「桁」の数学

ミノ～＋

2010 年 9 月 15 日

1　導入

ある数を「桁」という側面から探る。まず関数 $d_k^j(n)$ を定義する。（d は桁「digit」から）

d 関数の定義

自然数 n において $d_1^j(n)$（これ以降添え字の”1”は省略する）は各桁の数字を j 乗し、そのそれぞれを足す関数と定義する。
例:

$$\begin{aligned} d^2(321) &= 3^2 + 2^2 + 1^2 \\ &= 14 \end{aligned}$$

添え字 k に関しては、この d 関数の操作を k 回行うという意味である。厳密には

$$d_k^j(n) = d^j(d_{k-1}^j(n))$$

と帰納的に定義する。
例:

$$\begin{aligned} d_2^3(212) &= d^3(2^3 + 1^3 + 2^3) \\ &= d^3(17) \\ &= 1^3 + 7^3 \\ &= 344 \end{aligned}$$

2　コンピュータで計算させるための準備

前述の例のように、$d_2^3(212)$ を計算した。しかし、紙に実際書いてこの関数の性質を調べるにはとてもたいへんである。そこで (仮称) 十進 BASIC[*1]を用いて計算させてみたい。ここではその準備を行う。d 関数を計算するためには数 n から 1 桁目、2 桁目・・・の数字を取り出さなければな

[*1] フリーソフト　1000 桁モードなどのすばらしい機能がある数学に適したプログラミング言語。
http://hp.vector.co.jp/authors/VA008683/

らない。そこで BASIC の関数である MOD 関数を用いる。この関数は剰余を求める関数である。例：

$$\mathrm{MOD}(15,6)=3$$

これを用いれば、n=123 に対して $d(n)$,

$$\begin{aligned}
d(123) &= \mathrm{MOD}(123,10)+\frac{1}{10}(\mathrm{MOD}(123,100)-\mathrm{MOD}(123,10)) \\
&\quad +\frac{1}{100}(\mathrm{MOD}(123,1000)-\mathrm{MOD}(123,100)) \\
&= 3+\frac{1}{10}(23-3)+\frac{1}{100}(123-23) \\
&= 3+2+1=6
\end{aligned}$$

と計算できる。j 乗という要素を入れて数学的に書いてみると、

$$d^j(n)=\sum_{i=1}^{\infty}\Big\{\frac{1}{10^{i-1}}(\mathrm{MOD}(n,10^d)-\mathrm{MOD}(n,10^{d-1}))\Big\}^j \tag{1}$$

これより d 関数の定義

$$d_k^j(n)=\begin{cases}\displaystyle\sum_{i=1}^{\infty}\Big\{\frac{1}{10^{i-1}}(\mathrm{MOD}(n,10^d)-\mathrm{MOD}(n,10^{d-1}))\Big\}^j & (k=1)\\ d^j(d_{k-1}^j(n)) & (k\geq 2)\end{cases}$$

この式を参考にしてプログラミングする。FOR〜NEXT 構文を使って、k の要素を追加して BASIC でプログラミングすると

ソースコード 1　d 関数のプログラミング例

```
INPUT n , j , k
FOR l=1 TO k
    LET i=0
    FOR d=1 TO 1000
        LET i=i+((MOD(n,10^d)-MOD(n,10^(d-1)))/10^(d-1))^j
    NEXT d
    PRINT l , i
    LET n=i
NEXT l
END
```

このプログラムでは d 関数が k によって最終的な値になる過程も表示するようにした。また、1000 桁モードで使用するように書いたので、(1) 式中の ∞ は 1000 とした。（高速化ため適宜 1000 以下の数にしてももちろん良い。）計算結果が 1000 桁近くになる場合 ” 桁あふれ ” に注意して欲しい。

例：最初 i,j,k の値の入力を求められるので例えば「2,3,10」と入力すると、

1	8
2	512
3	134
4	92
5	737
6	713
7	371
8	371
9	371
10	371

$$\therefore d^3_{10}(2) = 371$$

ここで面白い結果を得た。すなわち、

$$371 = 3^3 + 7^3 + 1^3$$

調べてみると分かるが、このように k によって値が変化しないもの、ある周期でループするものが存在する。

MOD 関数

BASIC の MOD 関数について考えれば、ガウス関数[*2]を用いて

$$MOD(a,b) = a - b\left\lfloor \frac{a}{b} \right\rfloor$$

となることがわかる（実は MOD の定義式)。これを用いれば、

$$\begin{aligned} d^j(n) &= \sum_{i=1}^{\infty}\left\{\frac{1}{10^{i-1}}\left(\left(n - 10^i\left\lfloor\frac{n}{10^i}\right\rfloor\right) - \left(n - 10^{i-1}\left\lfloor\frac{n}{10^{i-1}}\right\rfloor\right)\right)\right\}^j \\ &= \sum_{i=1}^{\infty}\left(\left\lfloor\frac{n}{10^{i-1}}\right\rfloor - 10\left\lfloor\frac{n}{10^i}\right\rfloor\right)^j \end{aligned}$$

これより、より簡単な定義を得る。

d 関数の定義

$$d^j_k(n) = \begin{cases} \displaystyle\sum_{i=1}^{\infty}\left(\left\lfloor\frac{n}{10^{i-1}}\right\rfloor - 10\left\lfloor\frac{n}{10^i}\right\rfloor\right)^j & (k=1) \\ d^j(d^j_{k-1}(n)) & (k \geq 2) \end{cases}$$

この定義を用いて

[*2] 床関数とも言い、実数 x に対して x 以下の最大の整数として定義され, $\lfloor x \rfloor$、$[x]$ などと書かれる。同じような関数に天井関数がある。

ソースコード 2　d 関数のプログラミング例

```
INPUT n , j , k
FOR l=1 TO k
    LET i=0
    FOR d=1 TO 1000
        LET i=i+(INT(n/10^(d-1))-10*INT(n/10^d))^j
    NEXT d
    PRINT l , i
    LET n=i
NEXT l
END
```

を得る。

今後の研究

ここまで定義と d 関数のプログラミングについて考えてきた。ここまでで「桁」の数学を研究する準備が整ったわけである。今後は、ループする数はどのくらいありどのような数字なのか。また k に関して一定のなる数はどのくらい存在するのか（少し調べてみるとかなり少ないことがわかる）。どのくらいの周期でループするのか・・・などなど考えればたくさんある。少しずつ調べながら一般化し、証明していこうと思う。

あとがき

この「「桁」の数学」を書いているとき磯山雅著「バッハ＝魂のエヴァンゲリスト」を読んでいて p.276 にちょっと偶然とは思えない記述を発見した。以下に引用する。

> まず第一に、当時のルター正統派神学において、数のもつ意味の解釈がさかんだったことがあげられる。これをさかのぼっていくと、聖書における数の意味深い使用に突き当たる。たとえば、ペテロが、復活したイエスに命じられて網を上げると、そこには百五十三匹の魚がかかっていた。この百五十三という数は明らかに何ものかの象徴として意図的に選ばれたものであり、こうした象徴の解明に、神学は古くから関心を注いできた。（p.276）

d 関数についている程度の特徴ある数字は暗記していたのでこの数字を見た瞬間にピンときた。家にある聖書の該当箇所（ヨハネ福音書の最後の部分）を見ると、さらに驚くべきことに「3」の数字が回りに比較的多く見られる。つまり「153」と「3」をつなぐ

$$153 = 1^3 + 5^3 + 3^3$$

という関係である。古くからこの数字について研究されてきたようなのでこの関係式はもう指摘されていることかもしれないが、面白いところにこの数字が偶然出てきたので最後に紹介することにした。