(Version: 2021/6/26)

ガンマ関数の精度保証付き計算メモ

柏木 雅英

1　はじめに

ガンマ関数をどうやって計算してるか忘れそうなのでメモ。

2　ガンマ関数

(2013/8/8 のノートに従う)

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

を直接計算する方針。ただし、

$$\Gamma(x+1) = x\Gamma(x) \tag{2}$$

$$-x\Gamma(-x)\Gamma(x) = \frac{\pi}{\sin \pi x} \tag{3}$$

で変換できることも考慮する。

$1 \le x \le 2$ の範囲で (1) で計算し、(2) または (3) で変換する方針にする。

定数 $T \ge 1$ とし、(1) を

$$\Gamma(x) = \int_0^T t^{x-1} e^{-t} dt + \int_T^\infty t^{x-1} e^{-t} dt \tag{4}$$

と分解する。

(4) の後半は、次のように評価する。$T \ge 1$ なので $1 \le t$ であり、$0 \le x - 1 \le 1$ より

$$t^{x-1} \le t$$

が成立する。これにより、

$$\begin{aligned} \int_T^\infty t^{x-1} e^{-t} dt &\le \int_T^\infty t e^{-t} dt \\ &= \left[-e^{-t}(t+1) \right]_T^\infty \\ &= e^{-T}(T+1) \end{aligned}$$

と上から評価でき、また積分値は明らかに正なので、

$$\int_T^\infty t^{x-1}e^{-t}dt \in \left[0, e^{-T}(T+1)\right]$$

と区間評価出来る。

(4) の前半は、直接数値積分で評価する。数値積分は、ベキ級数展開 (Type-II PSA) に基づく方法で行う。ただし、原点で級数展開出来ないため、

$$\int_0^S t^{x-1}e^{-t}dt + \int_S^T t^{x-1}e^{-t}dt \tag{5}$$

のように分解し、後半は普通に積分する。前半は、t^{x-1} を除いた部分 e^{-t} を原点で Type-II PSA に展開し、それの各項に t^{x-1} を乗じたものの原始関数を求める方法で計算する。

パラメータは要調整だが、現在は S は自動決定、T は仮数部の bit 数 $\times \log 2$、PSA の次数 $= 18$ としている。

以上で $1 \le x \le 2$ の範囲のガンマ関数が計算できるようになった。それ以外の範囲については、(2) を用いて、$n = \lfloor x \rfloor - 1$ として、$x > 2$ に対しては

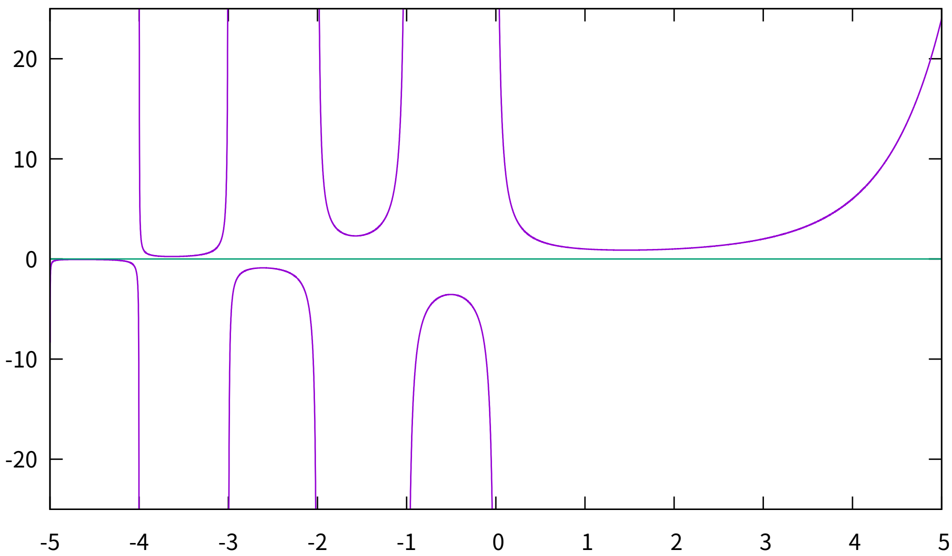
$$\Gamma(x) = \left(\prod_{i=1}^{n} x-i\right)\Gamma(x-n)$$

$x < 1$ に対しては

$$\Gamma(x) = \left(\prod_{i=n+1}^{0} \frac{1}{x-i}\right)\Gamma(x-n)$$

で計算できる。

大きな区間の入力に対しては、次のようにする。入力を $I = [\underline{I}, \overline{I}]$ とする。ガンマ関数の概形は図の通り。



よって、I が 0 または負の整数を含んでいる場合は、$[-\infty, \infty]$ を計算結果とすればよい。それ以外の場合は、ガンマ関数の極値点を含む場合は

$$\mathrm{hull}(\Gamma(\underline{I}), \Gamma(\overline{I}), \Gamma(\text{極値点}))$$

を、極値点を含まない場合は

$$\mathrm{hull}(\Gamma(\underline{I}), \Gamma(\overline{I}))$$

を結果とする。

以下、極値点の計算方法を述べる。ガンマ関数の極値点は、正の領域では

$$x_0 = 1.461632144968\ldots$$

負の領域では

$$\begin{aligned} x_1 &= -0.504083008\ldots \\ x_2 &= -1.573498473\ldots \\ x_3 &= -2.610720868\ldots \\ x_4 &= -3.635293366\ldots \\ &\vdots \end{aligned}$$

である。負の領域でのこの値の近似式

$$x_n \simeq -n + \frac{1}{\pi} \arctan\left(\frac{\pi}{\log n + \frac{1}{8n}}\right)$$

を初期値に用いて $\mathrm{digamma}(x) = 0$ に対する Newton 法を行い、高精度な極値点の近似値を得る。それを元に Krawczyk 法を用いて区間評価している。Krawczyk 法を行うには後述の精度保証付きの digamma 及び trigamma を使う。なお、極値点の計算には非常に時間を要するため、一度計算したらそれを覚えておいて再利用している。

3 ディガンマ関数

3.1 はじめに

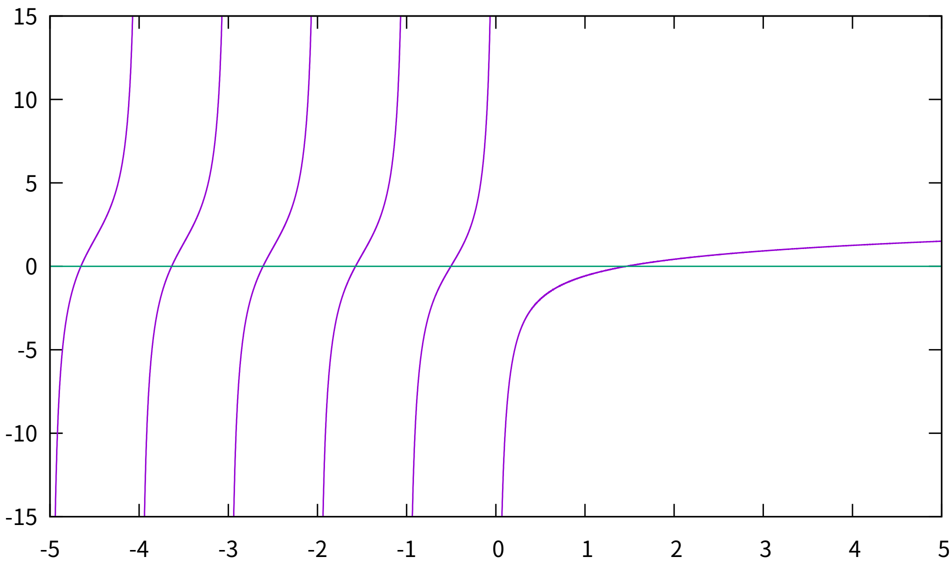
ディガンマ関数は

$$\mathrm{digamma}(x) = \frac{d}{dx} \log \Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)} \tag{6}$$

である。

2013 年頃雑に作ったが (第 3.2 節のアルゴリズム 0)、これだと x の絶対値が大きいときに精度が出ず計算時間もかかってしまうため、いろいろ改良を試みた。以下の節でそれらを説明する。区間幅の大きな区間 I に対する処理は共通で、幅 0 あるいは小さな幅の区間 x に対する計算方法をいくつか示す。

大きな区間の入力に対しては、次のようにする。入力を $I = [\underline{I}, \overline{I}]$ とする。ディガンマ関数の概形は図の通り。



よって、I が 0 または負の整数を含んでいる場合は、$[-\infty, \infty]$ を計算結果とすればよい。それ以外の場合は、単調増加なので単に $[\mathrm{digamma}(\underline{I}), \mathrm{digamma}(\overline{I})]$ を結果とする。

3.2　アルゴリズム 0

(2013/12/11 のノートに従う) x を点 (あるいはあまり幅の大きくない区間) とする。積分形

$$\mathrm{digamma}(x) = \int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1 - e^{-t}} \right) dt \tag{7}$$

を直接計算する方針。ただし、$1 \le x \le 2$ の範囲で (7) で計算することにし、それ以外の領域では、

$$\mathrm{digamma}(x+1) = \mathrm{digamma}(x) + \frac{1}{x} \tag{8}$$

で変換する。すなわち、$n = \lfloor x \rfloor - 1$ として、$x > 2$ に対しては

$$\mathrm{digamma}(x) = \mathrm{digamma}(x-n) + \sum_{i=1}^{n} \frac{1}{x-i}$$

$x < 1$ に対しては

$$\mathrm{digamma}(x) = \mathrm{digamma}(x-n) - \sum_{i=n+1}^{0} \frac{1}{x-i}$$

とする。

以下、$1 \le x \le 2$ に対して (7) を計算する方法を示す。定数 $T > 0$ とし、(7) を

$$\int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt = \int_0^T \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt + \int_T^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt \tag{9}$$

2 つに分ける。

(9) の後半は、次のように評価する。上界は、

$$\int_T^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt \le \int_T^\infty \frac{e^{-t}}{t} dt \le \int_T^\infty \frac{e^{-t}}{T} dt = \frac{1}{T} \left[-e^{-t} \right]_T^\infty = \frac{1}{T} e^{-T}$$

で、下界は、

$$\begin{aligned} \int_T^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt &\ge -\int_T^\infty \frac{e^{-xt}}{1-e^{-t}} dt \ge -\int_T^\infty \frac{e^{-xt}}{1-e^{-T}} dt \\ &= -\frac{1}{1-e^{-T}} \left[-\frac{1}{x} e^{-xt} \right]_T^\infty \\ &= -\frac{1}{1-e^{-T}} \frac{1}{x} e^{-xT} \end{aligned}$$

で評価できるので、

$$\int_T^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt \in \left[-\frac{1}{1-e^{-T}} \frac{1}{x} e^{-xT}, \frac{1}{T} e^{-T} \right]$$

となる。この区間幅をほぼ e^{-T} と見て、これが machine epsilon 程度になるように T を決めることにする。すなわち、working precision を 2^{-d} とすると $e^{-T} = 2^{-d}$ なので、$T = d \log 2$ と決める。

(9) の前半は、直接数値積分で評価する。数値積分は、ベキ級数展開 (Type-II PSA) に基づく方法で行う。ただし、原点で級数展開出来ないため、

$$= \int_0^S \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt + \int_S^T \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt$$

のように分解し、後半は普通に積分する。前半は、0 が除去可能特異点となっているため、

$$\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} = \frac{e^{-t} - \frac{e^{-xt}}{\frac{1-e^{-t}}{t}}}{t}$$

のように変形し、2 箇所の t で割っている部分でベキ級数を「約分」する。

パラメータは要調整だが、現在は S は自動決定、PSA の次数 $= 16$ としている。

3.3 アルゴリズム 1

$|x|$ が大きいときに (8) を何度も使うと精度が劣化し、計算時間もかかってしまう問題をなんとかする。

前節と同様に (7) を直接積分する方針だが、$1 \le x \le 2$ の範囲ではなく、$x \ge 1$ の場合にこの式で計算する。$0 \le x < 1$ ならば、

$$\mathrm{digamma}(x) = \mathrm{digamma}(x+1) - \frac{1}{x}$$

で変換する。$x < 0$ ならば、

$$\begin{aligned}\mathrm{digamma}(x) &= \mathrm{digamma}(1-x) - \frac{\pi}{\tan \pi x} \\ &= \mathrm{digamma}(1-x) - \pi \frac{\cos \pi x}{\sin \pi x}\end{aligned}$$

で変換する。前者の形だと $x = -0.5, -1.5, -2.5, \ldots$ で都合が悪いので後者の形で計算するとよい。

3.4 アルゴリズム 2

(2021/6/6 頃)

アルゴリズム 0,1 での (7) の代わりに、$1 \le x$ に対して

$$\mathrm{digamma}(x) = \log x + \int_0^\infty \left(\frac{1}{t} - \frac{1}{1-e^{-t}} \right) e^{-xt} dt \tag{10}$$

を使って計算する方針。

(10) の積分の部分をある定数 $T > 0$ を用いて

$$\int_0^\infty \left(\frac{1}{t} - \frac{1}{1-e^{-t}} \right) e^{-xt} dt = \int_0^T \left(\frac{1}{t} - \frac{1}{1-e^{-t}} \right) e^{-xt} dt + \int_T^\infty \left(\frac{1}{t} - \frac{1}{1-e^{-t}} \right) e^{-xt} dt \tag{11}$$

と 2 つに分ける。

(11) の後半部分

$$\int_T^\infty \left(\frac{1}{t} - \frac{1}{1-e^{-t}} \right) e^{-xt} dt$$

を区間評価する。上界は、

$$\int_T^\infty \left(\frac{1}{t} - \frac{1}{1-e^{-t}} \right) e^{-xt} dt \le \int_T^\infty \frac{e^{-xt}}{t} dt \le \int_T^\infty \frac{e^{-xt}}{T} dt = \frac{1}{T} \left[-\frac{1}{x} e^{-xt} \right]_T^\infty = \frac{1}{xT} e^{-xT}$$

で、下界は、

$$\begin{aligned}\int_T^\infty \left(\frac{1}{t} - \frac{1}{1-e^{-t}}\right) e^{-xt}dt &\geq -\int_T^\infty \frac{e^{-xt}}{1-e^{-t}}dt \geq -\int_T^\infty \frac{e^{-xt}}{1-e^{-T}}dt \\ &= -\frac{1}{1-e^{-T}}\left[-\frac{1}{x}e^{-xt}\right]_T^\infty \\ &= -\frac{1}{1-e^{-T}}\frac{1}{x}e^{-xT}\end{aligned}$$

で評価できるので、

$$\int_T^\infty \left(\frac{1}{t} - \frac{1}{1-e^{-t}}\right) e^{-xt}dt \in \left[-\frac{1}{1-e^{-T}}\frac{1}{x}e^{-xT}, \frac{1}{xT}e^{-xT}\right]$$

となる。この区間幅をほぼ e^{-xT} と見て、これが machine epsilon 程度になるように T を決めることにする。すなわち、working precision を 2^{-d} とすると $e^{-xT} = 2^{-d}$ なので、$T = \dfrac{d\log 2}{x}$ と決める。

(11) の前半は、直接数値積分で評価する。数値積分は、ベキ級数展開 (Type-II PSA) に基づく方法で行う。ただし、原点で級数展開出来ないため、

$$= \int_0^s \left(\frac{1}{t} - \frac{1}{1-e^{-t}}\right) e^{-xt}dt + \int_S^T \left(\frac{1}{t} - \frac{1}{1-e^{-t}}\right) e^{-xt}dt$$

のように分解し、後半は普通に積分する。前半は、0 が除去可能特異点となっているため、

$$\frac{1}{t} - \frac{1}{1-e^{-t}} = \frac{1 - \frac{1}{\frac{1-e^{-t}}{t}}}{t}$$

のように変形し、2 箇所の t で割っている部分でベキ級数を「約分」する。

3.5 アルゴリズム 3

(2021/6/13 頃)

アルゴリズム 2 での (10) の代わりに、$1 \leq x$ に対して

$$\mathrm{digamma}(x) = \log x - \frac{1}{2x} - 2\int_0^\infty \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt \tag{12}$$

を使って計算する方針。

(12) の積分の部分をある定数 $T > 0$ を用いて

$$\int_0^\infty \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt = \int_0^T \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt + \int_T^\infty \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt \tag{13}$$

と2つに分ける。

(13) の後半部分

$$\int_T^\infty \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt$$

を区間評価する。下界は0。上界を評価する。

$$\frac{1}{e^{2\pi t}-1} = \frac{e^{-2\pi t}}{1-e^{-2\pi t}} \le \frac{e^{-2\pi t}}{1-e^{-2\pi T}}$$

である。また、$\dfrac{t}{t^2+x^2}$ は、$\dfrac{d}{dt}\left(\dfrac{t}{t^2+x^2}\right) = \dfrac{x^2-t^2}{(t^2+x^2)^2}$ より $t=x$ で最大値を取るので、

$$\frac{t}{t^2+x^2} \le \frac{x}{x^2+x^2} = \frac{1}{2x}$$

が成立する。よって、

$$\int_T^\infty \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt \le \frac{1}{2x}\frac{1}{1-e^{-2\pi T}}\int_T^\infty e^{-2\pi t}dt = \frac{1}{4\pi x}\frac{e^{-2\pi T}}{1-e^{-2\pi T}}$$

と抑えられる。すなわち

$$\int_T^\infty \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt \in \left[0, \frac{1}{4\pi x}\frac{e^{-2\pi T}}{1-e^{-2\pi T}}\right]$$

が成立する。この区間幅をほぼ $e^{-2\pi T}$ と見て、これが machine epsilon 程度になるように T を決めることにする。すなわち、working precision を 2^{-d} とすると $e^{-2\pi T} = 2^{-d}$ なので、$T = \dfrac{d\log 2}{2\pi}$ と決める。

(13) の前半部分

$$\int_0^T \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt \tag{14}$$

の計算を考える。まず、直接数値積分することを考える。ベキ級数展開 (Type-II PSA) に基づく方法で行う。ただし、原点で級数展開出来ないため、

$$= \int_0^S \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt + \int_S^T \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt$$

のように分解し、後半は普通に積分する。前半は、0が除去可能特異点となっているため、

$$\frac{t}{e^{2\pi t}-1}$$

の部分でベキ級数を「約分」する。

(14) の計算は、x が巨大なときに (ベキ級数の係数に巨大数が混じるなどして) 難しくなるが、一方で x が巨大なときは非常に小さな値にしかならない。そこで、更に工夫することにする。$\dfrac{t}{e^{2\pi t}-1} \ge 0$ で定符号なので、ある $\theta \in [0,T]$ が存在して

$$\frac{1}{\theta^2+x^2}\int_0^T \frac{t}{e^{2\pi t}-1}dt = \int_0^T \frac{t}{(t^2+x^2)(e^{2\pi t}-1)}dt$$

が成立する。よって、(14) の計算は (精度は落ちるが)

$$\frac{1}{[0,T]^2+x^2}\int_0^T \frac{t}{e^{2\pi t}-1}dt \tag{15}$$

と計算してもよい。この項の区間幅を評価してみると、

$$\int_0^T \frac{t}{e^{2\pi t}-1}dt \leq \int_0^\infty \frac{t}{e^{2\pi t}-1}dt = \frac{1}{24}$$

で、

$$\frac{1}{[0,T]^2+x^2} = \frac{1}{[x^2, x^2+T^2]} = \left[\frac{1}{x^2+T^2}, \frac{1}{x^2}\right]$$

なので

$$\frac{1}{24}\left(\frac{1}{x^2}-\frac{1}{x^2+T^2}\right) = \frac{1}{24}\frac{T^2}{x^2(x^2+T^2)} \simeq \frac{T^2}{x^4}$$

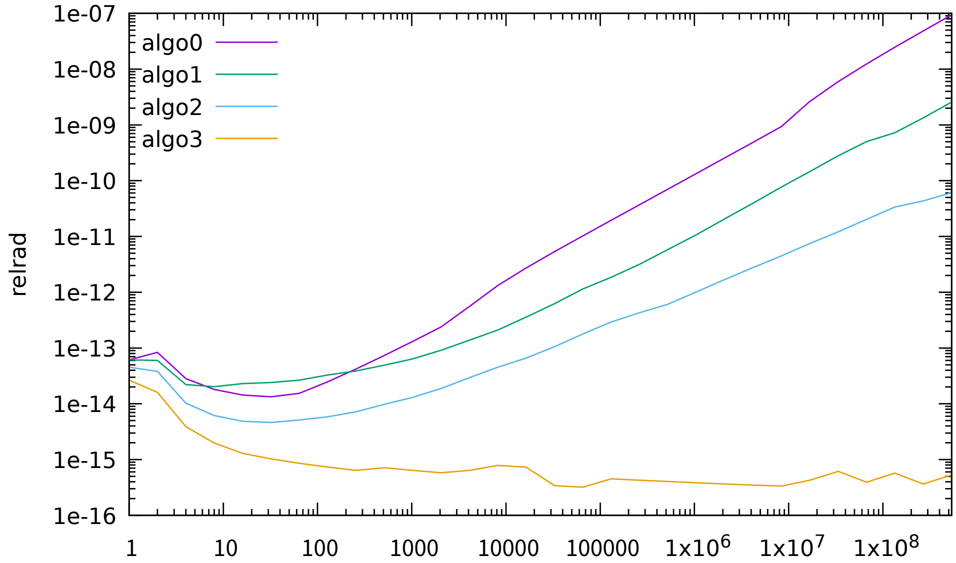
とする。machine epsilon を 2^{-d} とすると $\dfrac{T^2}{x^4} \leq 2^{-d}$ すなわち

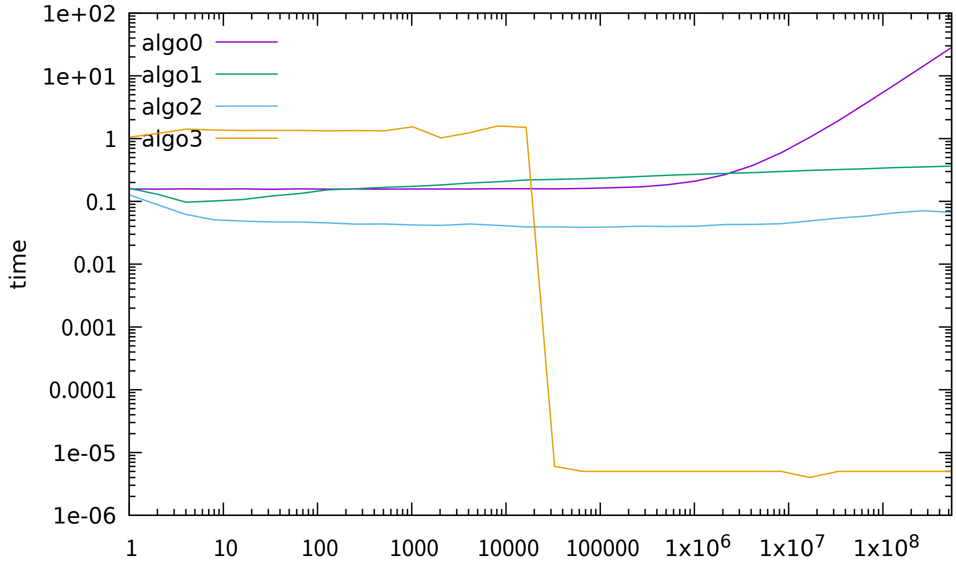
$$x \geq 2^{\frac{d}{4}}\sqrt{T}$$

程度に x が大きいとき、(14) の計算を (15) で行うことにする。(15) の計算の数値積分の部分はもはや x を含んでいないので、一回だけ計算して保存しておけば計算時間が大きく改善する。

3.6 ディガンマ関数まとめ

以上のアルゴリズム 0,1,2,3 について、入力 x に対する出力の相対半径、計算時間をグラフにしたものを以下に示す。





アルゴリズム 1 は、アルゴリズム 0 に対して精度、計算時間ともに改善が認められるが、満足いくものではない。

アルゴリズム 2 は、$\log x$ との差のみを積分で計算する式になっているので、精度、計算時間ともに大きく改善した。

アルゴリズム 3 は、$\log x - \frac{1}{2x}$ との差のみを積分で計算する式で、精度はアルゴリズム 2 に比べて大きく改善した。計算時間は x の大きさでの場合分けの影響が大きく、x が小さい部分では 1 秒程度とかなり遅いが、x が大きい部分では積分を事前に計算しておけるので超高速になっている。

2021/6/26 時点での実装は、アルゴリズム 3 が高速な領域、すなわち

$$x \geq 2^{\frac{d}{4}} \sqrt{\frac{d \log 2}{2\pi}}$$

ではアルゴリズム 3 を、それより小さい部分ではアルゴリズム 2 を使うハイブリッドになっている。更に改良を模索したい。(8) を逆に使って、x を大きい方にずらしてしまう作戦もありそう。

4 トリガンマ関数

(2014/6/7 のノートに従う)

$$\mathrm{trigamma}(x) = \frac{d}{dx}\mathrm{digamma}(x) \tag{16}$$

である。積分形

$$\mathrm{trigamma}(x) = \int_0^\infty \frac{te^{-xt}}{1-e^{-t}}dt \tag{17}$$

を直接計算する方針。ただし、

$$\mathrm{trigamma}(x+1) = \mathrm{trigamma}(x) - \frac{1}{x^2} \tag{18}$$

で変換できることも考慮する。

$1 \leq x \leq 2$ の範囲で (17) で計算し、(18) で変換する方針にする。

定数 $T > 0$ とし、(17) を

$$\int_0^\infty \frac{te^{-xt}}{1-e^{-t}}dt = \int_0^T \frac{te^{-xt}}{1-e^{-t}}dt + \int_T^\infty \frac{te^{-xt}}{1-e^{-t}}dt \tag{19}$$

と分解する。

(19) の後半は、次のように評価する。

$$\begin{aligned}
\int_T^\infty \frac{te^{-xt}}{1-e^{-t}}dt &\leq \int_T^\infty \frac{te^{-xt}}{1-e^{-T}}dt \\
&= \frac{1}{1-e^{-T}}\left[-\frac{e^{-xt}(xt+1)}{x^2}\right]_T^\infty \\
&= \frac{e^{-xT}(xT+1)}{(1-e^{-T})x^2}
\end{aligned}$$

積分値は明らかに正なので、

$$\int_T^\infty \frac{te^{-xt}}{1-e^{-t}}dt \in \left[0, \frac{e^{-xT}(xT+1)}{(1-e^{-T})x^2}\right]$$

と評価できる。

(19) の前半は、直接数値積分で評価する。数値積分は、ベキ級数展開 (Type-II PSA) に基づく方法で行う。ただし、原点で級数展開出来ないため、

$$\frac{te^{-xt}}{1-e^{-t}} = \frac{e^{-xt}}{\frac{1-e^{-t}}{t}}$$

のように変形し、t で割っている部分でベキ級数を「約分」する。

パラメータは要調整だが、現在は S は自動決定、T は仮数部の bit 数 $\times \log 2$、PSA の次数 $= 14$ としている。

以上で $1 \leq x \leq 2$ の範囲のトリガンマ関数が計算できるようになった。それ以外の範囲については、(18) を用いて、$n = \lfloor x \rfloor - 1$ として、$x > 2$ に対しては

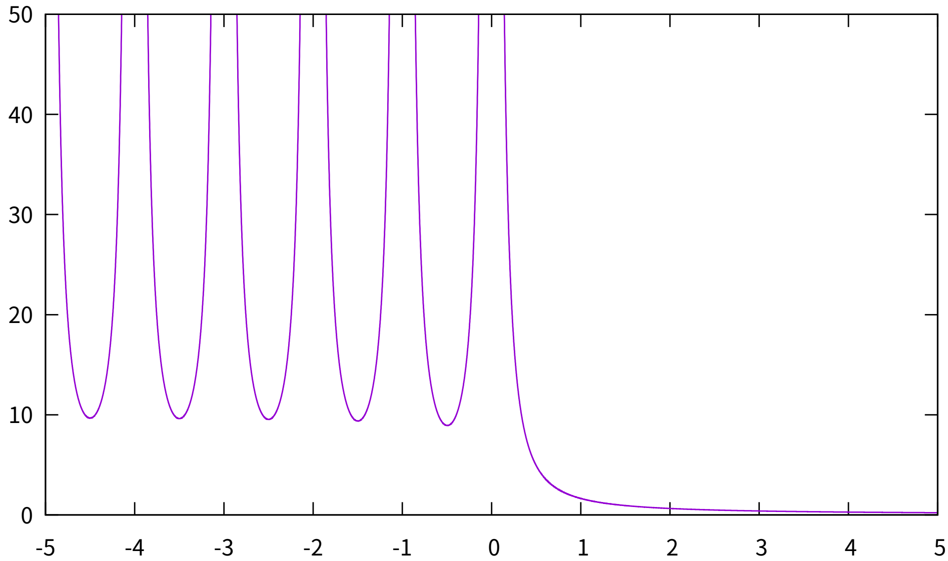
$$\mathrm{trigamma}(x) = \mathrm{trigamma}(x-n) - \sum_{i=1}^{n} \frac{1}{(x-i)^2}$$

$x < 1$ に対しては

$$\mathrm{trigamma}(x) = \mathrm{trigamma}(x-n) + \sum_{i=n+1}^{0} \frac{1}{(x-i)^2}$$

で計算できる。

トリガンマ関数の概形は図の通り。



トリガンマに関しては、幅の大きな区間の入力をきちんと扱っているのは現状正の領域のみ。負の領域に幅の大きな区間を入れると、精度保証はされているものの非常に大きな区間が出てくる場合がある。(トリガンマの極値点を精度保証するのをサボっている。)

5　ログガンマ関数

$$\mathrm{lgamma}(x) = \log |\Gamma(x)|$$

Γ 関数は大きな数が出てきて簡単にオーバーフローやアンダーフローが起きてしまうので、log を取った lgamma 関数がよく使われる。

計算は次のように行っている。$n = \lfloor x \rfloor - 1$ として、$x > 2$ に対しては

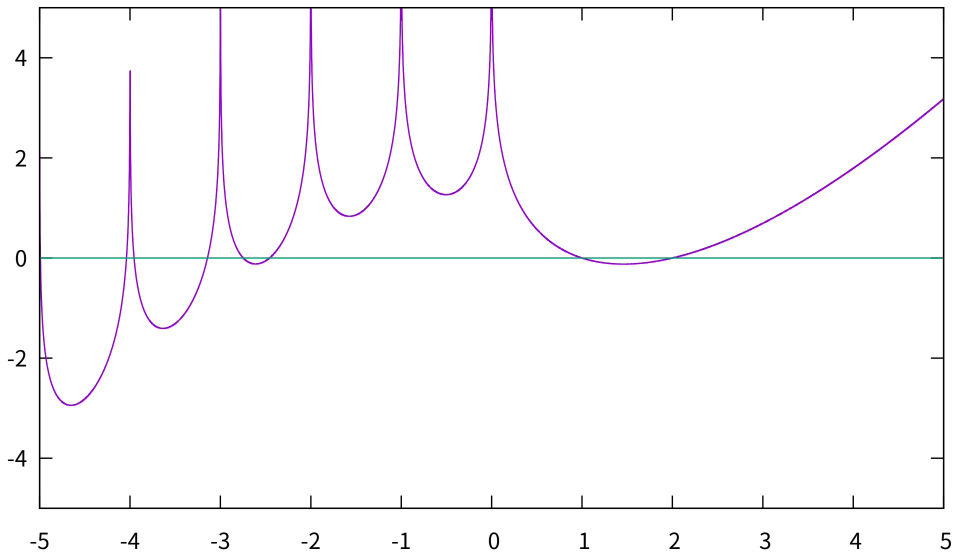
$$\mathrm{lgamma}(x) = \log(\Gamma(x-n)) + \sum_{i=1}^{n} \log(x-i)$$

$x < 1$ に対しては

$$\mathrm{lgamma}(x) = \log(\Gamma(x-n)) - \sum_{i=n+1}^{0} \log|x-i|$$

とする。

大きな区間の入力に対しては、次のようにする。入力を $I = [\underline{I}, \overline{I}]$ とする。lgamma 関数の概形は図の通り。



よって、I が 0 または負の整数を含んでいる場合は、上限は ∞、下限は複雑な場合分けになるが省略。それ以外の場合は、lgamma 関数の極値点を含む場合は

$$\mathrm{hull}(\mathrm{lgamma}(\underline{I}), \mathrm{lgamma}(\overline{I}), \mathrm{lgamma}(\text{極値点}))$$

を、極値点を含まない場合は

$$\mathrm{hull}(\mathrm{lgamma}(\underline{I}), \mathrm{lgamma}(\overline{I}))$$

を結果とする。

6 おまけ

なお、INTLAB では version7 からガンマ関数の精度保証付き計算が出来る。アルゴリズムは、本手法と同様に $1 \leq x \leq 2$ の範囲の計算に還元するが、その範囲の計算は

- 区間 $[1, 2]$ を更に細かい区間に分割する
- その各区間毎に、数式処理ソフトにより計算した値を元にして近似多項式とその最大誤差を予め計算しておき、その多項式を計算する。

という手法によっているらしい。恐らく間違った値を計算することはないとは思うが、「数式処理ソフトの計算値は精度保証されているのか」という疑問に答えられないように思える。

しかし、本稿で書いた手法そのままでは時間がかかりすぎる。本稿で書いた手法で生成した近似多項式を使うのがいいのかな。近似計算では Lanczos approximation が有力らしいが、それで精度保証は出来るのかな。