

算数数学教育の暗黒面

黒木玄 (Gen Kuroki)

2018-08-21

- Copyright 2018 Gen Kuroki
- License: MIT <https://opensource.org/licenses/MIT> (<https://opensource.org/licenses/MIT>)
- Repository: <https://github.com/genkuroki/HighSchoolMath>
(<https://github.com/genkuroki/HighSchoolMath>)

このファイルは次の場所できれいに閲覧できる:

- [高校数学の話題 HTML版](http://nbviewer.jupyter.org/github/genkuroki/HighSchoolMath/blob/master/MathEduDarkSide.ipynb)
(<http://nbviewer.jupyter.org/github/genkuroki/HighSchoolMath/blob/master/MathEduDarkSide.ipynb>)
- [高校数学の話題 PDF版](https://genkuroki.github.io/documents/HighSchoolMath/MathEduDarkSide.pdf) (<https://genkuroki.github.io/documents/HighSchoolMath/MathEduDarkSide.pdf>)

このノートの想定読者は大学である程度を数学を学んだ学生で高校で習った数学について見直したい人達である.

このファイルはJulia言語 (<https://julialang.org/>)カーネルの [Jupyter notebook](http://jupyter.org/) (<http://jupyter.org/>) である. 自分のパソコンにJulia言語 (<https://julialang.org/>)をインストールしたい場合には

- [WindowsへのJulia言語のインストール](http://nbviewer.jupyter.org/gist/genkuroki/81de23edcae631a995e19a2ecf946a4f)
(<http://nbviewer.jupyter.org/gist/genkuroki/81de23edcae631a995e19a2ecf946a4f>)

を参照せよ. このファイルは [JuliaBox](https://juliabox.com/) (<https://juliabox.com/>) でも使用できるかもしれない. このファイル中のJulia言語 (<https://julialang.org/>)のコードを理解できれば, Julia言語 (<https://julialang.org/>)からSymPy (<https://www.sympy.org>)を用いた数式処理や数値計算の結果のプロットの仕方を学ぶことができる.

目次

- [1 微分は分数商ではないのか？](#)
- [2 高校数学における三角関数の微積分は循環論法なのか？](#)
- [3 無理式とは根号内に文字を含む式のことなのか？](#)
- [4 単項式は多項式ではないのか？](#)
- [5 等式は方程式と恒等式に分類されるのか？](#)
- [6 掛算には順序があるのか？](#)
- [7 ゼロは倍数ではないのか？](#)
- [8 必要条件は必要十分条件ではないのか？](#)
- [9 文字を含む式は「操作の方法」でかつ「操作の結果」なのか？](#)
- [10 括弧は1つの数量を表すための記号なのか？](#)

In [1]:

```

1  using Plots
2  pyplot()
3  ▼ #gr(); ENV["PLOTS_TEST"] = "true"
4  #clibrary(:colorcet)
5  #clibrary(:misc)
6
7  function pngplot(P...; kwargs...)
8      sleep(0.1)
9      pngfile = tempname() * ".png"
10     savefig(plot(P...; kwargs...), pngfile)
11     showing("image/png", pngfile)
12 end
13 pngplot(; kwargs...) = pngplot(plot!(); kwargs...)
14
15 showing(mime, fn; scale="") = open(fn) do f
16     base64 = base64encode(f)
17     if scale == ""
18         display("text/html", """""")
19     else
20         display("text/html", """""")
21     end
22 end
23
24 using SymPy
25 ▼ #sympy[:init_printing](order="lex") # default
26 ▼ #sympy[:init_printing](order="rev-lex")
27
28 using SpecialFunctions
29 using QuadGK
30 using Elliptic.Jacobi: cd, sn

```

1 微分は分数商ではないのか？

一部に

$$\frac{dy}{dx}$$

は $dy \div dx$ の意味ではないので、 dx 分の dy と読むはいけない。

と教えている先生がいるようだ。これは本当に正しいだろうか？ 微分形式を含む現代数学における**微分**のスタイルを知っていれば、**そのような教え方は誤りになるので注意しなければいけない**。

実際、高木貞治『[解析概論](https://www.google.co.jp/search?q=%E9%AB%98%E6%9C%A8%E8%B2%9E%E6%B2%BB+%E8%A7%A3%E6%9E%90%E6%A6%82%E8) (<https://www.google.co.jp/search?q=%E9%AB%98%E6%9C%A8%E8%B2%9E%E6%B2%BB+%E8%A7%A3%E6%9E%90%E6%A6%82%E8>)

には以下のように書いてある。

In [2]: 1 `showimg("image/jpeg", "kaisekigairon-bibun1.jpg")`

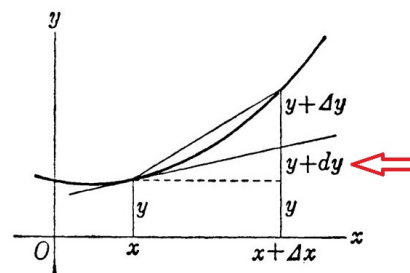
高木貞治『解析概論』p. 36より

函数 $y=f(x)$ のグラフにおいては, $\Delta y/\Delta x$ は点 (x, y) と点 $(x+\Delta x, y+\Delta y)$ とを結ぶ弦の勾配で, $\frac{dy}{dx}$ は点 (x, y) における接線の勾配である.

$\Delta x, \Delta y$ はグラフの上での点の座標の変動であるが, もしもグラフの代りに接線を取って, 接線上における点 (X, Y) の座標の変動を dx, dy で表わして, $dx=X-x$ (それは Δx と同一), また $dy=Y-y$ (それは Δy とは違う) とするならば

$$dy=f'(x)dx \quad (1)$$

は点 (x, y) における接線の方程式にほかならない. そのように dx, dy を単独に定義すれば, (1) の意味は明確である. しかし我々は点 (x, y) の近傍においてのみ (1) を用いるつもりであるから, dx を変数 x の微分 (differential), dy をそれに対応する函数 y の微分という.



In [3]: 1 `showimg("image/jpeg", "kaisekigairon-bibun2.jpg")`

高木貞治『解析概論』より

14. 微分の方法

37

$f'(x) \cdot \Delta x$ を点 x における函数 $y=f(x)$ の微分と名づけて, それを dy で表わすことにする. すなわちこの定義によれば

$$dy=f'(x)\Delta x. \quad (4)$$

今同様の意味において, x それ自身を x の函数とみれば, $x'=1$ だから

$$dx=\Delta x.$$

故に上記定義の下において, Δx は x の函数なる x の微分である. これを (4) に代入すれば,

$$dy=f'(x)dx. \quad (5)$$

これを

$$\frac{dy}{dx}=f'(x) \quad (6)$$

と書くならば, 記号 $\frac{dy}{dx}$ において dx および dy が各, 独立の意味を有するから, $\frac{dy}{dx}$ は商としての意味を有する. すなわち '微分商' というものである.

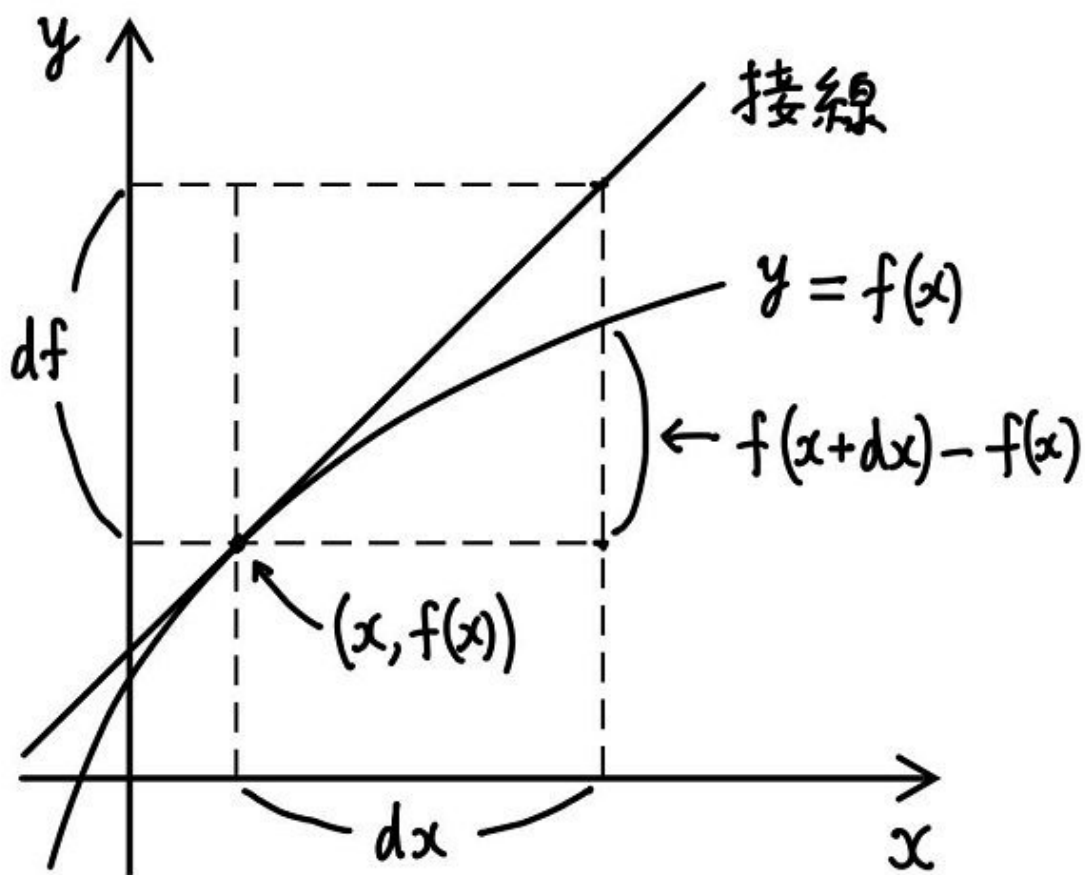
このように, 現代的の精密論法によって, Leibniz の漠然たる '微分商' が合理化される. また (5) によれば, $f'(x)$ は微分 dy における dx の係数であるから, それを '微分係数' というのも, もっともではある.

以上のように, 高木貞治『解析概論』には

$\frac{dy}{dx}$ は商として意味を有する.

とはっきり書いてある. 高木貞治氏による dx , dy の定義は次の図と同じである.

In [4]: 1 `showimg("image/jpeg", "bibun.jpg")`



$$(\text{接線の傾き}) = \frac{df}{dx} = df \div dx.$$

一般に $df \neq f(x+dx) - f(x)$.

dx は微小量でなくてもよい.

df の定義を $f(x+dx) - f(x)$ とするのではなく, 上の図のように取ることが, $\frac{df}{dx}$ を真の分数商とみなすときのポイントになる. 数学の定義はこのような「みもふたもない」ものが多い. この図の df の定義を一般化することによって1次の微分形式(differential form)が定義される. 微分形式は現代数学における基本的な概念である.

2 高校数学における三角関数の微積分は循環論法なのか?

答えはいいえである。

高校数学IIIの教科書には「曲線の長さを速さの積分で表す公式」が書いてある。その公式を用いて単位円弧の長さを定義することによって、弧度法の意味での角度を定義すれば循環論法にならないだけでなく、べき級数による天下りの定義やその他様々な技巧をこらした三角関数の再定義を行う必要もなくなり、高校数学のスタイルそのままの三角関数の定義に基き、さらに楕円関数論にも接続できるような三角関数論を展開できる。

数学的にはそういう事情になっているので、数学を本当に理解する気があるならば、「高校数学における三角関数の微積分は循環論法である」などと言ってはいけない。

実際の理論の展開の仕方の素描を「[高校数学の話題](https://github.com/genkuroki/HighSchoolMath) (<https://github.com/genkuroki/HighSchoolMath>)」の方に書いておいたので参照して欲しい。

3 無理式とは根号内に文字を含む式のことなのか？

高校の数学の教科書を見ると、

$\sqrt{x+1}$, $\sqrt{2x^2-3}$ などのように根号内に文字を含む式をその文字についての**無理式**といい、 x についての無理式で表された関数を x の**無理関数**という。(実教出版『数学III』2009年1月25発行)

のように書いてある。これは高校の数学の教科書外では通用しない可能性が高い定義である。

数については、 $\sqrt{2}$ だけではなく、 π のような超越数も無理数に含まれる。数と関数の類似に基けば、 $\sin x$ のような超越関数も無理関数と呼びたくなるのだが、上の定義に従うとそれは不可能になる。

さらに、無理式の定義がおそろしくあいまいである。 $\sqrt{\sin x}$ は無理式なのだろうか？

無理式(irrational expression)や無理関数(irrational function)は19世紀の数学の教科書に見付かる用語である。19世紀の時代遅れなスタイルが伝言ゲームによって21世紀の現代まで伝わっているのだろう。

上に引用したような定義になっていない「定義」は数学の本質と無関係である。数学を教えるときには、歴史的な経緯や数学とは無関係の「大人の事情」によって不適切な記述が教科書に残ってしまうことがある。

数学を教えるときには、教科書に忠実に従うことは決して**やってはいけないこと**である。

正しいか正しくないかを教科書に書いてあるかどうかで判断したり、「そう言っている人達がたくさんいるから」という理由で判断したりすることは非数学的な態度だと思われる。

4 単項式は多項式ではないのか？

非常に残念なことに、中学校と高校の数学の教科書における用語の体系は以下のようになっているように見える。

中学高校の数学教科書 現代的には常識的なスタイル

整式	多項式
単項式	単項式
多項式	複数の項を持つ多項式

現代ではほとんどの通信や放送がデジタル化されている。デジタル通信では情報の符号化とプライバシーを守るための暗号の技術が必要になる。それらの技術を理解するためには多項式環の理論も理解しておかなければいけない。そのような実用的な数学を学ぶためには、現代的には常識的なスタイルの用語法に従う必要がある。

このような事情になっているにもかかわらず、中学と高校の数学の教科書は19世紀以来の時代遅れのスタイルを採用してしまっている。

この点は改善されるべきなのだが、そのような改善が行われる目途は現時点ではまったくない。

5 等式は方程式と恒等式に分類されるのか？

これも答えはいいえである。

これもまた「19世紀の時代遅れのスタイルが中学高校での数学教育に残ってしまっている」という事実の一部分に過ぎない。

19世紀の教科書には、equations (等式という意味)を identities (恒等式)と conditional equations (条件によって成立したり成立しなかったりする等式、単に equation と書かれることも多い)に分けて説明するスタイルを採用しているものがあって、そのスタイルが日本語圏の中学高校の数学教育における「等式を恒等式と方程式に分けて説明するスタイル」として生き残っているものだと推測される。

あらゆる等式は単に両辺が等しいことを意味する式(記号列)に過ぎない。文脈によってニュアンスを変えた場合に恒等式と呼んだり、方程式と呼んだりするだけである。

数学を正しく理解するためにはそれらの曖昧な言葉に頼らずに「どのような条件のもとでどのような条件を満たす何を求めたいのか(何について考えているのか)」をクリアに考える必要がある。

例えば、 $ax + b = c$ を x に関する方程式とみなすときには、「 a, b, c が与えられているときに、等式 $ax + b = c$ を満たす x を求めること」を考えていることになる。

例えば、 $ax + b = c$ を x に関する恒等式とみなすときには、「すべての数 x について等式 $ax + b = c$ が成立するような a, b, c 」について考えていることになる。

「どのような条件のもとでどのような条件を満たす何を求めたいのか(何について考えているのか)」の普遍的なスタイルで常に考えるようにすれば19世紀由来の「等式を恒等式と方程式に分類するスタイル」は完全に忘れても大丈夫である。

6 掛算には順序があるのか？

7 ゼロは倍数ではないのか？

8 必要条件は必要十分条件ではないのか？

9 文字を含む式は「操作の方法」でかつ「操作の結果」なのか？

10 括弧は1つの数量を表すための記号なのか？

⏮ In []:	1	
-----------	---	--