

無限 n 進展開と進数表記の拡張

6年B組 川本 稜生
指導教員 川口 慎二

1. 要約

無限等比級数の公式を出発点として、整数を無限 n 進展開で表すことができた。また、そこから進数の負への拡張について考察した。

キーワード 無限等比級数、無限 n 進展開、無限 n 進数、 -2 進数、 $-n$ 進数

2. 研究の背景と目的

高校数学で扱う無限等比級数の公式では、収束条件として公比が -1 から 1 の間にあることが必要である。しかし、この条件を満たさない公比を代入した際においても、公式が成立するような数体系を構築することが可能であることを発見したため、従来とは異なる新たな数の表記法と、その利点について考察することにした。

3. 研究内容

3-1 無限 n 進展開

等比数列の初項を a 、公比を r として、無限等比級数の公式を以下に示す。

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r} \quad (-1 < r < 1) \quad \cdots \textcircled{1}$$

ここで、 $a=9$ 、 $r=10$ とすると、

$$\sum_{k=0}^{\infty} 9 \cdot 10^k = 9 + 90 + 900 + \cdots = \cdots 999$$

このとき、仮に $\textcircled{1}$ が成り立つとすると、

$$\cdots 999 = \frac{9}{1-10} = -1 \quad \text{となる。つまり、}$$

$$\cdots 999 = -1 \quad \cdots \textcircled{2} \text{を得る。}$$

ここで、 $\cdots 999$ に 1 を足すと、下のよう
に無限に 0 が続くため、これを 0 とみなせ

ば、 $\textcircled{2}$ は成立する。

$$\begin{array}{r} \cdots 999 \\ + \quad \quad 1 \\ \hline \cdots 000 \end{array}$$

さらに、 $\textcircled{2}$ を基準として、

$$\cdots 998 = -2, \quad \cdots 997 = -3, \quad \cdots 996 = -4,$$

$$\cdots 995 = -5, \quad \cdots, \quad \cdots 990 = -10, \quad \cdots$$

のように、順に定義していけば、負の整数全体を定義することができる。

次に、 $r=2$ の場合を考える。 $\textcircled{1}$ において
 $a=1$ 、 $r=2$ とすると、

$$\sum_{k=0}^{\infty} 1 \cdot 2^k = 1 + 2 + 4 + 8 + \cdots$$

となる。この和を表示することは困難だが、

2 進法に改めると、 $(1+10+100+\cdots)_{(2)}$ 、

つまり、 $(\cdots 111)_{(2)}$ と書ける。

以下、文中の特に表記のない数字は 10 進

数とする。また、 $\frac{1}{1-2} = -1$ であり、先の

例と同様の議論から、これは 2 進法における -1 を表していると言える。

これを一般化すると、 $a=n$ 、 $r=n$ のと

き、 $\sum_{k=0}^{\infty} (n-1)n^k$ は n 進法における -1 であると考えられる。

本稿では、このようにある数 m を無限桁の n 進数に変換する操作を m の無限 n 進展開と呼び、 n が整数のものを扱う。また、この操作によって展開された数を無限 n 進数と呼ぶ。

3-2 無限 n 進整数環

無限 n 進数についても、通常の数と同様に加法、乗法を定義することができる。以下の例は無限 10 進数によるものであるが、

$$-1 + (-1) = -2, \quad -1 \cdot (-1) = 1$$

などの計算結果と一致する。

$$\begin{array}{r} \cdots 999 \\ + \cdots 999 \\ \hline \cdots 998 \end{array}$$

のような加法や、乗法

$$\begin{array}{r} \cdots 99999 \\ \times \cdots 99999 \\ \hline \cdots 99991 \\ \cdots 9991 \\ \cdots 991 \\ \cdots 991 \\ \vdots \\ \hline \cdots 00001 \end{array}$$

が計算できる。

なお、筆算と同様にして減法の演算を行うこともできるが、無限 n 進数の場合、符号を用いずとも負の整数を表すことが可能なため、加法のみで十分である。

したがって、負の整数を無限 n 進数で置換した整数の集合 $\mathbb{Z}'$ は環の公理を満たし、無限 n 進整数環を構成する。

3-3 -2 進数

無限 n 進展開を用いて進数を負へと拡張

することを考える。

まず、以下のように -2 進数の表記と加法を定める。

$$(a_n \cdots a_2 a_1 a_0)_{(-2)} = \sum_{k=0}^n a_k \cdot (-2)^k \quad (a_j = 0, 1)$$

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 110 \end{array} \quad \begin{array}{r} 11 \\ + 1 \\ \hline 110 \end{array}$$

これらの計算規則は、各桁で等しく成り立つ。

このとき、 -1 を無限 -2 進展開すると、

$$\begin{aligned} \sum_{k=0}^{\infty} (-3) \cdot (-2)^k &= -3 + 6 - 12 + \cdots \\ &= (1101 + 11010 + 110100 + \cdots)_{(-2)} \\ &= (11)_{(-2)} = (-1)_{(10)} \end{aligned}$$

$$\begin{array}{r} 1101 \\ 11010 \\ 110100 \\ \vdots \\ + \vdots \\ \hline \cdots 000011 \end{array}$$

となり、 -1 に戻ることが確認できるため、この定義は妥当であると考えられる。

通常、10 進数を 2 進数に変換する際には、元の数値を 2 で割り続け、出てきた余りを並べるという手法が用いられるが、 -2 進数においても同様の計算が可能であり、10 進数と -2 進数是一对一で対応する。

また、無限 n 進数と同じく符号が不要であるが、有限桁で表現することができる。

乗法は 2 進数と全く同じであり、ビットシフト後に -2 進数の加法に基づいて計算すればよい。

$$\begin{array}{r} 0 \\ \times 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ \times 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

3-4 -n 進数

-2 進数の表記法を一般化すると、-n 進数の表記法が得られる。

$$(a_n \cdots a_2 a_1 a_0)_{-n} = \sum_{k=0}^n a_k \cdot (-n)^k$$

$$(a = 0, 1, 2, \dots, n-1)$$

基本的には n 進数の場合と同じ規則で計算可能であるが、 p ($p < n$) 繰り上がる際には 2 つ上の位に 1 を入れ、1 つ上の位に $n-p$ を入れる必要がある。これは、位の偶奇によって正負が入れ替わるため、 $(-n)^2$ 倍した位に足した分を補正する操作である。また、計算結果の先頭に無限に 0 が続くことを防ぐために、 $1(n-1)+1$ ($n=10$ のとき、 $19+1$) を 10 として扱う。これらの規則は、前述の -2 進数の基本演算規則にも内包されている。

乗法の場合も、繰り上がりを同様に処理することで計算できる。

4. 考察

一般に、コンピュータは 2 進数によって演算を行っており、負の数表現する場合には補数表現が用いられる。これは、オーバーフローを利用して「負の数を足す」ことによって、加法の内に減法を含め、演算処理を簡略化するためである。しかし、補数表現による数値はあくまで通常の 2 進

数として扱うこともでき、その場合は異なる値になってしまうため、変数には型の指定が必須となる。さらに、オーバーフローによる誤差が発生することもある。一方、-2 進数ならば各数値は正負を含めて一対一で対応しており、正の数は奇数桁、負の数は偶数桁となるため区別しやすい。当然、減法が必要ないという利点もある。同じ値を表現するのに 2 進数と比べてより多くの桁数が必要ではあるが、CPU の能力が年々向上していることを考えると実用も可能ではないだろうか。

5. 今後の展望

本稿では除法については特に言及しなかったが、余りの絶対値が最小になるように商を立てることで計算できると考えている。ただ、今回は完全に形式化することができなかったため、今後の課題とする。

また、加法、乗法についてもさらなる形式化を進め、-n 進数による演算を行えるプログラムを作成し、処理速度を比較する予定である。

6. 参考文献

[1] 加藤文元、中井保行 「天に向かって続く数」、日本評論社 (2016)

7. 謝辞

本研究を行うにあたり、顧問の川口先生にご指導いただきました。この場を借りて御礼申し上げます。