(x-s)/(s*x+1)\}$
ｓ：$\tan(i*\pi/2n)$
ここで、１６進内部表現のとき$\{(2i-1)*\pi/4n<1\}\wedge\{i*\pi/2n\geqq 1\}$となる区間ができるとそこの精度が低く、このようにならないｎを決める$(n=2,5,7,8,10,11,13\cdots)$。

7．あとがき

初等関数を効率よく、精度低下を防ぎながら計算する方法をまとめた。これらの算法は、ＮＥＣのＳＸシリーズ等で既に実現し、これらにより精度向上がなされたことを確認している。

最後に、組込み関数実現に関し色々ご協力くださったＮＥＣ基本ソフトウェア開発本部の片山部長代理、鈴木様に感謝します。

8．参考文献

山内，宇野，一松：電子計算機のための数値計算法Ⅲ，培風館，(1972)
井阪他：ベクトル計算機に適した三角関数計算法，情報処理学会第34回全国大会，(1987)