ガンマ関数のテイラー展開係数の計算

平　山　　　弘*

Calculation of Coefficients for Taylor Expansion of Gamma Function

Hiroshi HIRAYAMA

Abstract

The numerical calculation of coefficents for theTaylor expansion of Gamma function can be carried out easily by using the multiple-precision floating number operations. It is easy to estimate the errors of the coefficients by two different precision calculations. This calculation method is very effective if the sum of computing time and developing time for program is considered. For this calculation, it is found that the former results have some errors probably arise in roundoff.

1. はじめに

ある関数のテイラー（Taylor）展開の係数は，その関数の高階微分が解析的な方法で容易に求められるならば，数式処理などを使うことによって，容易に求めることができる。この性質を利用しているものが，高速自動微分法である。この方法は，単純な基本的な関数の集まりで表現できるものであるならば，非常に有効であるが，簡単に表現出来ない関数では，殆ど利用できない。

しかしながら，その高階微分係数が簡単に求められない場合，その展開係数は数値的方法を使って求めなければならない。このような場合，数値微分法を利用しなければならない。

数値微分法を利用して展開係数を求めることは，その計算が非常に桁落ちの激しい計算になるため，たかだか40桁程度の計算しか出来ない通常の電子計算機では，高次の展開係数を求めることはほとんど不可能である。このため，通常の数値計算では，高次の数値微分を避けるようにしている。Taylor展開の係数を計算するような場合，数値微分法を直接適用すれば，大きな桁落ちが生じ，意味のある計算ができない。このような場合，桁落ちを避けるためにその関数の解析的な性質を利用した特別な計算ルーチンを準備しなければならない。

Taylor展開の係数を求めるような計算は，一度計算すれば何回も計算する必要はないから，これだけのために特別のアルゴリズムを工夫しプログラムを作ることは，理論的に興味はあっても，実用的ではない。

本文では，単純に数値微分法を使いテイラー展開係数を求める。具体例としてgamma関数の逆数のTaylor展開の係数を計算する。桁落ちに対しては多倍長の演算を使うことによって避け，有効桁数を確保する。計算精度を変えることによってこのような計算では非常に重要な問題となる計算結果の検証が容易にできるという特徴をもっている。

この計算方法は，計算量が多くなるので従来の考え方では，あまり良い計算方法とはいえない。しかし，この計算のアルゴリズムは非常に単純であるため，プログラム作成は非常に容易である。アルゴリズムの検討を含むプログラム作成時間と計算実行時間の合計を考えるとき，この計算方法は，結果を得るために余り時間を必要としないという意味で非常に効率的な計算方法である。低コストの計算方法ともいえる。従来，計算機は非常に高価であり，また，一度作成されたプロ

平成3年9月30日受理
* 機械システム工学科

グラムは何回も実行されると仮定して，そのプログラムの善し悪しを判定してきた。しかしながら，近年の半導体技術の発展は，計算機を非常に低価格に変えたため，人間に負荷が多くなるような計算方法は，コスト上昇を招くのであまり良い計算方法とはいえない場合が非常に多くなってきている。

このように考えるとき，多倍長演算を使う計算方法は，このような問題に対して非常に有効であることがわかる。この計算によって，数学公式集[1)]の誤りを指摘することができた。

ベクトル計算機では，多倍長数の積は，非常に効率的に計算出来るので，ベクトル計算機が，将来容易に使えるならば，さらに，効果的な方法となると期待できる。

2. ガンマ関数の計算方法

ガンマ関数 $\Gamma(x)$ は，積分表示を使って

$$\Gamma(x)=\int_0^\infty t^{x-1}e^{-t}dt \qquad (1)$$

と定義できる。ガンマ関数の計算は数値計算ライブラリでは，漸近展開公式がよく使われる。この計算方法は，精度が高くなるにつれて非常に大きい数値で漸近展開式を利用しなければならないし，また，高次の漸近展開公式を使わなければならない。非常に大きな数値を使った場合，その関数値を計算しようとしている値に戻さなければならない。このために，漸化式

$$\Gamma(x+1)=x\Gamma(x) \qquad (2)$$

が利用されるが，この計算は，x が大きくなるとこの計算量は無視できないほど多くなる。高次の漸近展開式は，高精度の計算を行うための方法としては，非常に効率的な方法であるが，高次の漸近展開の係数を前もって計算しておく必要があり，また漸近展開の係数の計算に時間がかかるため，漸近展開を使う計算方法は高精度計算には通常使われない。

ここでは，(1) の積分を $t=N$ で打ち切る方法[3)]を利用する。すなわち，(1) の積分を

$$\int_0^\infty t^{x-1}e^{-t}dt=\int_0^N t^{x-1}e^{-t}dt+\int_N^\infty t^{x-1}e^{-t}dt \quad (3)$$

のように分割する。N が十分に大きい場合，第二項はいくらでも小さくできるので第一項を高精度で評価することによって，ガンマ関数を高精度で計算することができる。$-\frac{1}{2}\leqq x\leqq\frac{1}{2}$ のとき，(3) の右辺の第二項は

$$\int_N^\infty t^{x-1}e^{-t}dt=e^{-N}\int_0^\infty (N+s)^{x-1}e^{-s}ds \qquad (4)$$

$$\leqq e^{-N}\int_0^\infty \frac{e^{-s}}{\sqrt{N+s}}ds \qquad (5)$$

$$\leqq e^{-N}\int_0^\infty \frac{e^{-s}}{\sqrt{N}}ds \qquad (6)$$

$$\leqq \frac{e^{-N}}{\sqrt{N}} \qquad (7)$$

と評価できる。例えば，$N=500$ のとき第二項は 3.2×10^{-219} となるので 200 桁以上の精度で計算することができることになる。

(3) の第一項は，e^{-x} のテイラー展開

$$e^{-x}=\sum_{k=0}^{\infty}\frac{(-1)^k x^k}{k!} \qquad (8)$$

を代入して，積分を実行すると

$$\int_0^N t^{x-1}e^{-t}dt=\sum_{k=0}^{\infty}\frac{(-1)^k N^{k+x}}{k!(k+x)} \qquad (9)$$

となる。この級数は，収束半径無限大の級数である。理論的には十分な項数を計算すればいくらでも高精度で計算できる。N が大きいとき，大きな桁落ちが生じるので，出来るだけ小さい N で計算しなければならない。このように N を選んだとき，計算精度は求めたい精度のおよそ 2 倍の精度で計算しなければらない。

この級数は x が有理数の場合，(9) の多倍長数での除算をなくすことができるので，計算する x の値が有理数の場合，高速に計算できる。数値微分では，微分係数を計算したい点付近の関数値が分かれば微分係数を計算することができるから，その付近の有理数における関数値だけを使って微分係数を計算することができる。

負の数に対するガンマ関数の値は，関係式

$$-x\Gamma(-x)\Gamma(x)=\frac{\pi}{\sin \pi x} \qquad (10)$$

から計算することができる。

3. テイラー展開係数の計算

テイラー展開係数は，展開級数 $f(x)$ を

$$f(x)=\sum_{k=0}^{\infty} a_n x^n \qquad (11)$$

と仮定し，a_n を指定した点を通るように連立方程式を解くことによって決定することができる。このようにすると，通過点は不等間隔であっても容易に展開係数を求めることができる。等間隔であれば，数値微分の公式が使い易いが，そのためには，高次の数値微分公式を準備しておかなければならない。

5次程度までならば公式集に載っているが，20次，30次となると殆ど載っていない。たとえ載っているとしても，その公式が間違いなくプログラムに記述するのは次数が高くなるにつれて難しくなる。また，本などに記述されている公式は次数が上がるにつれて誤っている可能性が大きくなる。このため公式の検証は必須である。このようなことを考慮すると，連立方程式を解く方法が最も無難な方法と思われる。

多項式が $(x_i,\ y_i)$ $(i=1, \ldots, n)$ の点を通るように決定するには，つぎの連立方程式を

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \ldots & x_2^n \\ 1 & x_3 & x_3^2 & x_3^3 & \ldots & x_3^n \\ & & \vdots & & & \\ 1 & x_n & x_n^2 & x_n^3 & \ldots & x_n^n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \quad (12)$$

を解き，(11) の式の係数を決定しなければならない。この行列は Vandermonde の行列と呼ばれるものである。この型の方程式は，高速に解く方法[2]が存在する。通常の連立一次方程式は，その元数を n とするとその方程式を解くには n^3 のオーダの計算時間が必要であるが，この方程式は，n^2 のオーダで解くことができる。

点列 $(x_i,\ y_i)$ $(i=1, \ldots, n)$ を通る $n-1$ 次の多項式は，Lagrange の多項式によって表現できる。$n-1$ 次の多項式 $P_j(x)$ を次のように定義する。

$$P_j(x)=\prod_{\substack{k=1\\(k\neq j)}}^{n} \frac{x-x_k}{x_j-x_k}=\sum_{k=1}^{n} A_{jk}x^{k-1} \quad (13)$$

この多項式は，$x=x_i$ のとき

$$P_j(x_i)=\delta_{ij}=\sum_{k=1}^{n} A_{jk}x_i^{k-1} \quad (14)$$

であることが分かる。このことに注意すれば，(12) の解である多項式は

$$Q(x)=\sum_{k=1}^{n} P_k(x)y_k \quad (15)$$

であることがわかる。式 (14) から，要素が A_{ij} で表される行列は，式 (12) の左辺の行列の転置行列の逆行列であることがわかる。(13) を展開すれば，逆行列が求められることになる。(13) を全ての j について定義通りに計算すれば，かなりの計算量になるが，まず次の式 $r(x)$ を展開する。

$$r(x)=\prod_{k=1}^{n} (x-x_k) \quad (16)$$

式 (16) を $(x-x_j)$ で割り，x に x_j を代入すれば，(14) 式が得られる。この計算は，n のオーダの計算だから，すべての j に対して計算しても n^2 のオーダの計算量で済むことになる。このように計算すれば非常に高速な計算が可能となる。

Vandermonde の行列は，悪条件の方程式として知られている。これは，数値微分の悪条件が形を変えたものと考えられる。数値微分では，k 階微分係数を求めるとその有効数字は $1/(k+1)$ になるので，このことを考慮して計算すべき精度を決定しなければならない。

多項式の通る点は十分に接近している様に選べば，どの様に選んでも，ガンマ関数に近づくから，同じ結果が得られる。もし，x_i として整数値を採ることが出来るならば，ガンマ関数の計算は，関数値が整数値になるので厳密な計算が可能であり，その計算は非常に簡単になると期待できる。この様な方法で計算したとき，式 (11) がガンマ関数に近づくかどうかは，次の誤差解析の評価式[4]から容易にわかる。近似する関数を $f(x)$，$(m+1)$ 次の Lagrange の多項式を $P_m(x)$ とすると，

$$f(z)-P_m(z)=\frac{1}{2\pi i}\oint_C \frac{\phi(z)f(\zeta)}{\phi(\zeta)(z-\zeta)}d\zeta \quad (17)$$

ここで

$$\phi(x)=\prod_{k=0}^{m} (x-x_k) \quad (18)$$

積分路 C は，全ての関数値が与えられている点 x_k を囲む閉曲線である。x_k の値は整数値をとると制限されているので，正負の整数値を使ったとしても積分路の半径は $m/2$ 程度の大きな値となる。積分路が十分に大きな円にとり，その半径を r とすると，z がゼロに近い領域では，式 (17) の ϕ の商の部分は，m が1増加するごとに，その誤差は $1/r$ の大きさになる。すなわち r が1より大きい場合，この部分は収束することになる。しかし，(17) の $f(z)$ が ϕ の商の部分が小さくなる以上に大きくなる場合，収束しないことになる。ガンマ関数の場合，x_k を整数値としたとき，この方法に

よって収束しないことは容易にわかる。

4. 計　算　結　果

計算は，つぎのような式のテイラー展開係数である。

$$\frac{1}{\Gamma(x)}=\sum_{k=1}^{50} a_k x^k \qquad (19)$$

$N=600$ として（9）の級数を計算した。このときの精度は 10 進でおよそ 262 桁で計算できる。桁落ちが生じるため，計算は 10 進で約 500 桁で行った。ガンマ関数の値は，2,000 分の 1 間隔で原点から正負両方向に 50 点を採り，その点でのガンマ関数の値を計算して，1/$\Gamma(x)$ の 100 次の展開係数まで求めた。この計算を確かめるために，同じ計算を 3,000 分の 1 間隔で計算し，その誤差を求めた。この結果の係数を付録 1 に示す。低次の展開係数は高精度で計算出来るが高次の展開係数になるにしたがって，誤差は大きくなる。50 次の係数でその誤差は 5.877×10^{-80} である。100 では，係数より誤差のほうが大きくなる。このため，ここでは，十分に精度が確保された 50 次までの係数を付録 1 に示した。

この計算と同様な結果は，文献[1]に記されている。この文献と同精度で表示したものが，表 1 である。この計算結果は，ここでの計算と一致しない部分がある。この部分をアンダーラインを引いて示した。たぶん，計算途中の丸め処理により発生した誤差による不一致だと思われる。

Table 1.　Series Expansion for $1/\Gamma(x)$ (16D)
Underlined figures show different parts from M. Abramowitz and A. Stegun[1].

n	a_n
1	1.00000 00000 000000
2	0.57721 56649 015329
3	−0.65587 80715 20253$\underline{9}$
4	−0.04200 26350 340952
5	0.16653 86113 822915
6	−0.04219 77345 555443
7	−0.00962 19715 278770
8	0.00721 89432 46663$\underline{1}$
9	−0.00116 51675 918591
10	−0.00021 52416 7411$\underline{50}$
11	0.00012 80502 82388$\underline{1}$
12	−0.00002 01348 54780$\underline{8}$
13	−0.00000 12504 934821
14	0.00000 11330 272320
15	−0.00000 02056 338417
16	0.00000 00061 16095$\underline{1}$
17	0.00000 00050 02007$\underline{6}$
18	−0.00000 00011 812746
19	0.00000 00001 043427
20	0.00000 00000 077823
21	−0.00000 00000 036968
22	0.00000 00000 005100
23	−0.00000 00000 000206
24	−0.00000 00000 00005$\underline{3}$
25	0.00000 00000 00001$\underline{2}$
26	−0.00000 00000 000001

5. 結　　　論

多倍長数による計算は，丸め誤差の影響を小さくできるため，数学的に実行可能なものであれば，計算方法をそのまま記述することによって，非常に条件の悪い問題でも容易に解くことができる。この方法は，丸め誤差の影響を考慮したアルゴリズムで計算する場合と比較したとき，計算効率は，かなり悪くなる。しかしながら，丸め誤差の影響を考慮したアルゴリズムを考えたり，さがしたりするにはかなりの時間がかかる。この時間と計算機で実行する時間の和が問題を解決するために必要な時間（費用）である。

計算機が，非常に高価であった時代には，計算機で計算している時間がほぼ問題の解決するための費用であったが，半導体技術が発展し計算機が非常に低価格になっている現在，最小の費用で問題を解決するためには，多倍長数による計算が有効である問題が，かなり多く分野に存在すると思われる。特に計算量の小さい問題では，多倍長数による計算は非常に有効である。

現在，計算機の価格は急激に低下しており，現在大きな問題であるとして，多倍長数による計算が行われていない分野でも，将来数多く行われるものと思われる。この傾向は，多倍長数計算を実行できるソフトウエアが多数販売されていることからも推定できる。

ここで扱ったガンマ関数の Taylor 展開係数の計算については，文献[5]に示されている。また，ガンマ関数の対数の Taylor 展開の展開係数がリーマンのゼータ関数を使って表現できることが知られているので，こ

れによって，対数ガンマ関数の展開係数からガンマ関数の展開係数を計算することもできる。いずれの場合でも，ここで示した方法と比較してかなり複雑である。この計算から，文献[5]の展開係数には誤りがあることを示すことができた。

6. 参　考　文　献

1）Milton Abramowitz and Irene A. Stegun: Handbook of Mathematical Functions, Dover (1965).

2）W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling: Numerical Recipes, Chap. 2, Cambridge University Press (1986).

3）R.P. Brent: A Fortran Multiple-Precision Arithmetic Package. ACM Trans. Math. Software, Vol. 4, No. 1 (March 1978) 57-70.

4）F. ジョン（藤田，名取訳）: 数値解析講義，第4章，産業図書（1971），

5）渡部，名取，小国: 数値計算ソフトウエア，第5章，丸善（1989）.

Appendix 1
Series Expansion for $1/\Gamma(x)$ (70D)

$$\frac{1}{\Gamma(x)} = \sum_{k=1}^{\infty} a_k x^k$$

n	a_n						
1	1.0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
2	0.5772156649	0153286060	6512090082	4024310421	5933593992	3598805767	2348848677
3	−0.6558780715	2025388107	7019515145	3904812797	6638047858	4347292362	4456838708
4	−0.0420026350	3409523552	9003934875	4298187113	9450040110	6093522065	8129761800
5	0.1665386113	8229148950	1700795102	1052357177	8150224717	4340570468	9031789938
6	−0.0421977345	5554433674	8208301289	1873913016	5268418982	2486376918	8732754590
7	−0.0096219715	2787697356	2114921672	3481989753	6294225211	3002105138	8626273116
8	0.0072189432	4666309954	2395010340	4465727099	0480088023	8318001094	7811736225
9	−0.0011651675	9185906511	2113971084	0183886668	0933379538	4057443407	5052756200
10	−0.0002152416	7411495097	2815729963	0536478064	7824192337	8338750350	2674890856
11	0.0001280502	8238811618	6153198626	3281643233	9489209969	3677214900	5458380412
12	−0.0000201348	5478078823	8655689391	4210218183	8229483329	7979115261	1626709082
13	−0.0000012504	9348214267	0657345359	4738330922	4232265562	1153959815	3499231574
14	0.0000011330	2723198169	5882374129	6203307449	4332400483	8621075654	2955053954
15	−0.0000002056	3384169776	0710345015	4130020572	8365125790	2629337945	3468317253
16	0.0000000061	1609510448	1415817862	4986828553	4286727586	5719712320	8673240292
17	0.0000000050	0200764446	9222930055	6650480599	9130304461	2742494481	7189533788
18	−0.0000000011	8127457048	7020144588	1265654365	0557773875	9504932587	5909618926
19	0.0000000001	0434267116	9110051049	1540332312	2501914007	0982312581	2121087107
20	0.0000000000	0778226343	9905071254	0499373113	6077722606	8086181392	9388194355
21	−0.0000000000	0369680561	8642205708	1878158780	8576623657	0963451360	9951364845
22	0.0000000000	0051003702	8745447597	9015481322	8632318027	2688606970	7632117350
23	−0.0000000000	0002058326	0535665067	8322242954	4855237419	7460910808	1014718805
24	−0.0000000000	0000534812	2539423017	9823700173	1872793994	8989715478	1206821116
25	0.0000000000	0000122677	8628238260	7901588938	4662242242	8165455750	4563213660
26	−0.0000000000	0000011812	5930169745	8769513764	5868422978	3121155729	1804847879
27	0.0000000000	0000000118	6692254751	6003325797	7724292867	4071088494	0796648271
28	0.0000000000	0000000141	2380655318	0317815558	0394756670	9037086350	7503345256
29	−0.0000000000	0000000022	9874568443	5370206592	4785806336	9926028450	5931419036
30	0.0000000000	0000000001	7144063219	2733743338	3963370267	2570668126	5606251743
31	0.0000000000	0000000000	0133735173	0493693114	8647813951	2226802287	5059471761
32	−0.0000000000	0000000000	0205423355	1766672789	3250253513	5573379668	2037935238
33	0.0000000000	0000000000	0027360300	4860799984	4831509904	3309820148	6531169583
34	−0.0000000000	0000000000	0001732356	4459105166	3905742845	1564779799	0697491087
35	−0.0000000000	0000000000	0000023606	1902449928	7287343450	7354275310	0792641355
36	0.0000000000	0000000000	0000018649	8294171729	4430718413	1618786668	9894586842
37	−0.0000000000	0000000000	0000002218	0956242071	9720439971	6913626860	3797317795
38	0.0000000000	0000000000	0000000129	7781974947	9936688244	1448633059	4165619499
39	0.0000000000	0000000000	0000000001	1806974749	6652840622	2745415509	9715185596
40	−0.0000000000	0000000000	0000000001	1245843492	7708809029	3654674261	4395121194
41	0.0000000000	0000000000	0000000000	1277085175	1408662039	9020667775	1124647748
42	−0.0000000000	0000000000	0000000000	0073914511	6961514082	3461289330	1085528237
43	0.0000000000	0000000000	0000000000	0000113475	0257554215	7609541652	5946930639
44	0.0000000000	0000000000	0000000000	0000463913	4641058722	0299448049	0795222846
45	−0.0000000000	0000000000	0000000000	0000053473	3681843919	8875077418	1967098933
46	0.0000000000	0000000000	0000000000	0000003207	9959236133	5262286123	7279082794
47	−0.0000000000	0000000000	0000000000	0000000044	4582973655	0756882101	5903521246
48	−0.0000000000	0000000000	0000000000	0000000013	1117451888	1988712901	0584943899
49	0.0000000000	0000000000	0000000000	0000000001	6470333525	4381388681	8259327906
50	−0.0000000000	0000000000	0000000000	0000000000	1056233178	5035812186	0056107153