コラッツ予想の分析と拡張

5年B組　高橋　侑里
5年C組　石川　諒
指導教諭　川口　慎二

１．概要

21×2^x $(x\in\mathbb{Z},\ x\geq 0)$ について、コラッツ予想が成立することを証明した。また、コラッツ写像の n が奇数のときの条件を $g(n)=3n+a$　$(a\in\mathbb{N})$ に変更しても、コラッツ予想が成り立つことを、python を用いて検証した。

２．目的と背景

数学の未解決問題を調べている中で、コラッツ予想というものがあるのを知った。そして、四則演算のみで定式化されているという点や一見単純そうに見えるのに 80 年間も未解決であることに興味を持ち、研究を始めた。

コラッツ予想とは、

「自然数 n に対して n が偶数のとき、n を 2 で割り、n が奇数のとき、n を 3 倍して 1 を足す」

という操作を有限回繰り返すと、いずれ必ず 1 になるという予想である。

Lothar Collatz が 1937 年に本予想を提唱した。現在、カリフォルニア大学の Terence Tao が「ほとんどすべての正の整数において正しい」と発表しているが完全な証明は発見されていない。

本研究では 21×2^x $(x\in\mathbb{Z},\ x\geq 0)$ と 2^y $(y\in\mathbb{N})$ の形の数についてのコラッツ予想の証明とコラッツ写像の n が奇数のときの条件を $f(n)=3n+a$ $(a\in\mathbb{N})$ に拡張しても同様にコラッツ予想が成り立つか検証を行った。

３．研究内容

ここでは、21×2^x $(x\in\mathbb{Z},\ x\geq 0)$ の形の数についてコラッツ予想が成立するか 2^{6t} を 3 を法として調べる。

2^6 についてコラッツ予想の逆の操作を考え、

「逆の操作をする数から 1 を引き、3 で割る」という操作が可能なときは優先的に行う。

以下、「逆の操作をする数から 1 を引き、3 で割る」という操作を①、「逆の操作をする数に 2 倍する」という操作を②と表現する。すると、

$$2^6=64 \xleftarrow{①} 21 \xleftarrow{②} 42 \xleftarrow{②} 84 \xleftarrow{②} \cdots\cdots$$

となり、1 回目の①の逆算操作後の数は 3 の倍数となる。その後は、操作②しか行うことができない。

すなわち、コラッツ予想の操作で 2^6 に到達する数は、21×2^x $(x\in\mathbb{Z},\ x\geq 0)$、$2^y$ $(y\in\mathbb{N}$、$y\geq 6)$ から出来るため、21×2^x となる数はコラッツ予想の操作より 1 に到達するといえる。

そこで、2^{6t} $(t\in\mathbb{N})$ についても上記と同様のことが言えるのではないかと考えた。

【補題 1】　64^t-1 は 9 の倍数となる。

（証明）3 を法とする。

$$64^t-1=4^{3t}-1$$
$$=(4^t-1)(16^t+4^t+1)\ \cdots①$$

ここで、$4^t-1\equiv 0\ \cdots②$

$$16^t+4^t+1\equiv 1+1+1\equiv 0\ \ \cdots③$$

②, ③より、4^t-1 と 16^t+4^t+1 はともに 3 の倍数である。よって、①の式は 9 で割り切れる。　［終］

【命題 1】　21×2^x $(x\in\mathbb{Z},\ x\geq 0)$ について、コラッツ予想が成立する。

（証明）$2^{6t}=64^t$ に①の逆算操作を行うと得られる $\dfrac{64^t-1}{3}$ は補題 1 より、3 の倍数である。そのため、逆算操作で②の操作しか行うことができなくなる。これにより、2^{6t} は $\dfrac{2^x\{(64^t-1)\}}{3}$，$2^u$ $(u\in\mathbb{N},\ 6t\le u)$ からのみ出現するということがわかる。よって、21×2^x はコラッツ予想の操作を行うことにより、1 に到達することがいえる。

［終］

４．コラッツ予想の拡張

はじめに、コラッツ写像を定義する。

【定義 1】任意の正の整数 n に対して、コラッツ写像 f を次のように定義する。

$$f(n)=\begin{cases}\dfrac{n}{2} & (n:\text{偶数})\\ 3n+1 & (n:\text{奇数})\end{cases}$$

コラッツ予想は、コラッツ写像を用いると「任意の自然数に対して、コラッツ写像を繰り返し用いると、いずれは 1 に到達する。」と言い換えることができる。

【仮説】a を自然数として、

$$g(n)=\begin{cases}\dfrac{n}{2} & (n:\text{偶数})\\ 3n+a & (n:\text{奇数})\end{cases}$$

のように拡張されたコラッツ関数 g を定義する。このとき、任意の自然数に対して、関数 g を繰り返し用いるといずれは 1 に到達するのではないかと考えた。以下、これを「拡張されたコラッツ予想」と呼ぶことにする。ただし、今回は 1 から 1000 までの範囲で拡張されたコラッツ予想を確認する。

【命題 2】拡張されたコラッツ写像 $g(n)=3n+a$ （n：奇数）の a は必ず奇数でなければならない。

（証明）n が奇数のとき、a を偶数する $g(n)$ は奇数となる。よって、n が奇数ならば、$g(n)$ は増加関数となるため、拡張されたコラッツ予想が成立することはない。よって、a は奇数である。　［終］

そこで、Python を用いて操作を行った。

【仮説 2－1】

拡張されたコラッツ操作を行う数を a を除いて関数 g を繰り返し用いると、全ての数が 1 に到達する数が存在する。

【結果 2－1】

$3\le a\le 299$ のとき、そのような数は存在しないことがわかったが、m を任意の自然数とし、$g(am)$ の計算後の結果が $g(am)$ になっていた。

ここで次のような仮説を立てる。

【仮説 2－2】

コラッツ操作を行う数を a の倍数を除いて上記と同様の方法で計算を行うとすべての数が 1 に収束する数が存在する。

【結果 2－2】

プログラム（図 1）を作成し、検証してみると定数項が 41，43，107，113 のとき、コラッツ予想と似た法則が見られた。

```python
try:
    try:
        print('計算する値を入力してください')
        num = int(input())
        count = 0
        if num > 0:
            while True:
                num1 = num + 1
                ### if num1 % ??? == 0: の???を変更する ###
                if num1 % 41 == 0:
                    num1 = num1 + 1
                else:
                    num1 = num1
                print(str(num)+'の計算をします')
                while num != 1:
                    if num % 2 == 0:
                        num = num // 2
                        count = count + 1
                    else:
                        ### num = num * 3 + ??? の???を変更す
                        num = num * 3 + 41
                        count = count + 1
                print('          '+str(count)+'回計算しました')
                num = num1
                count = 0

    except ValueError:
            print('---ERROR---正しい値を入力してください')

except KeyboardInterrupt:
    print()
```

図１　Python での検証画面

また、Python を用いて以下のプログラムを作成した。このプログラムにより、コラッツ操作を行う際に人力よりも計算の正確性が向上したため、確認する際にはこれを使用した。

```python
num = int(input("整数を半角で入力してください>
count = 0
print(num)
while True:
    if num % 2 ==0:
        num = num //2
        print(num)
        count = count + 1
    elif num % 2 != 0 and num != 1:
        num = num * 3 + 1
        print(num)
        count = count + 1
    else:
        break

print(str(count)+"回計算しました")
```

図２　Python によるプログラム

５. 考察

コラッツ予想を Python と mod を用いて一部の証明や、拡張を行った。今回は 2^{6t} から考えたが、今後は別の始点からの逆算の操作を行えるか検討したい。また、拡張ではオイラー素数との関係について言及したが、オイラー素数とはまったく関係ない数も結果として現れたため、今後これらの数にはどのような規則性があるのか、また規則性はないのかを検討していきたい。さらに $g(am)$ がなぜ 1 に収束しないのかについても考えていきたい。

６．参考資料

[1] 数学の景色
https://mathlandscape.com/collatz/

７．謝辞

本研究にあたり協力してくださいました小川翼さん、中村一葉さん、熱心なご指導を頂いた顧問の川口慎二先生に感謝の意を表します。