

関数(数学)

白 清水

平成30年5月7日

数学における関数（かんすう、英: *function*、仏: *application*、独: *Funktion*、羅: *functio*、函数とも）とは、かつては、ある変数に依存して決まる値あるいはその対応を表す式の事であった。この言葉はライプニッツによって導入された。その後定義が一般化されて行き、現代的には数の集合に値をとる写像の一種であると理解される。

# 目 次

第 1 章	関数 ( 数学 )	3
1.1	表記の歴史 . . . . .	3
1.2	定義 . . . . .	4
1.3	現代的解釈 . . . . .	4
1.4	ブラックボックスモデルによる説明 . . . . .	5
1.5	関数の例 . . . . .	5
1.6	多変数関数と多価関数 . . . . .	6
1.7	陽表式と陰伏式 . . . . .	7
1.8	一般化 . . . . .	8
1.8.1	数列 . . . . .	8
1.8.2	汎関数 . . . . .	8
1.8.3	超関数 . . . . .	9

# 第1章 関数（数学）

## 1.1 表記の歴史

日本語としての関数はもともと「関数」（旧字体では函数）<sup>t</sup>と書く。函数という言葉は中国語から輸入されたものであり、中国での初出は1859年に出版された李善蘭の『代微積拾級』といわれる。

微積分について日本語で書かれた最初の本、花井静校・福田半編『筆算微積入門』（1880年）では「函数」が用いられている [1][2]。それに続く長澤龜之助訳『微分学』（1881年）、岡本則録訳『查氏微分積分学』（1883年）のいずれも用語を『代微積拾級』、『微積溯源』（1874年）などによっている [2]。明治初期に東京数學會社で数学用語の日本語訳を検討する譯語會が毎月開催され、その結果が『東京数學會社雑誌』で逐次報告されている。この報告に function の訳語は第 62 号（1884 年）の「原數」[3]と第 64 号（1884 年）の「三角法函数」[4]の二種類が登場する。一方、同誌の本文では 61 号（1884 年）や 63 号（1884 年）で「函数」が用いられている [5]。

「函」が漢字制限による当用漢字に含まれなかったことから、1950 年代以降同音の「関」へと書き換えがすすめられた [6]。この他、「干数」案もあった [7]。学習指導要領に「関数」が登場するのは中学校で 1958 年、高等学校で 1960 年であり、それまでは「函数」が用いられている [8]。「関数」表記は 1985 年頃までには日本の初等教育の段階でほぼ定着した [9]。

「函数」の中国語における発音は（ピン音: *hanshu*）であり、志賀浩二や小松勇作によればこれは function の音訳であるという [9][10]。一方、『代微積拾級』には「凡此變數中函彼變數則此為彼之函数」[11]とあり、また変数に天、地などの文字を用いて「天 = 函(地)」という表記もある。片野善一郎によれば、「函」の字義はつつむ、つつみこむであるから、「天 = 函(地)」という表現は「天は地を函む」ようにみえ [1]、従属変数（の表現）に独立変数が容れられている [2] という意味であるという。

なお、現代の初等教育の場においてはしばしば関数をブラックボックスのたとえで説明することがある [2][12][13]。この説明では、「函」を「はこ」と読むことと関連付けて説明されることもあるが、「函数」の語の初出は 1859 年なのに対し、「ブラックボックス」の語の初出は 1945 年ごろとされることに注意を要する。

## 1.2 定義

二つの変数  $x$  と  $y$  があり、入力  $x$  に対して、出力  $y$  の値を決定する規則  $x$  に特定の値を代入するごとに  $y$  の値が確定する) が与えられているとき、変数  $y$  を「 $x$  を独立変数 (independent variable) とする関数」或いは簡単に「 $x$  の関数」という。対応規則を明示するときは、適当な文字列 (特に何か理由がなければ、function の頭文字から  $f$  が選ばれることが多い) を使って

$$y = f(x) \quad (1.1)$$

のように対応規則に名前を付与する。 $x$  の関数  $y$  を  $f(x)$  と書いて、 $x = a$  を代入したときに決まる関数の値を  $f(a)$  と表すのである。

しかしここで、定数関数の例に示されるように、個々の  $y$  の値について対応する  $x$  の値が一つに決まるとは限らないことに注意しなければならない。この  $f(x)$  という表記法は 18 世紀の数学者オイラーによるものである。オイラー自身は、変数や定数を組み合わせてできた数式の事を関数と定義していたが、コーシーは上に述べたように、 $y$  という変数を関数と定義した。

$y$  が  $x$  の関数であることの別の表現として、変数  $y$  は変数  $x$  に従属するとも言い、 $y$  を従属変数 (dependent variable) と言い表す。独立変数が取り得る値の全体 (変域) を、この関数の定義域 (domain) と言い、独立変数が定義域のあらゆる値を取るときに、従属変数が取り得る値 (変域) を、この関数の値域 (range) という。

関数の終域は実数  $\mathbb{R}$  や複素数  $\mathbb{C}$  の部分集合であることが多い。終域が実数の集合となる関数を実数値関数 (real valued function) といい、終域が複素数の集合となる関数を複素数値関数 (complex valued function) という。それぞれ定義域がどのような集合であるかは問わないが、定義域も終域も実数の集合であるような関数を実関数 (real function) といい、定義域も終域も複素数の集合であるような関数を複素関数 (complex function) という。

## 1.3 現代的解釈

ディリクレは、 $x$  と  $f(x)$  の対応関係に対して一定の法則性を持たせる必要はないとした。つまり、個々の独立変数と従属変数の対応そのものが関数であり、その対応は数式などで表す必要はないという、オイラーとは異なる立場をとっている。集合論的立場に立つ現代数学では、ディリクレのように関数を対応規則  $f$  のことであると解釈する。それは二項関係の特別の場合として関数を定義するということであり、関数を集合から「数」のつくる集合への写像であると捉えると言う事である [14]。[要追加記述] よって、写像に用いる言葉をそのまま流用する事がある。

- 合成 (合成関数)
- 全射、単射 (一対一ともいう)、全単射 (双射、一対一対応とも言う)
- 逆 (逆関数)

などを挙げることができる。一方で、「数」に値を取る関数は一般の写像とは異なる性質を持つ。たとえば、像を用いて値毎の演算と呼ばれる函数同士の演算が定義できること： $x$  を任意として、

- $(f + g)(x) := f(x) + g(x),$
- $(f - g)(x) := f(x) - g(x),$
- $(fg)(x) := f(x)g(x),$
- $(f/g)(x) := f(x)/g(x),$

などが挙げられる。

## 1.4 ブラックボックスモデルによる説明

とくに教育において関数はブラックボックスで説明されることがあると前に述べたが関数は入力と出力が定義されたブラックボックスとして説明される [12]。集合論的立場に立つ現代数学では、ディリクレのように関数を対応規則  $f$  のことであると解釈する。それは二項関係の特別の場合として関数を定義するということであり、関数を集合から「数」のつくる集合への写像であると捉えると言う事である [14]。[要追加記述] よって、写像に用いる言葉をそのまま流用する事がある。

## 1.5 関数の例

- 一次関数： $f(x) = ax + b$  ( $a, b$  は定数,  $a \neq 0$ )
  - とくに、 $b=0$  のとき線形写像,  $a=1$  かつ  $b=0$  のとき恒等関数 (恒等写像、identity) になる
- 二次関数： $f(x) = ax^2 + bx + c$  ( $a, b, c$  は定数で,  $a \neq 0$ )
- 指示関数：

$$\chi_A(x) = \begin{cases} 1 & (x \in A), \\ 0 & (x \notin A). \end{cases} \quad (1.2)$$

以下に代表的な関数とその具体例の一覧表を掲げる [12][15]。すべての物を網羅しているわけではないことに注意されたい。

関数				具体例
代数関数	有理関数	定数関数	多項式関数	$f(x) = a$
			一次関数	$f(x) = ax + b$
			二次関数	$f(x) = ax^2 + bx + c$
			三次関数	$f(x) = ax^3 + bx^2 + cx + d$
		分数関数		$f(x) = \frac{a}{x}$
	無理関数		$f(x) = \sqrt{x}$	
初等関数	指数関数		$a^x, e^x, 2^x$	
	対数関数		$\log(x), \ln(x), \log_a(x)$	
	三角関数		$\sin(x), \cos(x), \tan(x)$	
	逆三角関数		$\sin^{-1}(x), \cos^{-1}(x), \tan^{-1}(x)$	
	双曲線関数		$\sinh(x), \cosh(x), \tan H(x)$	
特殊関数	ガンマ関数		$\Gamma(x)$	
	ベータ関数		$B(x, y)$	
	誤差関数		$erf(x)$	
	テータ関数			
	ゼータ関数		$\zeta(x)$	
	マチウ関数			

## 1.6 多変数関数と多価関数

複数の変数によって値が決定される関数を多変数関数と言う。これは複数の数の集合たちの直積集合から数の集合への写像であると解釈される。ベクトルの集合を定義域とする独立変数をもつ関数と解釈することもある。n 個の変数で決まる関数であれば、n 変数関数とも呼ばれ

$$y = f(x_1, x_2, \dots, x_n)$$

のように書かれる。例えば

$$y = x_1^2 + x_2^2$$

は二変数関数である。

一つの入力に複数の出力を返すような対応規則を関数の仲間として捉えるとき多価関数 (multi-valued function) と言う。常に n 個の出力を得る関数は n 価であるといい、その n を多価関数の価数と呼ぶ。例えば正の実数にその平方根を与える操作は正と負の二つ値を持つので、二価関数である。多価関数に対し、普通の一つの値しか返さない関数は一価関数といわれる。

多変数関数は独立変数がベクトルに値をとるものと解釈できるということを上に述べたが、逆に従属変数がベクトルの値を持つような写像も考えられ、それをベクトル値関数という。ベクトル値関数が与えられたとき、像のベクトルに対してその各成分をとり出す写像を合成してやることにより、通常の一価関数が複数得られる。つまり、定義域を共有するいくつかの関数を一つのベクトルとしてま

とめて扱ったものがベクトル値関数であるということができる。

一つの例として、実数全体  $\mathbb{R}$  で定義された二価の関数

$$f(x) = \pm\sqrt{1+x^2}$$

はベクトル値関数

$$f: \mathbb{R} \rightarrow \mathbb{R}^2; x \rightarrow f(x) = (\sqrt{1+x^2}, -\sqrt{1+x^2})$$

として扱うことができる。また、定義域の ”コピー” を作って定義域を広げてやることで、その拡張された定義域上の一価の関数

$$f: A \sqcup B \rightarrow \mathbb{R} (A = B = \mathbb{R})$$

$$f(x) = \begin{cases} -\sqrt{1+x^2} & x \in A \\ \sqrt{1+x^2} & x \in B \end{cases}$$

と見なすこともある。複素変数の対数関数  $\log$  は素朴には無限多価関数であるが、これを  $\log$  のリーマン面上の一価関数と見なすなど、定義域を広げて一価にする手法は解析的な関数に対してしばしば用いられる。

## 1.7 陽表式と陰伏式

多変数方程式がいくつかの関数関係を定義することもある。例えば

$$F(x, y) = 0$$

のような式が与えられているとき、 $x$  と  $y$  は独立に別々の値をとることはできない。 $x$  に勝手な値を与えるならば、 $y$  は  $x$  の値のよってとりうる値の制約を受けるからである。このことを以って、独立変数  $x$  と従属変数  $y$  が対応付けられると考えるとき、方程式  $F(x, y) = 0$  は  $x$  の関数  $y$  を陰 (implicit) に定めるといい、 $y$  を  $x$  の陰伏関数または陰関数 (implicit function) という。これに対して、 $y = f(x)$  と表されるような関数関係を、 $y$  は  $x$  の陽関数 (explicit function) である、あるいは  $y$  は  $x$  で陽 (explicit) に表されているなどと言い表す。

陰伏的な関数関係が  $F(x, y) = 0$  によって与えられていて、陽な関数関係  $y = f(x)$  が適当な集合  $D$  を定義域として  $F(x, f(x)) = 0$  を満たすなら、この陽関数  $y = f(x)$  は  $D$  上で関係式  $F(x, y) = 0$  から陰伏的に得られるという。関数の概念を広くとらず、一価で連続である場合や一価正則な場合などに考察を限ることはしばしば行われることであるが、そのような仮定のもとでは陰関数から陰伏的に得られる陽関数は一つとは限らず、一般に一つの陰関数は (定義域や値域でより分けることにより) 複数の陽関数に分解される。このとき、陰伏的に得られた個々の陽関数をもとの陰関数の枝という。また、陰関数の複数の枝を総じて扱うならば、陰関数の概念から多価関数の概念を得ることになる。例えば、方程式

$$y^2 - x^2 = 1$$

が定める陰関数  $y$  は全域で 2 つの一価連続な枝

$$f_1(x) = \sqrt{1+x^2},$$



$$f_2(x) = -\sqrt{1+x^2}$$

をもつ二価関数である。

また、媒介変数を導入して関係式を分解し、各変数を媒介変数の陽関数として表すことによって、陰関数を表すこともある。例えば、方程式  $2x^2 + 3y = 0$  は、媒介変数  $t$  を導入して

$$\begin{cases} x = 3t \\ y = 2t \end{cases}$$

と表すことができるが、これによって  $y$  と  $x$  の陰伏的な関数関係が表されていると考えるのである。

## 1.8 一般化

### 1.8.1 数列

有限集合からの関数は実質的に数の組あるいは数列と呼ばれる物になる（適当な演算をいれてベクトルと見ることもしできる）。それはつまり、集合の各元に数列を与えて  $1, 2, \dots, n$  と並べるとき、 $k = 1, 2, \dots, n$  に対して  $x_k = x(k)$  を対応付ける関数  $x$  を

$$(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$$

のかたちに表すのである。これは有限列であるが、無限列

$$(s_n)_{n \in \mathbb{N}} \in \mathbb{R}^{\mathbb{N}}$$

を考えれば、それは各自然数  $n$  に対して、数  $s_n$  を対応させる

$$s : \mathbb{N} \rightarrow \mathbb{R}; n \mapsto s_n$$

という関数を考えていることに他ならない。もっと一般に数の族を考慮に入れば、通常の実関数  $f = f(x)$  を  $x$  を添字に持つ実数の族

$$(f_x)_{x \in \mathbb{R}} \in \mathbb{R}^{\mathbb{R}}$$

と読み替えることができる。

### 1.8.2 汎関数

詳細は「汎関数」を参照

関数を変数に取る関数はとくに汎関数 (functional) と呼ばれる。特にある集合上の関数の作るベクトル空間から係数体への線型写像を線型汎関数 (linear functional) という。文脈によっては単に汎関数といえば線型汎関数を指すこともある。たとえば積分

$$F(f) = \int_{-\infty}^{\infty} f(x) dx$$

は可積分関数  $f$  を変数と見なして様々に取り替えることによって汎関数  $F$  を与える。積分は線型性を持つから、 $F$  は線型汎関数である。

有限個の変数の組を考えることも関数の一種であったから、汎関数

$$\mathcal{F}(f) = \mathcal{F}(f(x))$$

はひとつまたは複数のパラメータで添字付けられる一般には無限個の変数をもつ関数の一種

$$\mathcal{F}((f_x)_{x \in \mathbb{R}})$$

と見なすことができる。また、有限次元ベクトル空間は基底を固定することにより、その座標で表される係数体の有限個の直積と同型であるから、そこからの汎関数は多変数関数

$$F(x_1, x_2, \dots, x_n)$$

と同一視できる。

関数に対して数に対応付けるという汎関数の概念は、さらに関数に関数に対応付ける作用素の概念に一般化される。

### 1.8.3 超関数

詳細は「超関数」を参照

シュワルツの超関数（分布、英: distribution）の理論は、汎関数の一種（コンパクトな台を持つ無限階微分可能関数の作る空間上の連続線型汎関数）として超関数を定義する。通常の局所可積分関数に測度を掛けて積分作用素として見ると、この意味で超関数と見なされる。

この様な連続線型汎関数を用いた定式化の方向で distribution よりも大きいクラスとしては、超分布 (ultradistribution) が知られている。

一方、佐藤の超関数（英: hyperfunction）は層係数コホモロジー等の代数的手法を用いて定義される。この代数的手法の解析学への導入により、線型微分方程式系の代数化である  $D$  加群の理論等、代数解析学と呼ばれる分野が開かれた。以上の超関数のクラスは局所化可能、言い換えれば層を成すという事が重要である。