

漸化式を用いるクンマー関数 $U(a, b, x)$ の数値計算法

Recurrence Technique for the Numerical Computation of
Kummer Function $U(a, b, x)$

吉田 年雄
Toshio Yoshida

$a > 0$, $b \geq 0$, $x > 0$ の場合のクンマー関数 $U(a, b, x)$ は, 漸化式を用いる数値計算法, すなわち, Miller の方法により計算できる. $a = a' + n$ ($0 < a' \leq 1$, n : 非負の整数) として, a' と n を定義する. m を適当に選ばれた正の整数とし, α を小さな任意定数とする. $F_{m+1}(x) = 0$ と $F_m(x) = \alpha$ を出発値として, $f_n(x) = (\Gamma(a' + n)/\Gamma(a'))U(a' + n, b, x)$ が満たす漸化式 $F_{k-1}(x) = \{(x + 2a' - b + 2k)F_k(x) - (a' - b + k + 1)F_{k+1}(x)\}/(a' + k - 1)$ を繰り返し使うことにより, $f_n(x)$ の値は計算式 $f_n(x) \div x^{-a'} F_n(x) / \sum_{k=0}^m \Gamma(a' - b + k + 1)/(k! \Gamma(a' - b + 1)) F_k(x)$ により求めることができる. 本稿では, この $f_n(x)$ の計算法の誤差解析の結果を用いて, 要求精度で関数値を計算するために必要な漸化式の繰り返し回数を求めている. この繰り返し回数は, a , b , x の値により変わる. 与えられた任意の a , b , x の値に対して, 漸化式の繰り返し回数は簡単に定めることはできない. それで, 漸化式の繰り返し回数を与えなくても, 任意の要求精度で自動的に関数値を求めることができるドイフルハートの方法を提案する.

キーワード: クンマー関数 漸化式を用いる計算法 ドイフルハートの方法

1 はじめに

クンマー関数¹⁾は合流型超幾何関数とも呼ばれ, 各種の特殊関数を含む重要な関数である. 本稿では, 以下, n は整数, ν , x , a , b は実数とする. このクンマー関数が満足する微分方程式は

$$x \frac{d^2 w}{dx^2} + (b - a) \frac{dw}{dx} - aw = 0 \quad (1)$$

であり, $x = 0$ で確定特異点, ∞ で不確定特異点をもつ. この微分方程式の解は, 第1種クンマー関数

$$M(a, b, x) = 1 + \frac{ax}{b} + \frac{(a)_2 x^2}{2!(b)_2} + \cdots + \frac{(a)_n x^n}{n!(b)_n} + \cdots \quad (2)$$

と第2種クンマー関数

$$U(a, b, x) = \frac{\Gamma(1 - b)}{\Gamma(a - b + 1)} M(a, b, x) + \frac{\Gamma(b - 1)}{\Gamma(a)} x^{1-b} M(a - b + 1, 2 - b, x) \quad (3)$$

である。ここで、 $(a)_n$ はポツホハンマーの記号

$$(a)_n = \frac{\Gamma(a+n)}{\Gamma(a)} = a(a+1)(a+2)\cdots(a+n-1) \quad (n \geq 1), \quad (a)_0 = 1 \quad (4)$$

である。ここで、 $\Gamma(a)$ はガンマ関数である。

$U(a, b, x)$ の特別な場合は、次式に示すように各種の特殊関数となる。

$$U\left(\nu + \frac{1}{2}, 2\nu + 1, 2x\right) = \pi^{-\frac{1}{2}} e^x (2x)^{-\nu} K_\nu(x) \quad (K_\nu(x) : \text{第2種変形ベッセル関数}) \quad (5)$$

$$U\left(\nu + \frac{1}{2}, 2\nu + 1, -2ix\right) = \frac{1}{2} \pi^{\frac{1}{2}} e^{i[\pi(\nu+\frac{1}{2})-x]} (2x)^{-\nu} H_\nu^{(1)}(x) \quad (H_\nu^{(1)}(x) : \text{第1種ハンケル関数}) \quad (6)$$

$$U\left(\nu + \frac{1}{2}, 2\nu + 1, 2ix\right) = \frac{1}{2} \pi^{\frac{1}{2}} e^{-i[\pi(\nu+\frac{1}{2})-x]} (2x)^{-\nu} H_\nu^{(2)}(x) \quad (H_\nu^{(2)}(x) : \text{第2種ハンケル関数}) \quad (7)$$

$$U(1-a, 1-a, x) = e^x \Gamma(a, x) \quad (\Gamma(a, x) : \text{不完全ガンマ関数}) \quad (8)$$

$$U(1, 1, -x) = -e^{-x} \text{Ei}(x) \quad (\text{Ei}(x) : \text{積分指数関数}) \quad (9)$$

$$U\left(-\frac{1}{2}\nu, 0, 2x\right) = \Gamma\left(1 + \frac{1}{2}\nu\right) e^x k_\nu(x) \quad (k_\nu(x) : \text{Bateman 関数}) \quad (10)$$

$$U\left(-\frac{1}{2}\nu, \frac{1}{2}, \frac{1}{2}x^2\right) = 2^{-\frac{1}{2}\nu} e^{x^2/4} D_\nu(x) \quad (D_\nu(x) : \text{Weber 関数}) \quad (11)$$

$$U\left(\frac{1}{2}, \frac{1}{2}, x^2\right) = \pi^{\frac{1}{2}} e^{x^2} \text{erfc } x \quad (\text{erfc } x : \text{余誤差関数}) \quad (12)$$

この $U(a, b, x)$ に対して、Temme^{3, 4)} により、漸化式を用いる計算法、すなわち、ミラー (Miller) の方法が提案されている。しかし、今までに、この計算法の誤差解析は行われていない。それで、本論文では、 $a \geq 0, b \geq 0, x > 0$ の場合について、漸化式を用いる $U(a, b, x)$ の数値計算法の誤差解析を述べる。

ミラーの方法は、予め、計算の前に繰り返し回数を決めておかなければならない。しかし、与えられた任意の a, b, x の値に対して、漸化式の繰り返し回数は簡単に定めることはできない。それで、漸化式の繰り返し回数を与えなくても、任意の要求精度で自動的に関数値を求めることができるドイフルハートの方法を提案する。

本稿では、 $a \geq 0, b \geq 0, x > 0$ の場合のクンマー関数 $U(a, b, x)$ の計算法について述べている。この方法は、 x が大きく、 b が大きくない場合に有効である。

2 漸化式を用いる計算法 (ミラーの方法)

$U(a, b, x)$ の定義式 (3) を計算すると、第1項と第2項の和で桁落ちが起き、有効精度が失われる。それで、本稿では、

$$a = a' + n \quad (0 < a' \leq 1 \quad n : \text{非負の整数}) \quad (13)$$

として、

$$f_n(x) = \frac{\Gamma(a' + n)}{\Gamma(a')} U(a' + n, b, x) \quad (14)$$

に対して、漸化式を用いる方法、すなわち、ミラーの方法を適用する。以後、この $f_n(x)$ の計算法について述べる。

m を適当に選ばれた正の整数とし、 α を小さな任意定数とする。

$$F_{m+1}(x) = 0, \quad F_m(x) = \alpha \quad (15)$$

を初期値として、式 (14) の関数 $f_k(x)$ が満足する漸化式

$$(a' + k - 1)F_{k-1}(x) = (x + 2a' - b + 2k)F_k(x) - (a' - b + k + 1)F_{k+1}(x) \quad (16)$$

を繰り返し使うことにより、 $F_{m-1}(x)$, $F_{m-2}(x)$, $\dots$, $F_0(x)$ を順次、計算する。そのとき、ある $N(< m)$ に対して、 $n = 0, 1, \dots, N$ についての $f_n(x)$ の計算式は

$$f_n(x) \doteq x^{-a'} F_n(x) \bigg/ \sum_{k=0}^m \epsilon_k F_k(x) \quad (17)$$

として与えられる。ここで、

$$\epsilon_k = \frac{(a' - b + 1)_k}{k!} = \frac{\Gamma(a' - b + k + 1)}{k! \Gamma(a' - b + 1)} \quad (18)$$

である。なお、整数 m は漸化式の繰り返し回数となっている。

$U(a, b, x)$ の漸近展開式は

$$U(a, b, x) \asymp x^{-a} \sum_{k=0}^{\infty} \frac{(-1)^k (a)_k (a - b + 1)_k}{k! x^k} \quad (19)$$

で与えられ、 a あるいは $a - b + 1$ が 0 以下の整数の場合には有限級数となる。それは漸近展開式ではなくなるので厳密な式となる。それで、 $a = 0$ のときには、

$$U(0, b, x) = 1 \quad (20)$$

であることが分かる。

3 誤差解析

漸化式 (16) の一般解は

$$F_n(x) = \xi f_n(x) + \eta g_n(x) \quad (21)$$

として表される。ここで、

$$g_n(x) = \frac{\Gamma(a' + n)}{\Gamma(a' - b + n + 1)} M(a' + n, b, x) \quad (22)$$

であり、 ξ と η は任意定数である。これらの任意定数は式 (15) によって決められる。したがって、

$$F_{m+1}(x) = \xi f_{m+1}(x) + \eta g_{m+1}(x) = 0 \quad (23)$$

が得られる。式 (21) と (23) から η を消去すると次式を得る。

$$F_n(x) = \xi \left(f_n(x) - \frac{f_{m+1}(x) g_n(x)}{g_{m+1}(x)} \right) \quad (24)$$

$U(a' + k, b, x)$ の積分表示式を用いると、

$$\Gamma(a' + k) U(a' + k, b, x) = \int_0^\infty e^{-xt} t^{a'+k-1} (1+t)^{b-a'-k-1} dt \quad (25)$$

が得られる． ϵ_k を

$$\epsilon_k = \frac{(a' - b + 1)_k}{k!} = \frac{\Gamma(a' - b + 1 + k)}{k! \Gamma(a' - b + 1)} \quad (26)$$

と置く．そのとき，式 (25) に ϵ_k を乗じ， k についての和を取ると，

$$\begin{aligned} \sum_{k=0}^{\infty} \epsilon_k \Gamma(a' + k) U(a' + k, b, x) &= \int_0^{\infty} e^{-xt} t^{a'-1} (1+t)^{b-a'-1} \sum_{k=0}^{\infty} \frac{(a' - b + 1)_k}{k!} \left(\frac{t}{1+t} \right)^k dt \\ &= \int_0^{\infty} e^{-xt} t^{a'-1} (1+t)^{b-a'-1} \left(1 - \frac{t}{1+t} \right)^{-a'+b-1} dt \\ &= \int_0^{\infty} e^{-xt} t^{a'-1} dt \\ &= \Gamma(a') x^{-a'} \end{aligned} \quad (27)$$

となる．したがって， $f_k(x)$ に対して，次の関係式

$$\sum_{k=0}^{\infty} \epsilon_k f_k(x) = x^{-a'} \quad (28)$$

が成り立つ．式 (24) と上式より，

$$\sum_{k=0}^m \epsilon_k \left(\frac{F_k(x)}{\xi} + \frac{f_{m+1}(x) g_k(x)}{g_{m+1}(x)} \right) + \sum_{k=m+1}^{\infty} \epsilon_k f_k(x) = x^{-a'} \quad (29)$$

が得られる．式 (24) と上式から ξ を消去すると次式が求められる．

$$f_n(x) = \frac{x^{-a'} F_n(x)}{\sum_{k=0}^m \epsilon_k F_k(x)} \{1 - \Phi_m(a', b, x)\} + \frac{f_{m+1}(x) g_n(x)}{g_{m+1}(x)} \quad (30)$$

ただし，

$$\Phi_m(a', b, x) = x^{a'} \left(\sum_{k=0}^m \epsilon_k \frac{f_{m+1}(x) g_k(x)}{g_{m+1}(x)} + \sum_{k=m+1}^{\infty} \epsilon_k f_k(x) \right) \quad (31)$$

である．この $\Phi_m(a', b, x)$ は n に依存しないことに注意することが大切である．式 (30) は $f_n(x)$ の計算式 (17) の誤差を的確に表している．

式 (30) の右辺の第 2 項を $f_n(x)$ で割ったものを

$$\Theta_{mn}(a', b, x) = \frac{f_{m+1}(x) g_n(x)}{f_n(x) g_{m+1}(x)} \quad (32)$$

と置く．そのとき，式 (30) は次のように表される．

$$f_n(x) = \frac{x^{-a'} F_n(x)}{\sum_{k=0}^m \epsilon_k F_k(x)} \{1 - \Phi_m(a', b, x)\} + f_n(x) \Theta_{mn}(a', b, x) \quad (33)$$

計算式 (17) の相対精度

$$\varepsilon_{mn}(x) = \frac{x^{-a'} F_n(x) / \sum_{k=0}^m \epsilon_k F_k(x) - f_n(x)}{f_n(x)} \quad (34)$$

は、式 (33) を計算式 $x^{-a'} F_n / \sum_{k=0}^m \epsilon_k F_k(x)$ について解き、上式を代入すれば、

$$\varepsilon_{mn}(x) = \frac{\Phi_m(a', b, x) - \Theta_{mn}(a', b, x)}{1 - \Phi_m(a', b, x)} \quad (35)$$

が得られ、 $|\Phi_m(a', b, x)| \ll 1$ のときには、

$$\varepsilon_{mn}(x) \doteq \Phi_m(a', b, x) - \Theta_{mn}(a', b, x) \quad (36)$$

と近似できる。

式 (15) を初期値として、漸化式 (16) を繰り返し適用することにより得られた $F_{m-1}(x)$, $F_{m-2}(x)$, $\dots$, $F_0(x)$ を用いて、式 (17) により、10 進 p 桁の精度で $f_n(x)$ を計算できるためには、次の 2 つの不等式

$$|\Phi_m(a', b, x)| < 0.5 \times 10^{-p} \quad (37)$$

$$|\Theta_{mn}(a', b, x)| < 0.5 \times 10^{-p} \quad (38)$$

が成り立てばよい。

$f_n(x)$ の計算法について説明する。数値実験より、 $\Phi_m(\bar{a}, b, x)$ は m の単調減少関数であることが分かる。また、 $g_{m+1}(x)$ は、式 (22) より、 b が大きいときには、少なくとも $m > b$ で m の単調増加関数であることが分かる。したがって、少なくとも $m > b$ のときには、 $\Phi_m(\bar{a}, b, x)$ の評価式は m の単調減少関数であることが分かる。ここで、次の a) と b) を満たすものを M として定義する。

a) M' を、 $m \geq M'$ で式 (37) が成り立つ整数とする。

b) M' の最小値を M とする。

また、式 (38) ($\Theta_{mn}(\bar{a}, b, x)$ の不等式) で、 m を M に置き換えた式が満足する最大の n を N として定義する。そのとき、 $0 \leq n \leq N$ において、 $f_n(x)$ は、 m を M とした計算式 (17) により、10 進 p 桁の精度で計算できることになる。

$n > N$ である n に対しては、式 (38) は成り立たないので、 m の値は M より大きくする必要がある。すなわち、式 (38) が成り立つまで m を大きくしなければならない。

式 (37) と (38) の不等式から、要求精度で $f_n(x)$ を求めるための漸化式の繰り返し回数を決めることができる。 $\Phi_m(a', b, x)$ と $\Theta_{mn}(a', b, x)$ の値を求めるためには、 $f_n(x)$ と $g_n(x)$ を計算する必要がある。これらの値は誤差の評価に用いるので、数桁の有効桁があればよい。 $g_n(x)$ は式 (2) により求めることができる。 $f_n(x)$ の値は、定義式 (3) の計算で、有効桁が数桁残っている場合には、それを使えばよいし、または、本稿の方法を用いて、低い要求精度で値を求めてもよい。

4 漸化式の繰り返し回数

表 1 には、 $a' = 0.2$, $b = 0.1$ で要求精度が倍精度 (10^{-16}) のとき、 x を変えたときの M と N を示す。この表より、 M と N は、 x が大きいほど小さく、本方法は x が大きいときに有効な計算法であることが分かる。表 2 には、 $x = 50$, $a' = 0.2$ で要求精度が倍精度のとき、 b を 0 から 31 まで 1 刻みで変えたときの M と N を示す。 b の値により、 M は 10 から 27 までの値となっている。

このように、 M は、 a' , b , x により変わる。任意の値の a' , b , x に対して、 M を与えることは難しい。実際の応用では、例えば、 a' が $0.5 \leq a \leq 0.6$, b が $2 \leq b \leq 3$, x は $20 \leq x \leq 50$ というように限定された場合の関数値が必要になることが多いと思われるが、そのような場合には、 M は x の 1 次式で近似した $M = 0.244x + 24.2$ により与えることができる。しかし、汎用の計算プログラムでは、このように限定することはできない。そこで、漸化式の繰

表 1 $a' = 0.2, b = 0.1$ のときの M および N

x	M	N	x	M	N
2	177	44	60	13	5
3	121	31	70	12	4
5	76	20	80	12	5
8	51	14	90	11	4
10	42	12	100	11	4
20	25	8	200	8	3
30	19	6	300	7	3
40	16	5	400	7	3
50	14	5	500	6	2

表 2 $x = 50, a = 0.2$ のときの M および N

b	M	N	b	M	N
0	15	5	16	15	8
1	12	4	17	16	9
2	11	3	18	17	10
3	10	3	19	17	9
4	10	3	20	18	10
5	10	4	21	19	10
6	10	4	22	20	11
7	10	4	23	20	11
8	10	4	24	21	11
9	11	5	25	22	12
10	11	5	26	23	13
11	11	5	27	24	14
12	12	6	28	25	15
13	13	7	29	25	14
14	13	6	30	26	15
15	14	7	31	27	16

り返し回数を指定することなく，任意精度で関数値を求めることができるドイフルハート（Deuffhard）の方法^{2, 5)}を適用することが考えられる．

ドイフルハートの方法はミラーの方法を行列表現し，

$$\sum_{k=0}^{\tilde{m}} d_k f_k(x) \quad (39)$$

を要求精度で求める方法である（ d_k は定数係数， $\tilde{m}$ は適当な正整数）．また， $d_n = 1$ ($0 \leq n \leq \tilde{m}$)，他の d_k を 0 としたときの結果は，ミラーの方法で $f_n(x)$ を求めたものと等価である（打ち切り誤差が同じである）．それで，ミラーの方法の誤差解析の結果は，そのままドイフルハートの方法にも適用できる．

5 ドイフルハート (Deuflhard) の方法

ここでは, 3 項漸化式

$$a_k(x)T_k + b_k(x)T_{k+1} + c_k(x)T_{k+2} = 0$$

の最小解 $\{P_k\}$ ($\{P_k\}$ と線形独立である $\{Q_k\}$ に対して, $\lim_{k \rightarrow \infty} P_k/Q_k = 0$ であるもの) について, 重み d_k ($k = 0, 1, \dots, n$) つきの和

$$s_n = \sum_{k=0}^n d_k P_k \quad (40)$$

を任意の精度で求めることができるドイフルハートの方法²⁾を説明することにする. $m (> n)$ を適当な整数とする.

3 項漸化式に対するミラーの方法は次の $m+1$ 元の連立 1 次方程式を解くことと同じである.

$$\begin{aligned} a_k(x)P_k + b_k(x)P_{k+1} + c_k(x)P_{k+2} &= 0 \quad (k = 0, 1, \dots, m-2) \\ a_{m-1}(x)P_{m-1} + b_{m-1}(x)P_m &= 0 \\ \sum_{k=0}^m \epsilon_k P_k &= \gamma \end{aligned} \quad (41)$$

以下で, 正方行列とベクトルは $m+1$ 次とする. 上の連立 1 次方程式は

$$M\mathbf{p} = \mathbf{r} \quad (42)$$

と書くことができる. ここで,

$$M = \begin{bmatrix} a_0 & b_0 & c_0 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{m-2} & b_{m-2} & c_{m-2} & \\ & & & a_{m-1} & b_{m-1} & \\ \epsilon_0 & \cdots & \cdots & \cdots & \epsilon_m & \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} P_0 \\ \vdots \\ \vdots \\ \vdots \\ P_m \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \gamma \end{bmatrix}$$

である. $\mathbf{d}$ を

$$\mathbf{d} = (d_0, d_1, \dots, d_n, 0, \dots, 0)^T$$

とする. そのとき,

$$M^T \mathbf{u} = \mathbf{d} \quad (43)$$

とおいて, $\mathbf{u} = (u_0, u_1, \dots, u_m)^T$ を定義する. 式 (40) より, s_n は

$$s_n = \mathbf{d}^T \mathbf{p} = \mathbf{d}^T M^{-1} \mathbf{r} = (M^{-T} \mathbf{d})^T \mathbf{r} = \mathbf{u}^T \mathbf{r} = \gamma u_m \quad (44)$$

と表されるので, u_m を求めればよい. 式 (43) を解くために, M^T を LU 分解する.

$$M^T = LU \quad (45)$$

ここで,

$$L = \begin{bmatrix} a_0 & & & & & \\ b_0 & \ddots & & & & \\ c_0 & \ddots & a_{m-2} & & & \\ & \ddots & b_{m-2} & a_{m-1} & & \\ & & c_{m-2} & b_{m-1} & a_m & \end{bmatrix}, \quad U = \begin{bmatrix} 1 & & & & f_0 \\ & \ddots & & & \vdots \\ & & \ddots & & \vdots \\ & & & 1 & \vdots \\ & & & & f_m \end{bmatrix}$$

である. $\mathbf{f} = (f_0, f_1, \dots, f_m)^T$ とおき, 式 (45) の M^T の m 列に注目すると,

$$L\mathbf{f} = \boldsymbol{\epsilon} \quad (46)$$

が成り立つことが分かる. ただし, $\boldsymbol{\epsilon} = (\epsilon_0, \epsilon_1, \dots, \epsilon_m)^T$ である. 上式を解くと, $f_0, f_1, \dots, f_m$ が求められる.

次式

$$\mathbf{e} = U\mathbf{u} \quad (47)$$

のように, $\mathbf{e} = (e_0, e_1, \dots, e_m)^T$ を定義すると, 上式より,

$$e_m = f_m u_m \quad (48)$$

が得られる.

$$L\mathbf{e} = \mathbf{d} \quad (49)$$

を解いて, e_m を求めれば, 式 (48) より, u_m を決めることができる. したがって, s_n の近似値 $s_n^{(m)}$ を求めるアルゴリズムは, 式 (46), (49) と式 (44) より次のようになる.

$$\begin{aligned} f_{-1} &= 0, \quad f_0 = \frac{\epsilon_0}{a_0}, \quad e_{-1} = 0, \quad e_0 = \frac{d_0}{a_0} \\ &(\text{for } k = 1, 2, \dots, m) \{ \\ &\quad f_k = (\epsilon_k - b_{k-1}f_{k-1} - c_{k-2}f_{k-2})/a_k \\ &\quad e_k = (d_k - b_{k-1}e_{k-1} - c_{k-2}e_{k-2})/a_k \\ &\quad \} \\ s_n^{(m)} &= \gamma \frac{e_m}{f_m} \end{aligned} \quad (50)$$

上のアルゴリズムにおいて, 1 回の反復で 2 つの漸化式の計算が必要であることに注意しよう. また, 上述の行列とベクトルは $m+1$ 次であるが, 次元を 1 つ増やすには, 上のアルゴリズムで, そのまま k の繰り返しをもう 1 回追加すればよい. したがって, e_k/f_k が所要の精度内に収束するまで, k の繰り返しを行えばよいことになる.

このアルゴリズムでは, 1 回の反復で 2 つの漸化式の計算が必要である. また, 上述の行列とベクトルは $m+1$ 次であるが, 次元を 1 つ増やすには, 上のアルゴリズムで, そのまま k の繰り返しをもう 1 回追加すればよい. したがって, e_k/f_k が所要の精度内に収束するまで, k の繰り返しを行えばよい.

2. の「漸化式を用いる計算法 (Miller の方法)」で述べた関数については, Deuffhard の方法を適用することができる. Deuffhard の方法は, 任意の要求精度で関数値を求めることができるが, 計算時間は約 2 倍になる. しかし, 要求精度が低くてよい場合には, この方法が有利になる.

6 ドイフルハートの方法を用いた $f_n(x)$ の計算法

ドイフルハートの方法を用いた $f_n(x)$ の具体的な計算法は次のとおりである.

$f_{-1} = 0, \quad f_0 = \epsilon_0/a', \quad e_{-1} = 0, \quad e_0 = 1/a'$ を初期値として, 下の 2 つの式を $k = 1, 2, \dots$ と繰り返す.

$$\begin{cases} f_k = \{ \epsilon_k + (x + 2a' - b + 2k)f_{k-1} \\ \quad - (a' - b + k)f_{k-2} \} / (a' + k) \\ e_k = \{ d_k + (x + 2a' - b + 2k)e_{k-1} \\ \quad - (a' - b + k)e_{k-2} \} / (a' + k) \\ s_k = x^{-a'} e_k / f_k \end{cases} \quad (51)$$

この繰り返しは、 s_k が要求精度に収束するまで行う。ただし、 ϵ_k については、式 (18) より、

$$\epsilon_0 = 1, \quad \epsilon_k = \frac{a' - b + k}{k} \epsilon_{k-1} \quad (k \geq 1) \quad (52)$$

が成り立つので、これを利用するとよい。このようにして、 $f_n(x)$ を s_k の収束値として求めることができる。

ドイフルハートの方法は任意精度で関数値を求めることができるという利点があるが、式 (51) から分かるように、2 つの漸化式を計算しなければならないので、ミラーの方法の約 2 倍の計算量を必要とするという欠点もある。

7 適用範囲

漸化式 (51) の計算では、桁落ちが起こる場合がある。図 1 には、 $x \leq 500$ の場合について、桁落ちにより、関数の計算値の相対精度が 10 進 2 桁低下する b の最小値を点線で示す（これは、要求相対精度が 10^{-16} のとき、計算値の相対誤差が 10^{-14} 以上となる b の最小値である）。数値実験より、 $0 < a' \leq 1$ において、 a' が大きいほど、この最小値は大きくなるので、この図には、 $a' = 1$ の場合のものが示されている。この線より上の部分で桁落ちが起きるので、この点線より下が適用範囲となる。

実際に計算するときには、

$$b = 0.56x + 6.2 \quad (53)$$

の下領域を適用範囲とするとよい。

適用範囲外での $f_n(x)$ （したがって、 $U(a' + n, b, x)$ ）の関数値は、まず、適用範囲内の $U(a' + n, b, x)$ と $U(a' + n, b - 1, x)$ を計算し、次の漸化式を繰り返し用いて求めることができる。

$$U(a' + n, b + 1, x) = \{(x + b - 1)U(a' + n, b, x) + (a' + n - b + 1)U(a' + n, b - 1, x)\} / x \quad (54)$$

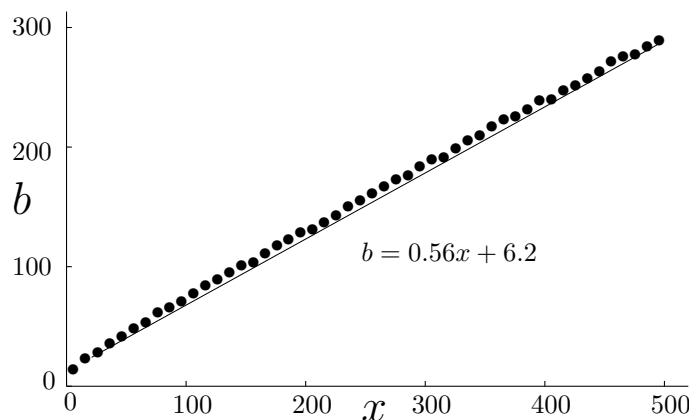


図 1 ドイフルハートの方法で桁落ちを起す領域（点線の上）

8 クンマー関数 $U(a, b, x)$ の計算プログラム

以下で示す $U(a, b, x)$ の計算プログラムのうち、最初の 2 つは直前で述べた適用範囲で有効である。

8.1 漸化式を用いる方法（Millerの方法）による $U(a' + n, b, x)$ の計算法

m を適当な整数として, $F_{m+1}(x) = 0$, $F_m(x) = \alpha$ (α : 任意定数) を初期値として, 漸化式 (16) を用いて, $F_{m-1}(x)$, $F_{m-2}(x)$, $\dots$, $F_0(x)$ を順次, 計算し, 式 (17) により, $f_n(x)$ を求め, 式 (14) から, $U(a' + n, b, x)$ の値を得ることができる.

計算式 (17) は

$$f_n(x) \doteq \frac{x^{-a'} F_n(x)}{\sum_{k=0}^m \epsilon_k F_k(x)} = \frac{x^{-a'} F_n(x)}{\sum_{k=0}^m \frac{\Gamma(a' - b + k + 1)}{k! \Gamma(a' - b + 1)} F_k(x)} \quad (55)$$

と表される. この漸化式を用いる方法において, ϵ_k ($k = 0, 1, \dots, m$) のうち, 最初に必要なものは

$$\epsilon_m = \frac{\Gamma(a' - b + m + 1)}{m! \Gamma(a' - b + 1)} \quad (56)$$

であるが, ガンマ関数を用いて計算することは能率的ではない.

$$\frac{\Gamma(\beta + k)}{\Gamma(\beta)} = \begin{cases} 1 & (k = 0) \\ \beta(\beta + 1) \cdots (\beta + k - 1) & (k \geq 1) \end{cases} \quad (57)$$

であることを利用し, 漸化式

$$\frac{\Gamma(\beta + k - 1)}{\Gamma(\beta)} = \frac{1}{(\beta - k - 1)} \frac{\Gamma(\beta + k)}{\Gamma(\beta)} \quad (58)$$

によって計算するのがよいと思われるが, この場合には, $\beta = a' - b + 1$ として, 最初に, $\beta(\beta + 1) \cdots (\beta + m - 1)$ の計算が必要になる. $m!$ の計算も必要になる.

そこで, 式 (55) を次のように変形する (分子, 分母を ϵ_m で割っている).

$$f_n(x) \doteq \frac{x^{-a'} F_n(x) \frac{m! \Gamma(a' - b + 1)}{\Gamma(a' - b + m + 1)}}{\sum_{k=0}^m \frac{\Gamma(a' - b + k + 1)}{k! \Gamma(a' - b + 1)} \frac{m! \Gamma(a' - b + 1)}{\Gamma(a' - b + m + 1)} F_k(x)} \quad (59)$$

上式は次のように表すことができる.

$$f_n(x) \doteq \frac{x^{-a'} \mathcal{E}_0 F_n(x)}{\sum_{k=0}^m \mathcal{E}_k F_k(x)} \quad (60)$$

ただし,

$$\mathcal{E}_k = \frac{m! \Gamma(a' - b + k + 1)}{k! \Gamma(a' - b + m + 1)} \quad (61)$$

である.

このようにすれば, $\mathcal{E}_m = 1$ を初期値として,

$$\mathcal{E}_{k-1} = \frac{k}{a' - b + k} \mathcal{E}_k \quad (62)$$

により, $\mathcal{E}_{m-1}$, $\mathcal{E}_{m-2}$, $\dots$, $\mathcal{E}_0$ を計算することができ, 能率的である.

以下のプログラムでは, a' は ad で表している. 4 番目の入力引数として, m がある. これは, 漸化式の繰り返し回数を表しており, 要求精度で関数値を求めることができる繰り返し回数を予め入力して置く必要がある. 適切な繰り返し回数を入力して置くことは, 一般には困難である.

```

double cfhgu_m(double a,double b,double x,int m)
{
    double ad,rm,f0,f1,f2,fn,rk,epsk,s,gabgan;
    int k,i,n;
    if(a==0.0)
        return 1.0;
    n=ceil(a)-1.0;
    ad=a-(double)n;
    rm=m;
    f2=0.0;
    f1=1.0e-100;
    epsk=1.0;
    s=f1;
    for(k=m;k>=1;--k){
        rk=k;
        f0=((x+ad+ad+rk+rk-b)*f1+(b-rk-ad-1.0)*f2)/(ad+rk-1.0);
        f2=f1;
        f1=f0;
        if(k==n+1)
            fn=f0;
        epsk=epsk/(ad-b+rk)*rk;
        s=s+f0*epsk;
    }
    if(n==0)
        gabgan=1.0;
    else{
        gabgan=1.0;
        for(i=1;i<=n;++i)
            gabgan=gabgan/(ad+(double)(i-1));
    }
    return pow(x,-ad)*fn*epsk/s*gabgan;
}

```

8.2 ドイフルハートの方法による $U(a, b, x)$ の計算プログラム

以下のプログラムでは、 a' は ad で表している。4 番目の入力引数である eps に要求精度を与えれば、その精度で関数値が求められる。

```

double cfhgu_dh(double a,double b,double x,double eps)
{
    double ad,fm,f0,f1,em,e0,e1,epsk,rk,xa,s,sold,ak,bk,ck,gabgan;
    int k,i,n;
    if(a==0.0)
        return 1.0;
    n=ceil(a)-1.0;
    ad=a-(double)n;
    fm=0.0;
    f0=1.0/ad;

```

```

em=0.0;
if (n==0)
    e0=f0;
else
    e0=0.0;
epsk=1.0;
k=1;
rk=k;
xa=pow(x,-ad);
s=1.0;
do{
    sold=s;
    epsk=epsk*(ad-b+rk)/rk;
    ak=ad+rk;
    bk=b-x-ad-ad-rk-rk;
    ck=ad-b+rk;
    f1=(epsk-bk*f0-ck*fm)/ak;
    if (k==n)
        e1=(1.0-bk*e0-ck*em)/ak;
    else
        e1=(-bk*e0-ck*em)/ak;
    k=k+1;
    rk=k;
    if (f1!=0.0)
        s=xa*e1/f1;
    else
        s=1.0e300;
    fm=f0;
    f0=f1;
    em=e0;
    e0=e1;
}while(k<=n||fabs((s-sold)/s)>eps);
if (n==0)
    gabgan=1.0;
else{
    gabgan=1.0;
    for(i=1;i<=n;++i)
        gabgan=gabgan/(ad+(double)(i-1));
}
return s*gabgan;
}

```

8.3 適用範囲を考慮したドイフルハートの方法による $U(a, b, x)$ の計算プログラム

適用範囲を考慮したプログラムを次に示す。このプログラムには、8.2 の関数 cfhgu_dh を付加することが必要である。

```

#include <stdio.h>
#include <math.h>

```

```

double cfhgu_dh_r(double,double,double,double);
int main(void)
{
    double a,b,x,eps,v;
    int n,ib;
    a=1.2;
    b=2.3;
    x=20.0;
    eps=1.0e-16;
    v=cfhgu_dh_r(a,b,x,eps);
    printf("a=%f b=%f x=%f v=%23.16e\n",a,b,x,v);
}

double cfhgu_dh(double,double,double,double);
double cfhgu_dh_r(double a,double b,double x,double eps)
{
    double xc,bc,fb,bmbc,b0,b1,u0,u1,u2,ri,b12ri;
    int i,ib,ibmbc;
    if(a<0||b<0||x<=0){
        printf("parameter error!\n");
        return -1;
    }
    bc=0.56*x+6.2;
    if(b<bc)
        return cfhgu_dh(a,b,x,eps);
    ib=b;
    fb=b-(double)ib;
    bmbc=b-bc;
    ibmbc=bmbc;
    b1=b-(double)ibmbc;
    b0=b1-1.0;
    u0=cfhgu_dh(a,b0,x,eps);
    u1=cfhgu_dh(a,b0+1.0,x,eps);
    for(i=1;i<=ibmbc;++i){
        ri=i;
        b12ri=b1-2.0+ri;
        u2=((x+b12ri)*u1+(a-b12ri)*u0)/x;
        u0=u1;
        u1=u2;
    }
    return u2;
}

```

9 おわりに

本稿では、特殊関数の数値計算法として、漸化式を用いる方法（Miller の方法）について説明した。さらに、この漸化式を用いる方法を行列表示し、精度更新型にして、任意の要求精度で関数値を計算できる Dueflhard の方法についても説明した。さらに、 $U(a,b,x)$ を計算するための C 言語プログラムを載せた。

参考文献

- [1] Abramowitz, M., Stegun, I.A., Handbook of Mathematical Functions, pp.503-515, Dover (1970).
- [2] Deuffhart, P., A Summation Technique for Minimal Solutions of Linear Homogeneous Difference Equations, Computing, 18, pp.1-13 (1977).
- [3] Temme, N.M.: The Numerical Computation of the Confluent Hypergeometric Function $U(a, b, z)$, *Numerische Mathematik*, Vol.41, pp.63-82 (1983).
- [4] Temme, N.M., Special Functions, An Introduction to the Classical Functions of Mathematical Physics, pp.171-177, John Wiley & Sons (2007).
- [5] 二宮市三編, 数値計算のわざ (第2章 吉田年雄: Bessel 関数の計算法), 共立出版, 東京, pp.21-41 (2006).



吉田年雄

工学部情報工学科教授

数値計算法の研究, 開発

特殊関数の数値計算法

ベッセル関数の数値計算法

級数の和の公式の研究