お話：数値解析 第8回
収束の加速法(後編)

長田直樹

1　はじめに

これまで3回にわたり、円周率およびライプニッツ級数の加速にまつわる収束の加速法の話をしてきた。前編(2008年8月号)では、線型収束の加速に有効なエイトケン Δ^2 法とリチャードソン補外について、中編と中編の2(9-10月号)では、交代級数の加速に有効なオイラー変換、e 変換(実装は ϵ 算法)、レヴィン t 変換を取り上げた。中編の2(10月号)では、一般化リチャードソン補外とその実装である E 算法についても話した。

今回は平方数の逆数の和 $\sum_{i=1}^{\infty} 1/i^2$ を巡る加速法の話から始める。

2　平方数の逆数の和

自然数の平方の逆数の和

$$\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots$$

を求めることは、17世紀後半から18世紀半ばまでヨーロッパの数学界において大きな問題(バーゼル問題)であった。この問題に解答を与えたのはL. オイラーである。彼は1735年サンクトペテルスブルグのアカデミーに提出した『逆数の級数の和について』[4]において

$$\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6} \tag{1}$$

と解決した。

バーゼル問題の答えを数値的に検証するためオイラーは、オイラー・マクローリンの公式(本誌2008年7月号)を用いて $s_\nu = \sum_{i=1}^{\nu} 1/i^2$ の漸近展開を求めた。1755年『微分学教程』[5, p.361]において、次の補題を与えている。

補題 1　$\nu \to \infty$ のとき、漸近展開

$$s_\nu = \sum_{i=1}^{\nu} \frac{1}{i^2} \sim \frac{\pi^2}{6} - \frac{1}{\nu} + \frac{1}{2\nu^2} - \sum_{k=1}^{\infty} \frac{B_{2k}}{\nu^{2k+1}}$$

が成立する。ここで、B_{2k} はベルヌイ数である。

証明　$\nu < \mu$ を任意の自然数とする。区間 $[\nu, \nu+\mu]$ において $f(x) = 1/x^2$ にオイラー・マクローリンの公式を適用する。μ 分割した複合台形公式による値を T_μ とする。

$$T_\mu = s_{\nu+\mu} - s_\nu + \frac{1}{2\nu^2} - \frac{1}{2(\nu+\mu)^2}$$

$$f^{(2k-1)}(x) = -\frac{(2k)!}{x^{2k+1}}$$

$$\int_\nu^{\nu+\mu} f(x)dx = \left[-\frac{1}{x}\right]_\nu^{\nu+\mu} = \frac{1}{\nu} - \frac{1}{\nu+\mu}$$

より、

$$s_{\nu+\mu} - s_\nu \sim \frac{1}{\nu} - \frac{1}{\nu+\mu} - \frac{1}{2\nu^2} + \frac{1}{2(\nu+\mu)^2} + \sum_{k=1}^{\infty} B_{2k}\left(-\frac{1}{(\nu+\mu)^{2k+1}} + \frac{1}{\nu^{2k+1}}\right)$$

$\mu \to \infty$ とすると、(1)より求める結果が得られる。□

偶数番目のベルヌイ数を書き出しておく。

表 1: 偶数番目のベルヌイ数

k	1	2	3	4	5
B_{2k}	$\frac{1}{6}$	$-\frac{1}{30}$	$\frac{1}{42}$	$-\frac{1}{30}$	$\frac{5}{66}$
k	6	7	8	9	10
B_{2k}	$-\frac{691}{2730}$	$\frac{7}{6}$	$-\frac{3617}{510}$	$\frac{43867}{798}$	$-\frac{174611}{330}$

3　オイラーの数値計算

補題1より

$$\frac{\pi^2}{6} = s_\nu + \frac{1}{\nu} - \frac{1}{2\nu^2} + \sum_{k=1}^{m} \frac{B_{2k}}{\nu^{2k+1}} + O\left(\frac{1}{\nu^{2m+3}}\right) \tag{2}$$

が成立する。オイラーは (2) において $\nu = 10, m = 8$ として s_ν の極限値 ($= \pi^2/6$) を計算した。オイラーの計算は表 2 のようである。

表 2: オイラーの計算 ($\nu = 10$)

s_ν	1.54976 77311 66540 690
$1/\nu$	0.1
$-1/2\nu^2$	−0.005
	1.64476 77311 66540 690
B_2/ν^3	0.00016 66666 66666 666
	1.64493 43978 33207 356
	中略
	1.64493 40668 48225 335
B_{14}/ν^{15}	0.00000 00000 00001 166
B_{16}/ν^{17}	−0.00000 00000 00000 071
	1.64493 40668 48226 430

小数点以下第 19 位を切り捨て (一部は四捨五入し) ているため、小数点以下第 18 位に丸め誤差がある。オイラーは計算の最後に「(ベルヌーイ数は) 発散するにもかかわらず、それ (1.644934066848226430) は真の和 ($\pi^2/6$) を与えることは明らかである。」[8] と述べている。

オイラーは漸近展開の $1/\nu^{17}$ までで打ち切っているが、$B_{18}/\nu^{19} = 0.00000\ 00000\ 00000\ 005$ まで用いると、より正確な 1.64493 40668 48226 435 が得られる。真値は 1.64493 40668 48226 43647 である。

オイラーは漸近展開の $1/\nu^{17}$ までで打ち切った理由を書いてないが、

1. $\sum_{i=1}^{\infty} 1/i^k, k = 2, \ldots, 16$ の値を 17 桁与えてある [5, p.365] ので、17 桁必要だったが余裕を持って 19 桁計算した。

2. $B_{18} = 43867/798, B_{20} = -174611/330$ と絶対値が大きくなる。

ことなどが考えられる。

4 対数収束の加速

s に収束する数列 $\{s_\nu\}$ が

$$\lim_{\nu\to\infty} \frac{s_{\nu+1} - s}{s_\nu - s} = 1$$

を満たすとき、対数収束 (2008 年 5 月号) という。級数 (1) の部分和 $s_\nu = \sum_{i=1}^{\nu} 1/i^2$ のなす数列は、補題 1 より

$$\lim_{\nu\to\infty} \frac{s_{\nu+1} - \dfrac{\pi^2}{6}}{s_\nu - \dfrac{\pi^2}{6}} = \lim_{\nu\to\infty} \frac{-\dfrac{1}{\nu+1} + O(\dfrac{1}{(\nu+1)^2})}{-\dfrac{1}{\nu} + O(\dfrac{1}{\nu^2})} = 1$$

なので $\pi^2/6$ に対数収束する。

対数収束は一般に収束が遅い。$s_\nu = \sum_{i=1}^{\nu} 1/i^2$ は 1000(10^3) 項までから 3 桁、100 万 (10^6) 項までから 6 桁しか正しい値は得られない。収束の速さはライプニッツ級数と同程度である。対数収束する数列の極限値を得るには、一般には何らかの加速法が必要になる。

対数収束する数列 $\{s_\nu\}$ の主な加速法は、大きく 4 つに分けられる。

1. 補正項 (誤差項) の情報を用いて加速する。

2. 部分列 (たとえば $\{s_\nu\}_{\nu=1,2,4,8,16,\ldots}$) を加速する。

3. 漸近展開の情報を用いて加速する。

4. 数列 $\{s_\nu\}$ に数列変換を適用する。

5 補正項の利用

補正は

$$\text{真値} = \text{近似値} + \text{補正}$$

により定義する。(2) の右辺第 2 項以下は補正項である。誤差 = 近似値 − 真値 なので補正は誤差の符号を変えたものである。

対数収束級数に対してではないが、交代級数に対する補正項を用いた加速は、オイラーより 300 年以上前にインドのマーダヴァによるライプニッツ級数の和を計算する公式 (本連載 10 月号) が起源である。

オイラー・マクローリンの公式により漸近展開の係数を求めることができれば、補正項による加速計算ができる。この方法は、3 節で述べたオイラーのバーゼル問題の数値計算 (表 2) に始まる。

6 部分列の加速

実用上現れる対数収束数列の多くは

$$s_\nu \sim s + \frac{c_1}{\nu} + \frac{c_2}{\nu^2} + \frac{c_3}{\nu^3} + \cdots \tag{3}$$

あるいは

$$s_\nu \sim s + \frac{c_1}{\nu^2} + \frac{c_2}{\nu^4} + \frac{c_3}{\nu^6} + \cdots \tag{4}$$

の型の漸近展開を持つ。

バーゼル問題の級数は (3) の場合である。直径 1 の円に内接する正 ν 角形の周長は

$$\nu \sin\frac{\pi}{\nu} = \pi - \frac{\pi^3}{3!\nu^2} + \frac{\pi^5}{5!\nu^4} - + \cdots \tag{5}$$

なので、(4) の場合である。なお、(5) は漸近展開の係数に未知の π を含んでいるので補正による加速には使えない。

数列 $\{s_\nu\}$ が

$$s_\nu \sim s + c_1\lambda_1^\nu + c_2\lambda_2^\nu + c_3\lambda_3^\nu + \cdots, \tag{6}$$

の形の漸近展開を持つとき、(6) は**幾何学的漸近展開** [12] と呼ばれる。幾何学的漸近展開をもつ数列は、エイトケン Δ^2 法、リチャードソン補外あるいは ϵ 算法などで加速できる。

(3) あるいは (4) を満たす数列の部分列 $\{s_{2^\nu}\}$ は

$$s_{2^\nu} \sim s + c_1(2^{-1})^\nu + c_2(2^{-2})^\nu + c_3(2^{-3})^\nu + \cdots$$
$$s_{2^\nu} \sim s + c_1(4^{-1})^\nu + c_2(4^{-2})^\nu + c_3(4^{-3})^\nu + \cdots$$

なる漸近展開を持つ。円周率の計算で関孝和がエイトケン Δ^2 法を用い、建部賢弘がリチャードソン補外を用いそれぞれ加速に成功した理由は、部分列 $\{s_{2^\nu}\}$ が幾何学的漸近展開を持つことによる。

複合台形公式の誤差は分割数の逆数のオーダーであるので、分割数を 2 倍 2 倍と増やしていくと幾何学的収束になる。ロンベルク積分法はこのことに基づきリチャードソン補外を適用したものである。

7 漸近展開の利用

対数収束数列の漸近列と係数が両方とも既知のときは、5 節の補正による加速が適用できる。(漸近列については 2008 年 6 月号を見よ。) 漸近列が既知であれば、係数が未知であっても前回話した E 算法により加速することができる。

$s_\nu = \sum_{i=1}^{\nu} 1/i^2$ の漸近列は補題 1 より、

$$\nu^{-1}, \nu^{-2}, \nu^{-3}, \nu^{-5}, \nu^{-7}, \nu^{-9}, \ldots$$

がとれるので、

$$g_j(\nu+l) = \begin{cases} (\nu+l)^{-j} & j = 1,2,3 \\ (\nu+l)^{3-2j} & j \ge 4 \end{cases} \tag{7}$$

とおくと E 算法が適用できる。結果は表 3 のようである。以下の表 3-5 では倍精度 (10 進 16 桁弱) で計算し小数第 12 位を四捨五入し表示している。ただし、最良の値は小数第 15 位まで示した。

表 3: $\sum_{i=1}^{\nu} 1/i^2$ に E 算法を適用

ν	$E_0^{(\nu)}$	$E_1^{(\nu-1)}$	$E_2^{(\nu-2)}$
1	1.00000000000		
2	1.25000000000	1.50000000000	
3	1.36111111111	1.58333333333	1.62500000000
4	1.42361111111	1.61111111111	1.63888888889
5	1.46361111111	1.62361111111	1.64236111111
10	1.54976773117	1.63976773117	1.64470600277
ν	$E_3^{(\nu-3)}$	$E_4^{(\nu-4)}$	$E_5^{(\nu-5)}$
4	1.64351851852		
5	1.64467592593	1.64483855434	
10	1.64493105421	1.64493397472	1.64493406098
ν	$E_6^{(\nu-6)}$	$E_7^{(\nu-7)}$	$E_8^{(\nu-8)}$
10	1.64493406611	1.64493406667	1.64493406677
ν	$E_9^{(\nu-9)}$		
10	1.644934066785491		

10 項までの部分和から $E_9^{(1)} = 1.644934066785491$ が得られ、$\pi^2/6$ と 10.42 桁一致している。

8 数列変換の利用

対数収束数列の加速に適した数列変換は各種開発されている。ここでは、レヴィン u 変換とウィンの ρ 算法を取り上げる。

8.1 レヴィン u 変換

レヴィン変換は、E 算法において

$$g_j(\nu+l) = R_{\nu+l}(\nu+l)^{1-j}$$

としたものである (2008 年 11 月号)。ここで、$R_\nu = \nu\Delta s_{\nu-1} = \nu(s_\nu - s_{\nu-1})$ とおくのが u 変換、

$$R_\nu = \frac{\Delta s_\nu \Delta s_{\nu-1}}{\Delta^2 s_{\nu-1}} = \frac{(s_{\nu+1}-s_\nu)(s_\nu - s_{\nu-1})}{s_{\nu+1} - 2s_\nu + s_{\nu-1}}$$

とおくのが v 変換、$R_\nu = \Delta s_\nu = s_{\nu+1} - s_\nu$ とおくのが t 変換である。

$s_\nu = \sum_{i=1}^{\nu} 1/i^2$ にレヴィン u 変換を適用する。$R_\nu = 1/\nu^2$ なので、$j = 1, 2 \ldots$ に対し

$$g_j(\nu + l) = (\nu + l)(\nu + l)^{-2}(\nu + l)^{1-j} = (\nu + l)^{-j}$$

である。($j = 1, 2, 3$ の場合は、(7) に一致するので $T_k^{(\nu)}, k = 1, 2, 3$ は表 3 に一致する。なお、このようなことが生じるのは $s_\nu = \sum_{i=1}^{\nu} 1/i^2$ のみである。)

表 4: $\sum_{i=1}^{\nu} 1/i^2$ にレヴィン u 変換を適用

ν	$T_0^{(\nu)}$	$T_1^{(\nu-1)}$	$T_2^{(\nu-2)}$
1	1.00000000000		
2	1.25000000000	1.50000000000	
3	1.36111111111	1.58333333333	1.62500000000
4	1.42361111111	1.61111111111	1.63888888889
5	1.46361111111	1.62361111111	1.64236111111
10	1.54976773117	1.63976773117	1.64470600277
ν	$T_3^{(\nu-3)}$	$T_4^{(\nu-4)}$	$T_5^{(\nu-5)}$
4	1.64351851852		
5	1.64467592593	1.64496527778	
10	1.64493105421	1.64493499097	1.64493417081
ν	$T_6^{(\nu-6)}$	$T_7^{(\nu-7)}$	$T_8^{(\nu-8)}$
10	1.64493405358	1.644934057048	1.64493406373
ν	$T_9^{(\nu-9)}$		
10	1.644934066239456		

10 項までの部分和から $T_9^{(1)} = 1.644934066239456$ が得られ、$\pi^2/6$ と 9.43 桁一致している。

8.2 ρ 算法

D.A. スミスと W.F. フォードが加速法の比較を行ったことは前回 (2008 年 11 月号) 話した。彼らは 8 種類の対数収束級数を取り上げた。そのうち 6 個の級数に対してはウィンの ρ 算法が最良の結果を与えたが、残り 2 つに対しては ρ 算法は加速に失敗した。8 種すべてに対してはレヴィン u 変換が最良であった。

この結果についてスミスとフォードは、「ρ 算法は 8 つの問題のうち 6 つの問題では、u 変換や θ 算法よりよい実行結果である。(2 つで失敗したという) 信頼性のなさと相殺してもわずかな利点である。」とした。

ρ 算法は P. ウィン [13] が ϵ 算法と同時期に提案した収束の加速法である。ϵ 算法に類似の菱形算法である。数列 $\{s_\nu\}$ に対する ρ 算法は

$$\begin{aligned} \rho_{-1}^{(\nu)} &= 0 \\ \rho_0^{(\nu)} &= s_\nu \\ \rho_k^{(\nu)} &= \rho_{k-2}^{(\nu+1)} + \frac{k}{\rho_{k-1}^{(\nu+1)} - \rho_{k-1}^{(\nu)}}, \\ & k = 1, 2, \ldots, \end{aligned} \tag{8}$$

により定義される。

ρ 算法を $s_\nu = \sum_{i=1}^{\nu} 1/i^2$ に適用した結果は表 5 のようになる。

表 5: $\sum_{i=1}^{\nu} 1/i^2$ に ρ 算法を適用

ν	$\rho_0^{(\nu)}$	$\rho_2^{(\nu-2)}$	$\rho_4^{(\nu-4)}$
1	1.00000000000		
2	1.25000000000		
3	1.36111111111	1.65000000000	
4	1.42361111111	1.64682539683	
5	1.46361111111	1.64583333333	1.64489489489
10	1.54976773117	1.64503088906	1.64493357289
ν	$\rho_6^{(\nu-6)}$	$\rho_8^{(\nu-8)}$	
10	1.64493407610	1.644934066277279	

$\rho_8^{(2)} = 1.644934066277279$ は $\pi^2/6$ と 9.46 桁一致しており、レヴィン u 変換と同程度である。この例に見られるように、ρ 算法は $\sum_{i=1}^{\nu} 1/i^2$ などに効率よく働く加速法であるが、あまり利用されてない。最大の理由は発表以来 30 数年にわたり収束定理がなかったことによると思われる。あるいは規則性が分からなかったということかもしれない。ρ 算法の収束定理は筆者により与えられた。

定理 1 (長田 [6]) 漸近展開

$$s_\nu \sim s + \nu^{-1}\left(c_0 + \frac{c_1}{\nu} + \frac{c_2}{\nu^2} + \cdots\right)$$

を持つ数列 $\{s_\nu\}$ に ρ 算法を適用したとき、

$$\rho_{2k}^{(\nu)} = s + O((\nu + k)^{-1-2k}), \quad \nu \to \infty$$

が成立する。

証明 略 □

定理 2 (長田 [7]) 漸近展開

$$s_\nu \sim s + \nu^\theta\left(c_0 + \frac{c_1}{\nu} + \frac{c_2}{\nu^2} + \cdots\right), \quad \mathrm{Re}\,\theta < 0 \tag{9}$$

を持つ数列 $\{s_\nu\}$ に ρ 算法を適用したとき、

$$\lim_{\nu\to\infty} \frac{\rho_{2k}^{(\nu)} - s}{s_{\nu+2k} - s} = 0$$

となるための必要十分条件は、θ が負の整数のときである。

証明 略 $\square$

スミスとフォードの問題で ρ 算法が失敗した 2 題の問題はいずれも (9) の θ が整数ではなかった。収束定理が得られた後も ρ 算法の利用は増えてないようである。

http://www.cis.twcu.ac.jp/~osada/rikei/ に ρ 算法の C 言語によるプログラムをおいておく。

9 加速不可能性

線型収束する数列はエイトケン Δ^2 法で加速することができる。対数収束する数列を１つの数列変換で加速することはできないだろうか。これについては、J.P. ドゥラエと B. ジェルマンボンによって否定的に解決されている。これらのことは１冊のテキスト [2] にまとめられている。

J.P. ドゥラエは数列の集合に残存性という概念を導入し、残存性をもつ数列の集合に属す数列全体を１つのアルゴリズムで加速することはできないということを証明した [1]。

そのことから次の定理が導かれる。

定理 3 (J.P. ドゥラエ・B. ジェルマンボン [3]) 対数収束する数列全体の集合 LOG は１つのアルゴリズムで加速することはできない。

証明 日本語で書かれたものに [11] がある。 $\square$

10 対数収束数列の加速 — まとめ

今回はバーゼル問題の級数を巡る加速法の話をした。10 項までの部分和からそれぞれの加速法で $\pi^2/6$ と何桁一致するかをまとめておく。補正による方法以外の計算は倍精度 (10 進 16 桁弱) によっている。

表 6: 級数 $\sum_{i=1}^{\nu} 1/i^2$ に対する加速法の比較

方法	データ	一致桁数
オイラーの計算 (補正)	表 2	17.41
漸近列による E 算法	表 3	10.42
レヴィン u 変換	表 4	9.43
ρ 算法	表 5	9.46

漸近展開した際の、漸近列と係数の両方のデータを用いたオイラーの計算 (補正) が最も精度が良く、漸近列のデータを用いた E 算法がそれに続く。$s_1, \ldots, s_{10}$ の値のみを用いた ρ 算法、レヴィン u 変換は 1 桁程度劣る。といっても、スミスとフォードの比較テストに取り上げられた他の 3 つの算法に比べれば、格段に優れている。

対数収束数列に対しては万能の加速法は存在しないので、与えられた数列の漸近展開を考え、適切な加速法を選ぶのが良い。対数収束数列には、漸近列に $(\log \nu)^p/\nu^q, (p, q \in \mathbb{N})$ などの項を持つものもある。これらに対してはレヴィン u 変換や ρ 算法は無力であるが、レヴィン変換を拡張した変換 GREP で加速できることがある。詳細は [9] を見よ。

11 加速法を終えるにあたって

前編、中編、中編の 2、後編と 4 回にわたった加速法の話はこれで終える。数値計算において加速法を選択するポイントを挙げておく。

1. 反復が 1 次収束より高次数のときは、加速法は必要ない。
2. 線型収束数列 $\{s_\nu\}$ が漸近展開
$$s_\nu = s + \sum_{j=1}^{\infty} c_j \lambda_j^\nu, \quad \lambda_j \neq 1 (j = 1, 2, \ldots) \quad (10)$$
を満たすとき、
   (a) $c_1, c_2, \ldots; \lambda_1, \lambda_2, \ldots$ が既知の場合、補正による加速。
   (b) $\lambda_1, \lambda_2, \ldots$ が既知の場合、リチャードソン補外による加速。
   (c) $\lambda_1, \lambda_2, \ldots$ が未知の場合、ϵ 算法による加速。
3. 線型収束であるが漸近展開が未知の場合は、エイトケン Δ^2 法、レヴィン u 変換による加速。
4. 交代級数の場合は、エイトケン Δ^2 法、レヴィン u, t 変換による加速。
5. 対数収束数列 $\{s_\nu\}$ が漸近展開
$$s_\nu \sim s + \sum_{j=1}^{\infty} c_j g_j(\nu), \quad \lim_{\nu \to \infty} \frac{g_1(\nu + 1)}{g_1(\nu)} = 1$$
を満たすとき、

(a) c_j と $g_j(\nu)$ がともに既知の場合、補正による加速。

(b) $g_j(\nu)$ が既知の場合、一般化リチャードソン補外 (E 算法) による加速。

- $g_j(\nu) = \nu^{-j}$ のときは ρ 算法、レヴィン u 変換による加速。
- $g_j(\nu) = \nu^{-mj}, m \in \mathbb{N}$ のときは、部分列 $\{s_{2^\nu}\}$ を加速。

(c) $g_j(\nu)$ が未知の場合は難しい。

参考文献

[1] J.P. Delahaye, Algorithmes pour suites non convergentes, Numer. Math. 34(1980), 333–347.

[2] J.P. Delahaye, Sequence Transformation, Springer, 1988

[3] J.P. Delahaye and B. Germain-Bonne, The set of logarithmically convergent sequences cannot be acceleratrd, SIAM J. Numer. Anal. 19(1982), 840–844.

[4] L. Euler, De summis serierum reciprocarum (逆数の級数の和について), Commentarii academiae scientiarum Petropolitanae 7(1740), 123-134 `http://www.math.dartmouth.edu/~euler/pages/E41.html`

[5] L. Euler, Institutiones calculi differentialis cum eius usu in analysi finitorum ac doctrina serierum(微分学教程), 1755 `http://www.math.dartmouth.edu/~euler/pages/E212.html`

[6] N. Osada, A convergence acceleration method for some logarithmically convergent sequences, SIAM J. Numer. Anal 27(1990), 178–189.

[7] N. Osada, An acceleration theorem for the ρ-algorithm, Numer. Math. 73(1996), 521–531.

[8] D.J. Pengelley, Dances between continuous and discrete: Euler's summation formula, in R.E. Bradley et al. Euler at 300, MAA, 2007.

[9] A. Sidi, Practical Extrapolation methods, Cambridge, 2003.

[10] D.A. Smith and W.F. Ford, Acceleration of linear and logarithmic convergence, SIAM J. Numer. Anal. 16(1979), 223–240.

[11] 杉原正顯・室田一雄、数値計算法の数理、岩波書店、1994

[12] G. Walz, Asymptotics and Extrapolation, Akademie Verlag, 1966

[13] P. Wynn, On a Procrustean technique for the numerical transformation of slowly convergent sqeuences and series, Proc. Camb. Phil. Soc., 52(1956), 663–671.

(おさだ なおき/東京女子大学)