

2024年度

# 卒業研究

総合理工学科

松田 修 研究室



2024年度 松田研究室 メンバー

岡崎 莉輝斗（先進科学系）

澤田 晴丈（先進科学系）

指導教員：松田 修

卒業研究

虫の音に関する研究

先進科学系

岡崎莉輝斗

指導教員 松田修

令和 7 年 1 月 20 日

## 概要

私は鈴虫の鳴き声を再現しようとした。そのために鈴虫の鳴き声の周波数を測定した。すると主な周波数は 4000Hz, 6000Hz, 8000Hz と分かった。そしてその音を再現するために wave pad 音声編集ソフト及び Audacity を使用した。最終的に鈴虫の鳴き声との周波数と差の平均を 62.8 から 13.5 にまで縮めることに成功した。

## Abstract

I tried to reproduce the sound of bell crickets. To do this, I measured the frequency of bell crickets' sounds. I found that the main frequencies are 4000Hz, 6000Hz, and 8000Hz. I then used Wave Pad Audio Editing Software and Audacity to reproduce this sound. I was able to reduce the average difference with the bell cricket's sound from 62.8 to 13.5.

## 目次

|                         |    |
|-------------------------|----|
| 1. はじめに .....           | 1  |
| 2. 鈴虫の鳴き声の調査.....       | 1  |
| 3. 鈴虫の鳴き声をパソコン上で再現..... | 2  |
| 4. 鈴虫の鳴き声を実物で再現.....    | 4  |
| 5. 鈴虫と松虫のハイブリッドの作成..... | 4  |
| 6. 相関を用いた鈴虫の音の再現.....   | 6  |
| 7. 差の平均を用いた鈴虫の音の再現..... | 7  |
| 7.1 11500 Hz の調整.....   | 11 |
| 7.2 13000 Hz の調整.....   | 14 |
| 7.3 14500 Hz の調整.....   | 15 |
| 7.4 16000 Hz の調整.....   | 16 |
| 7.5 17000 Hz の調整.....   | 17 |
| 7.6 20000 Hz の調整.....   | 18 |
| 7.7 21500 Hz の調整.....   | 19 |
| 8. おわりに .....           | 20 |

## 1. はじめに

初めに私がこの研究をしようと思ったきっかけは、虫の出す音をその原理から人工的に再現することは可能なのかという点である。この原理とは鈴虫が音を出す際に、羽を擦り合わせる動作のことである。これまでに実際に虫の音をその原理から人工的に再現されたことはない。世界初の研究をしたいと思ったからである。

## 2. 鈴虫の鳴き声の調査

研究の第一段階として、様々な虫の音を「wave pad 音声編集ソフト」で、フーリエ解析を行いその音の周波数を調査した。結果として、鈴虫の波長のピークに周期性が現れていたため、今回の目標である再現という観点から鈴虫は適している。そこから鈴虫に着目して研究を進めていった。実際に鈴虫の周波数の波形は以下の図のようになる。

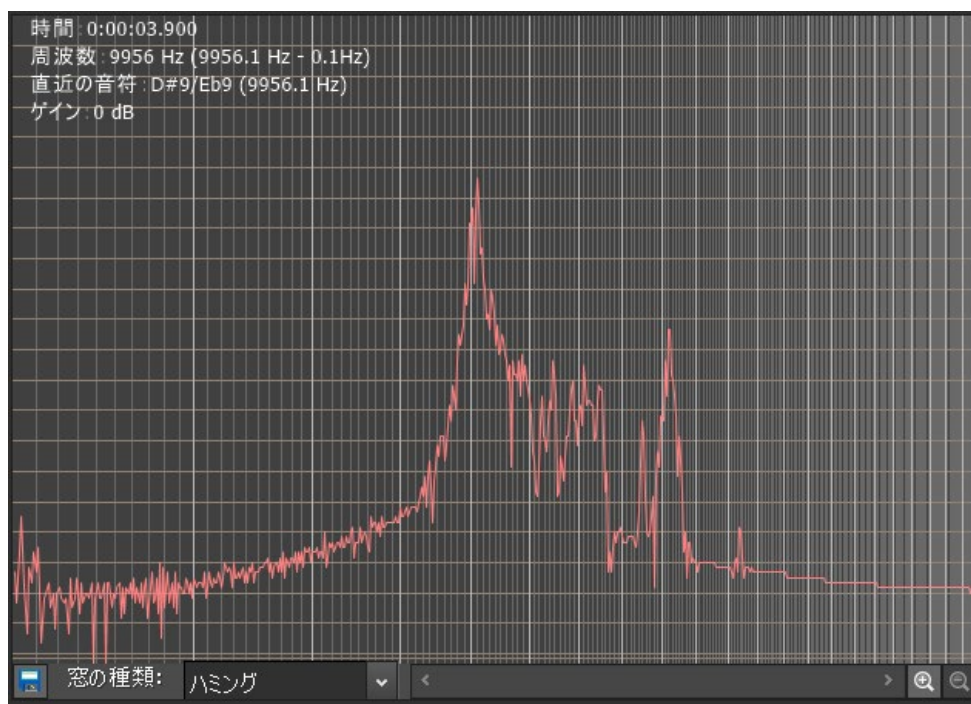


図 1 鈴虫の周波数の波形

これを再現するために、まず鈴虫の鳴き声の原理について調べた。鈴虫の鳴き声は羽をこすり合わせることで音が出る仕組みとなっている。従って鈴虫は鳴き声といっ

ているが鳴いているわけではない。羽の付け根にはノコギリの刃のような細かい凹凸の彫られたヤスリ器と、硬い板状になった摩擦器という2つの器官がある。この2つの器官が羽をこすり合わせる際に、摩擦器がヤスリ器に引っかかりながら動く、この動作が連続的に起こることによって音が鳴るという仕組みである。この動作を鈴虫は1秒間に60往復して音を鳴らしている。しかし、この動作だけでは人間の耳に届くような大きな音は出ないため、鈴虫は音の増幅を行なっている。音の増幅の仕組みとは、1つは「固有振動」である。物体に音や光などの振動が加えられることによって、その物体は特定の振動数の振動をするという性質を持つ。2つ目に、2つの振動の山と山、谷と谷がぶつかることでその振動が大きくなるという性質がある。この2つの性質によって音を増幅することができる。つまり「音源」から出た音を「固有振動が音源と同じ物体」に与えることによって音は増幅される。このように、固有振動を利用することによって振動を増幅させることを「共振」という。この共振によって鈴虫は音を増幅させることができる。鈴虫の羽の中央部には発生させた音と共振する部位がある。その部位が弦楽器の弦を張るように羽を立てることによって、我々の耳に届くような音に増幅させることができている。

### 3. 鈴虫の鳴き声をパソコン上で再現

次にパソコン上で鈴虫の音を再現した。前節の図1の波形では、波長のピークの4000Hzと8000Hzが他の周波数と比較して突出しており、その次に6000Hzが突出しているのが確認できた。この音を再現するために、「Audacity」という音声編集ソフトを使用した。始めに、4000Hzと8000Hzを図2のような実際の音声の鈴虫の波形に沿った形にして、音を作成した。作成した鈴虫の音と本来の鈴虫の音の近似性は、人間の感覚とプログラムによって判定している。プログラムについては第6節から詳しく説明する。



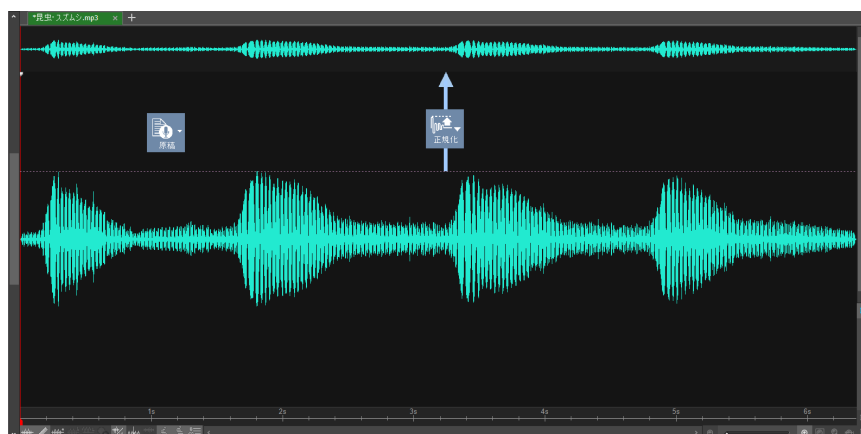


図 2 鈴虫の音声波形

結果として、作成した音声は機械的な音にしか聞こえず、鈴虫の音色の再現という目標には不十分であった。この欠陥を取り除く為に、再現する音声に断続性を持たせた。鈴虫の名前にもある鈴は内部にある玉が外殻と接触することによって音が鳴る。鈴の音は、笛の音のような持続的で一定の長さを持つ音ではなく、断続的に何度も短音が繰り返し鳴るものである。これは鈴虫が1秒間に60回羽をこすり合わせることによって音を出しているということにも共通している。このような共通点から昔の人はこの虫の名を鈴虫と名付けたのかと考えると、昔から伝わる名前が名付けられる際にどのように考えてその名前になったのかが知れて面白いと感じる。

この断続性の付与により、人間の感覚上からの評価ではあるが格段に鈴虫の音の再現性が向上した。完成したものが下図になる。

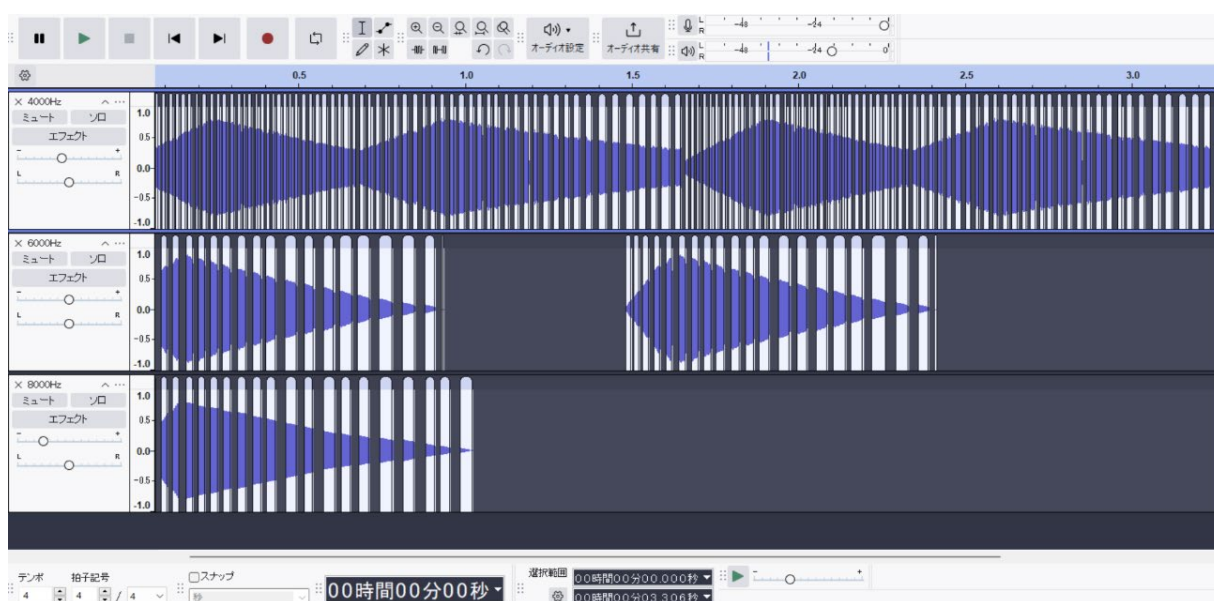


図 3 断続性を付与した鈴虫の鳴き声の再現

このように 4000Hz は一番ピークが大きかったため、音と音の間隔をできる限り短くし鈴虫本来の音に近づけるようにした。そして 6000Hz の波形は特殊で 4000Hz と 8000Hz の振幅が大きくなるときにはその振幅が小さくなるという反比例の関係であったため、それも再現した。そして 8000Hz は 1 回しか繰り返していない。理由としてこれは感覚に頼るものにはなってしまうが、完璧に鈴虫の波形に合わせようとする逆逆に機械感が増していき、鈴虫の音からはどんどん離れていっていると感じたためこのあたりは聞いた際の感覚で作っていった。再現しようとしている音が、虫という生物のものであるため常に全く同じ音を出している訳ではない。そのため、こうした曖昧さも加味する必要があるが、今回は取り扱わない。

#### 4. 鈴虫の鳴き声を実物で再現

ここまでパソコン上での鈴虫の音の再現は出来てきたため、次は実際のものを使用して再現した。鈴虫は羽と羽をこすり合わせることで音を出すため、身の回りにあるものをこすり合わせて音の再現を試みた。下敷きや紙、のこぎりやナイフなど薄いものを使用した。その中で、のこぎりと目の細かいナイフをこすり合わせた際の音が最も鈴虫の鳴き声に近似していた。のこぎりやナイフの構造は、鈴虫の羽の構造と同様に凹凸がありそれらを擦ることによって、断続的な短音を出すことができる。しかし、問題点は単にこすり合わせるだけでは鈴虫の 4000Hz や 8000Hz のような高周波数には届かない。そのため、一般的な楽器の中でヴァイオリンが鈴虫に類似していることを発見した。ヴァイオリンは弦と弦をこすり合わせて音を出している。その音はボディによって増幅されている。その仕組みは鈴虫の音の原理に近似している。ただし、ヴァイオリンの出せる最高周波数でも 4000Hz を少し超える程度であった。そのため、一般的な楽器では 8000Hz には届くことはない。参考文献のサイトを参考にモーターとペットボトルの蓋を使用して鈴虫の音を再現した。しかしながら鈴虫の音は出ず、ただペットボトルの蓋が擦れている音にしか聞こえなかった。以上のように、これまでのこぎりやモーターなど様々なものを使用して鈴虫の音を再現しようと試みてきたが、鈴虫の音を再現することに失敗した。そのためこれ以上研究を行ったとしても卒業研究の間に形にすることは難しいと判断し、やむなく鈴虫の鳴き声を物体として再現することを断念することになってしまった。

#### 5. 鈴虫と松虫のハイブリッドの作成

次の方針として、虫と虫の異なる種同士をかけ合わせたハイブリッド昆虫について

の調査を行なった。ハイブリッド昆虫は現実では実現不可能であるため、調査内容は理論上のものである。例えば、鈴虫と松虫に子孫が出来ればどのような鳴き声で鳴くのかという調査である。この調査のためにまず松虫の鳴き声をまたパソコン上で再現した。

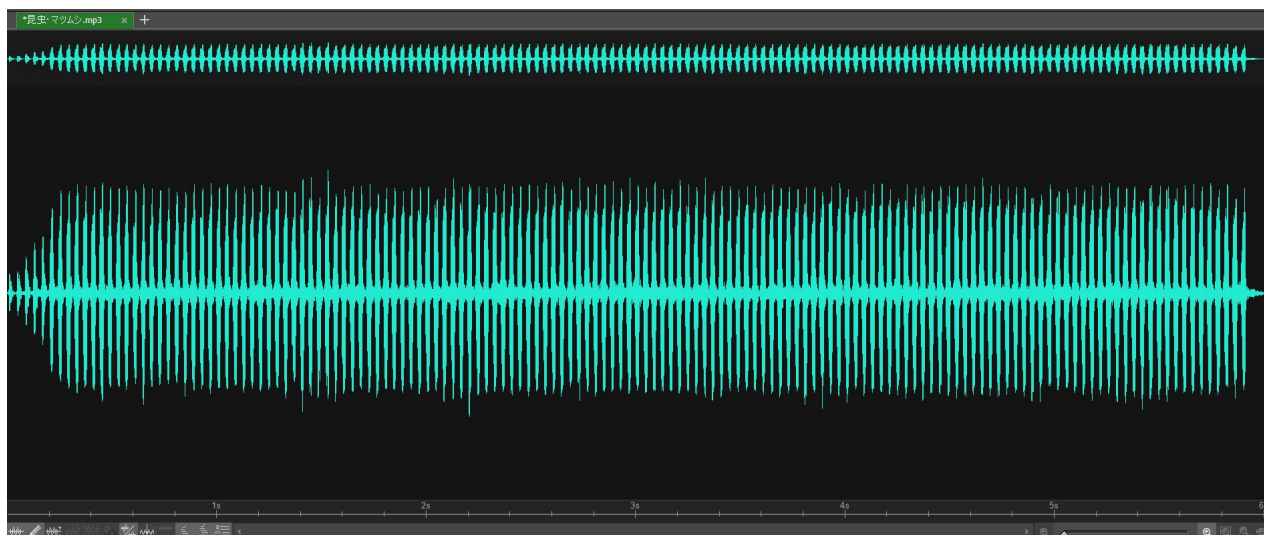


図 4 松虫の音声波形

そしてこの図4が実際の松虫の音声波形である。鈴虫とは異なり6秒間ほど均一に鳴き続けている。松虫は波形が単調なのだが、感覚的なものではあるが音に独特な「強さ」がありそれを機械で再現するのが想定していた以上に難しかった。また鈴虫と松虫の子孫の鳴き声を鈴虫の音源と松虫の音源をかけ合わせて作成したが、2匹の異なる虫が鳴いている様にしか聞こえなかった。そこで、例えば鈴虫の鳴き声を松虫のように6秒間均一に鳴いているように調整したが、それは鈴虫が「頑張って」鳴いているようにしか聞こえなかった。逆に、松虫を鈴虫の波形のようにピークが上下するように調整しても、これもまた感覚的なものでしかないが松虫が「疲れている」状態で鳴いているように聞こえただけであり、その鈴虫と松虫の間のような音は作れていない。そのため、これは波形を変えるだけではその鳴き声の状態は変わるが、結局元の音源の虫が鳴いているとしか聞こえない。そのため周波数を調整することがハイブリッド昆虫をつくる方法がと考えられた。そうして周波数をいじるなどして調整してみたが、鈴虫の性質を残しつつ、松虫の性質も持った鳴き声をつくることはできなかった。鈴虫と松虫のハイブリッドの鳴き声という、空想上のものをつくるには、その音を聞いた人々がその音がハイブリッドであると信じるための説得力を与える必要があったのだと考えられる。そしてその説得力というのが見当もつかず、終着点も分からなくなってしまったため、この研究もやむなく断念することになってしまった。

## 6．相関を用いた鈴虫の音の再現

ここまで様々な方針を立て、数多く実践してきたが中々結果を出すことができず、形にすることが出来なかったため最終的な目標としてパソコン上でつくる虫の音を限りなく本物に近づけることを最終目標と設定した。そのために、今まで作ってきた鈴虫の音と本物の鈴虫の音の相関関係からどれだけ近づけられているのかを調べることにした。その調査方法としては十進 **basic** を用いた。そして、そのスペクトルの相関を調べるために用いたプログラムは以下の付録に記載する。相関関係を調べることでどの部分を調整すれば、本物の音に近づけるかの指標になると考えられる。そして、実際に自身が作った鈴虫の音の相関を測定した結果が以下になる。

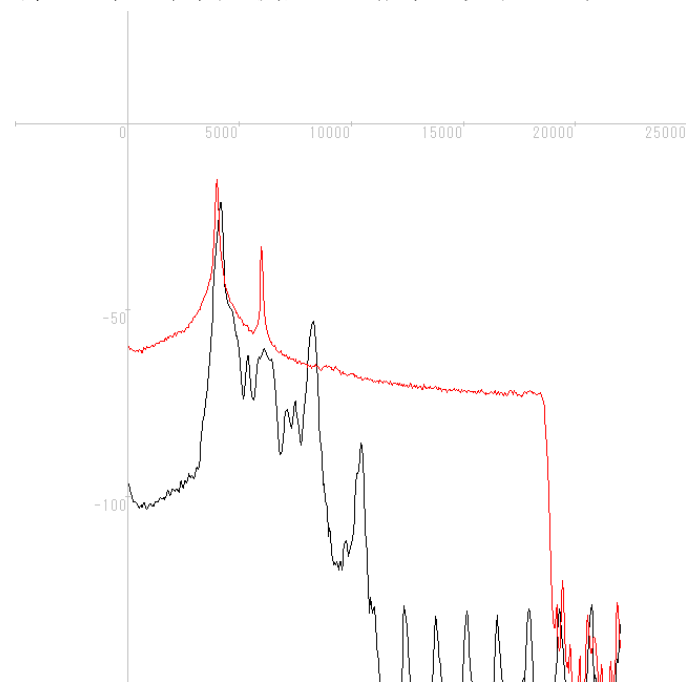


図 5 鈴虫スペクトル相関

このグラフは黒線が本物の鈴虫のスペクトルで、赤線が作った鈴虫の音となる。そして相関係数  $r = 0.587757034309996$  という結果が出た。この相関係数の値が 1 に近づけば近づくほど正の相関が強いという事になるため、これを 1 に近づけることを目標とした。まずこの図 5 を見ると 1 番大きいピークが 4000 Hz でほぼ合っているのだが、2 番目のピークの 6000 Hz は飛び越えすぎておりそこを抑えなくてはならないという事が分かった。また特徴として 10000 Hz を超えた辺りから小さいピークが一定間隔で続いている。このため課題点は 6000 Hz のピークを少し抑え、10000 Hz 以上の周波数を全体的に抑えることであると分かった。しかし、ここで以下に示す問題が発生した。

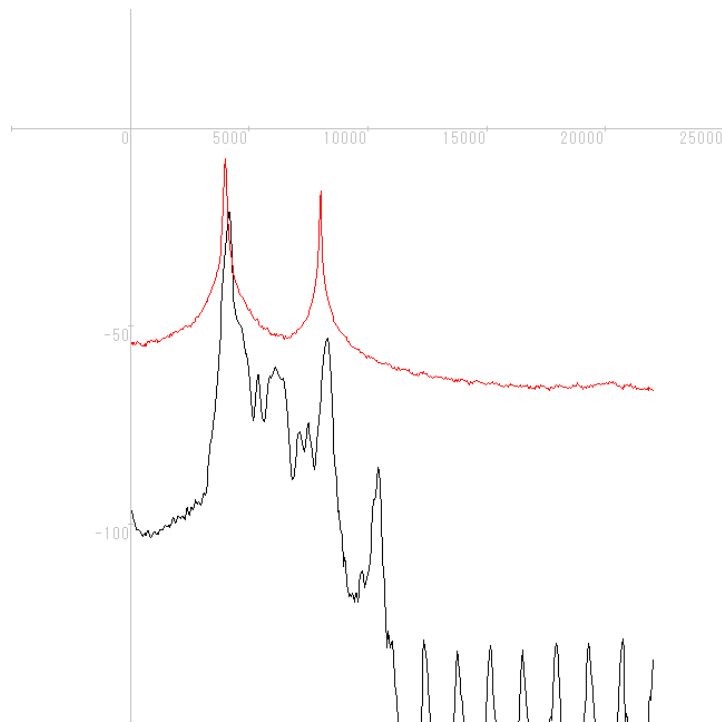


図 6 鈴虫スペクトル相関 2

色々周波数をいじっていたところ、図 6 のようになり 6000 Hz はちょうど良くなったが 4000 Hz や 8000 Hz のピークが飛び抜けてしまい、それに伴って 10000Hz 以上の部分も大きくなってしまった。これを見てこの結果の相関係数は下がっているだろうと想定したのだが、結果はその反対の数値を示した。結果は相関係数  $r=0.878986827643607$  と先ほどの相関よりも高くなってしまった。全く近づいているとは思えない状態でのこの数値であったため、この方法では問題があると判明した。この問題を解明するために、改めて相関について調べることにした。するとこれが問題であった理由として、相関というのは 2 種のデータの関連性であるため、必ずしも 2 種のデータが近くなるにつれて相関係数が高くなることでは無いということだった。そして相関というのは外れ値によって大きく相関係数が変化してしまうため、今回の調査には向いていない調査方法であったと分かった。

## 7. 差の平均を用いた鈴虫の音の再現

そこで次なる調査方法として差の平均を用いた。この差の平均は、単純に 2 種のデータ間の差がどのくらい離れているかを測定するものであるため、差を小さくしていけば本来の鈴虫の音に近づけられるということである。そしてこのスペクトルの差の平均を求める為に用いたプログラムも以下の付録に記載する。ここから更に鈴虫の音の精度を

上げようと、鈴虫の音を一から作ることにした。まず鈴虫の音というのは大きく見ると図2のようになっている。しかしこれを再現するだけではあまり近づけたとは言えなかった。そこで以下の図のようにまで拡大し、一音一音がどのようにして出来ているのか観察した。

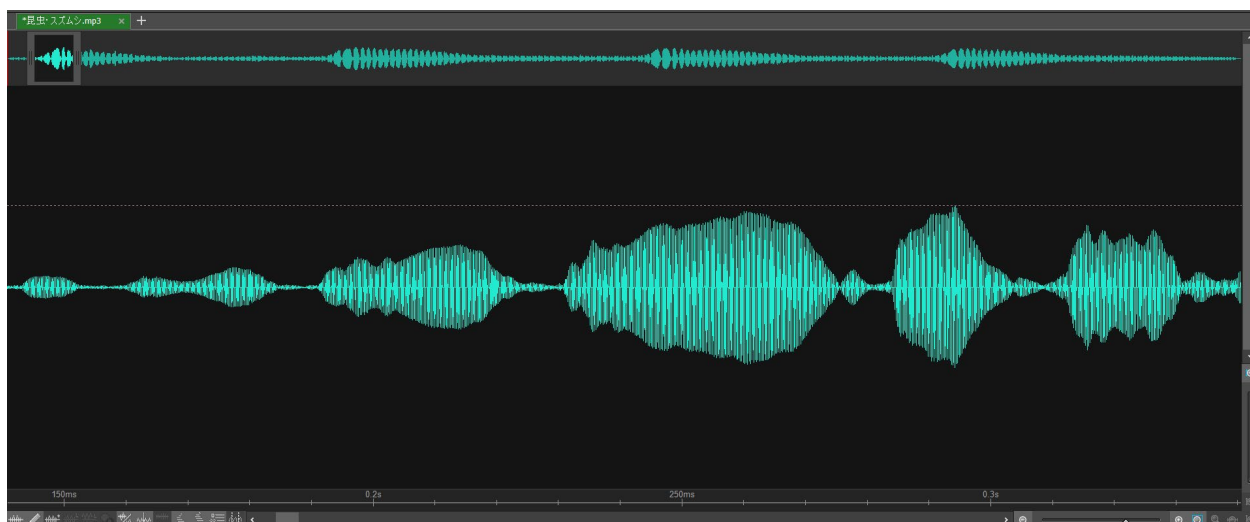


図 7 鈴虫の波形の拡大図

鈴虫の音は大体一音が 0.01 秒～0.03 秒の塊で出来ていることが分かった。そして振幅の大きさも 1 つ 1 つ異なっていたため、そこも合わせられるように作った。そして完成したのが以下の図である。

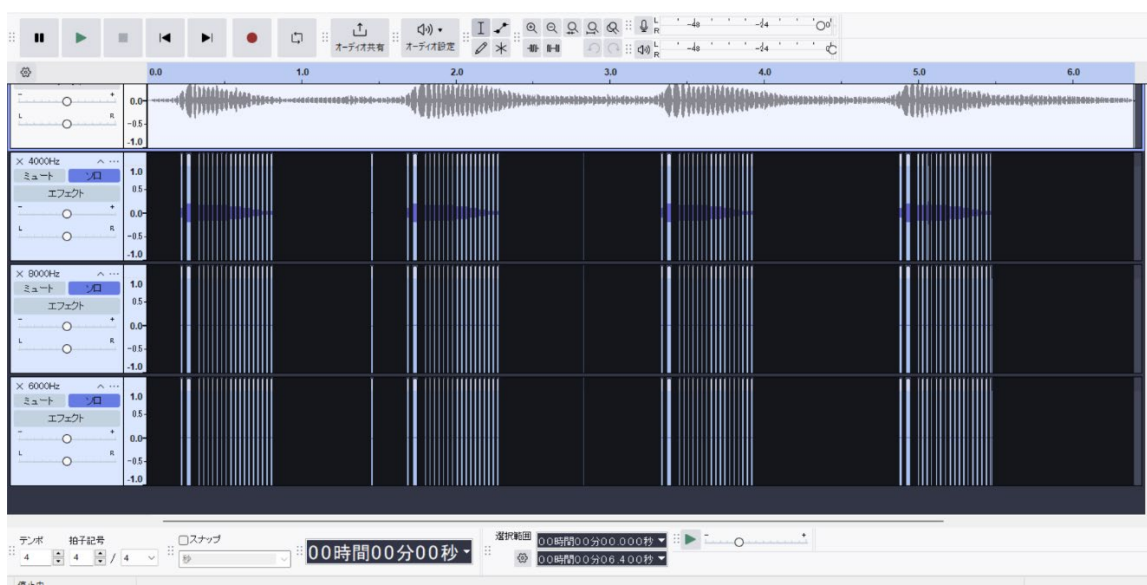


図 8 鈴虫再現完成図



一番上は手本とした本物の鈴虫の音で、下に順に 4000 Hz、8000 Hz、6000 Hz となっている。0.01 秒の音を 6 秒の間に何百個と作ったため、全体から見ると重なりすぎて見えなくなってしまっている。そこで拡大したものを以下に図として示す。

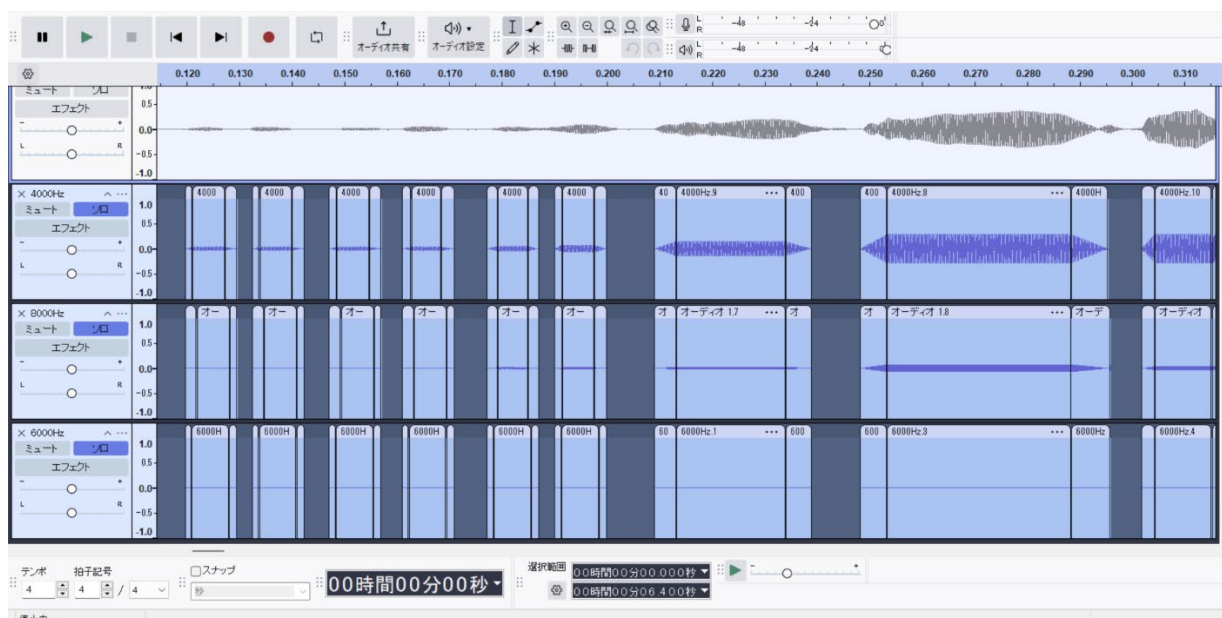


図 9 鈴虫再現完成図拡大

まず 4000 Hz の部分は一番大きいピークであったから、大体本来の鈴虫の音に近い振幅に合わせた。そしてその次に大きいピークである 8000 Hz は振幅を抑えて作った。これは、一回作った際に本来の鈴虫の 8000 Hz が想像以上に振幅が小さかったためである。最後に 6000 Hz は、これは聞こえてしまうと、ノイズのように聞こえてしまうため人間の耳でははっきりとは分からないくらいの振幅で作った。具体的な数値で言うと 6000 Hz はほとんどの振幅を 0.005 で作った。この音を作っている中で特に難しかった事は、この参考にした鈴虫の音が見ても大体の振幅くらいしか分からず、周波数ごとの振幅は分からないため感覚で作るしかなかった事であった。そして機械で作るとなると 1 つ 1 つの周波数ごとに作るしかないので、振幅の調整を探り探りでする必要があったのも難しい点だった。こうして完成した鈴虫の音を差の平均のプログラムにかけた結果が以下の図のようになった。

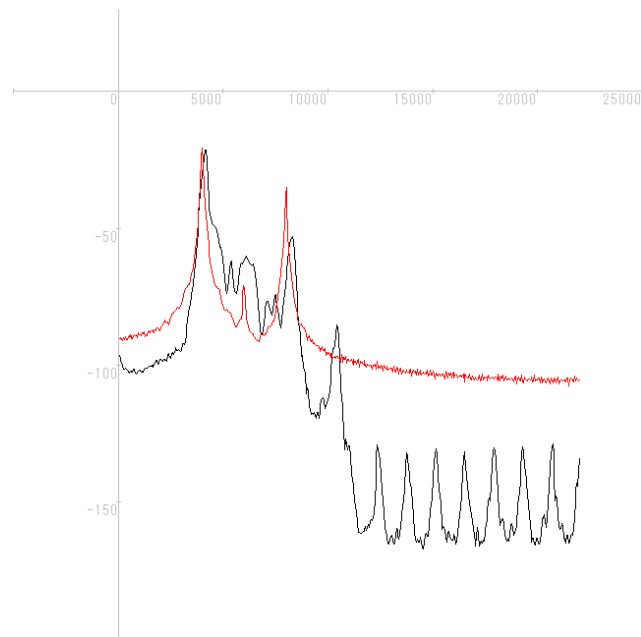


図 10 スペクトルの差の平均 1

この実験では、差の平均 = 32.2354565283757 という結果となった。4000 Hz は完全に一致したが、やはり 6000 Hz を小さくしすぎたのと、8000 Hz が少し大きく、10000 Hz 以上を下げていかないといけないということが分かった。そして参考のため今まで作っていた図 3 の鈴虫の音の差の平均も求めた。以下にその図を示す。

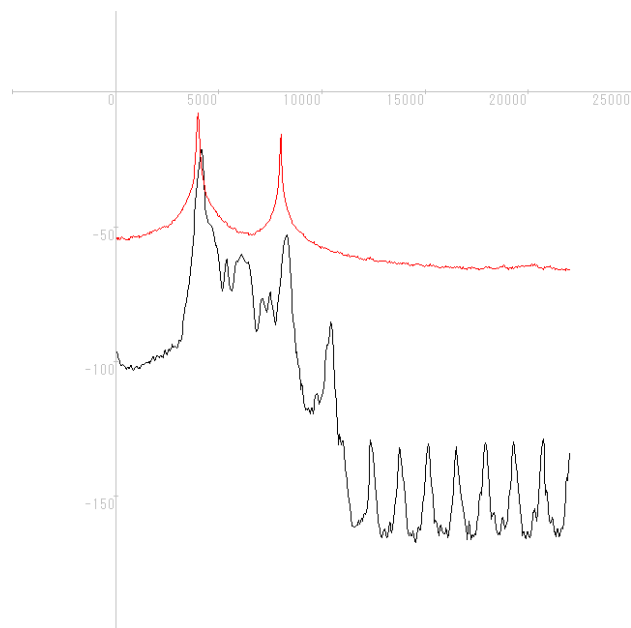


図 11 スペクトルの差の平均 2



この実験では、差の平均 = 62.8301862700587 という結果となった。明らかに先ほどの新しく作った鈴虫の音よりずれていたため、このスペクトルの差の平均の値が大きいというのは妥当であり、この差の平均を求める方法で調査するのは適していると分かった。

## 7.1 11500 Hz の調整

そして次に、10000 Hz 以上の周波数を抑えることが新たな課題となった。解決方法の一つとしてノッチフィルターという機能が挙げられる。この機能は周波数を指定し、その周波数に対して数値を与えることによって、その部分を小さくすることができるというものである。この数値がどのくらい入れたら良いのか分からず、試行錯誤を繰り返した。その過程を以下に載せていく。

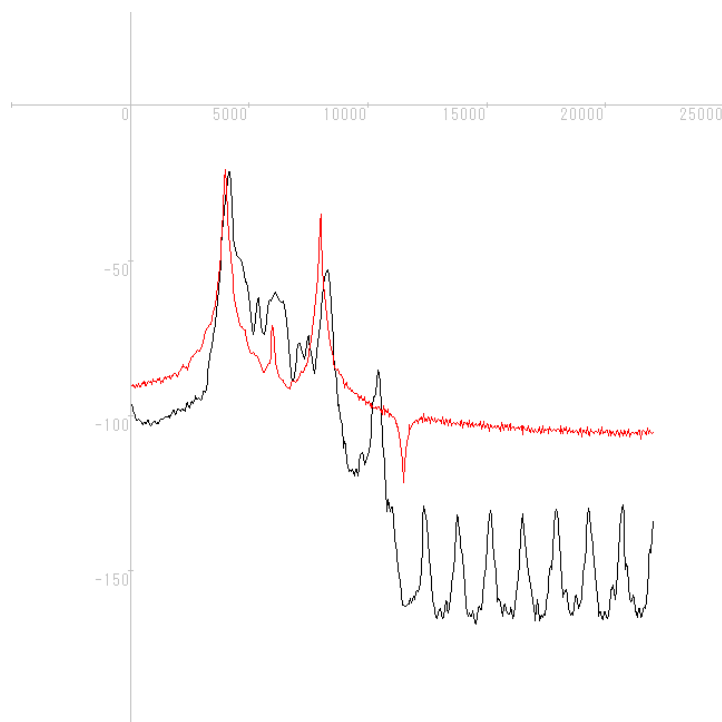


図 12 11500 Hz にノッチフィルターを 10 与えた差の平均

この実験では、差の平均  $m=31.8638519373777$  という結果となった。これはまず試しに 11500 Hz にノッチフィルターを 10 与えた結果である。確かに 11500 Hz の部分を小さくすることが出来ている。そして差の平均もノッチフィルターをかける前と比べると小さくなっている。ノッチフィルターで与える数値というのは値が大きくなるほど幅が狭くなることから、この数値をより大きくすればもっと周波数を抑えることが出来ると予想された。次は 11500 Hz にノッチフィルターを 100 与える結果を以下の図に示す。

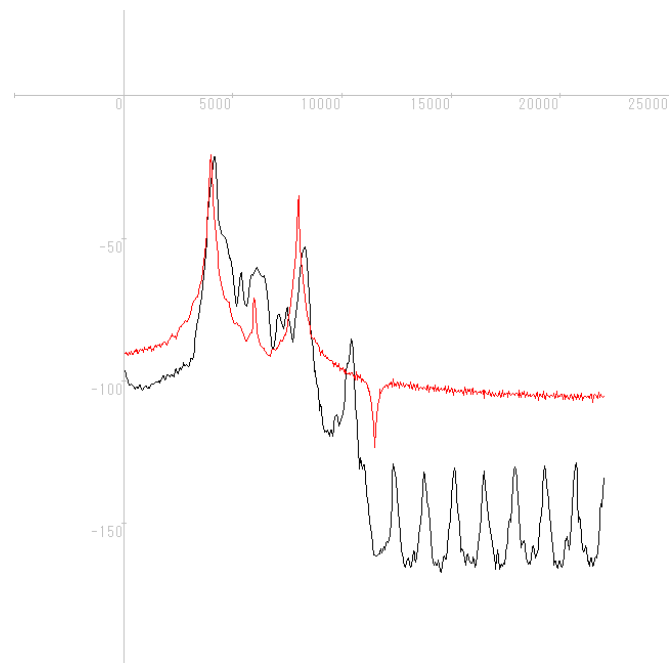


図 13 11500 Hz にノッチフィルターを 100 与えた差の平均

この実験では、差の平均 = 31.8484975733855 という結果となった。想定通り与える数値を大きくすれば、差の平均も小さくすることが出来た。しかし、下がったのが 0.015 ほどとあまり変化が見られなかった。そのため与える数値が少なかったと予想され、今度はノッチフィルターを 1000 与えた。以下に図として示す。

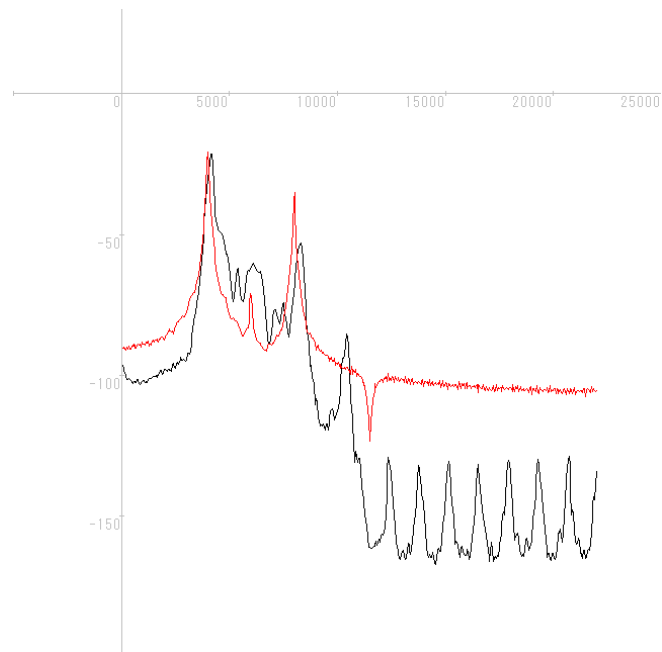


図 14 11500 Hz にノッチフィルターを 1000 与えた差の平均

この実験では、差の平均 = 31.8483916790607 という結果となった。結果を見ると先ほどの 100 与えたものから 0.0001 しか変化していない。これを見るにこの機能ではこのくらいが限界なのかもしれないと感じてきた。ならば逆に数値を小さくしてみればどうなるのかを検証してみた。試しにノッチフィルターを 1 与えた結果を以下に示す。

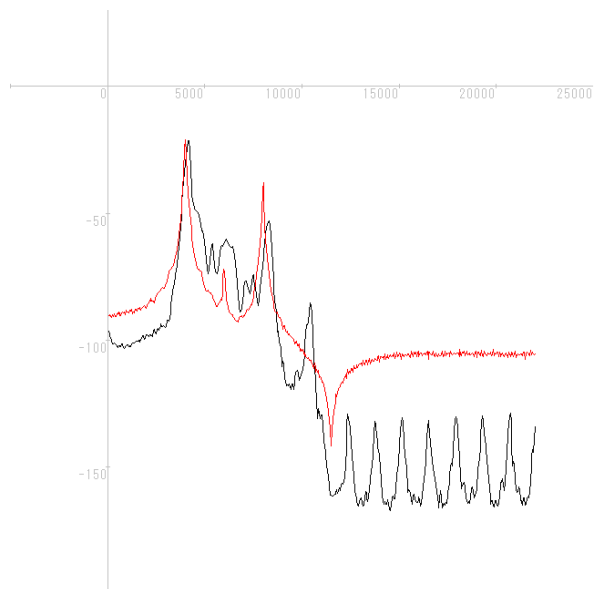


図 15 11500 Hz にノッチフィルターを 1 与えた差の平均

この実験では、差の平均 = 29.5631720293542 という結果となった。先ほどと比べて明らかに小さくすることができた。この結果からこの数値は小さくするほど、その周波数を下げることが出来ることが分かった。そしてもう少し小さくしたいと思い、ノッチフィルターを 0.1 与えてみた。その結果を以下に示す。

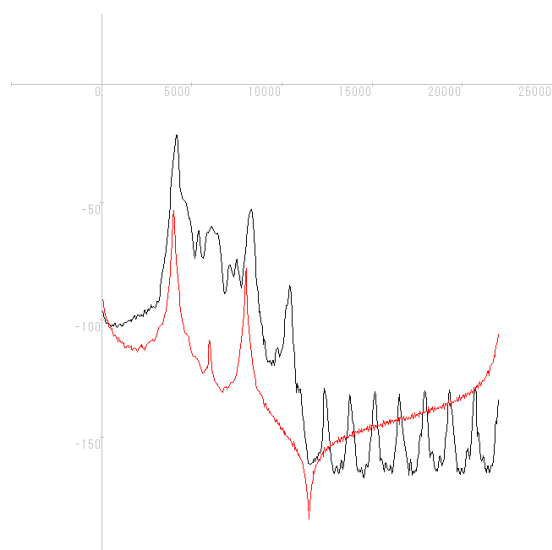


図 16 11500 Hz にノッチフィルターを 0.1 与えた差の平均

この実験では、差の平均 = 24.6225287260274 という結果となった。これは明らかに小さくしすぎた。想像以上に小さくなってしまい、他の 4000 Hz や 8000 Hz までもつられて小さくなってしまっている。しかしこれは良い発見であったと思う。ノッチフィルターで周波数を下げると他の周波数もつられて下がるのならば、その値次第で他の周波数も共に調整できると考えられた。そして今回はノッチフィルターで下げすぎてしまったため、先ほどの 1 と 0.1 の中間の 0.5 ならばちょうど良いだろうと予想し、次は 0.5 を与えた結果を以下に示す。

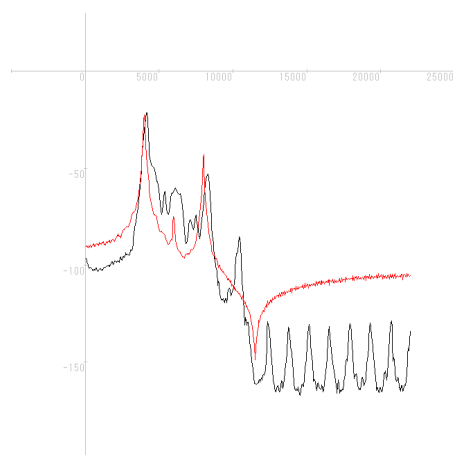


図 17 11500 Hz にノッチフィルターを 0.5 与えた差の平均

この実験では、差の平均 = 27.9924117338552 という結果となった。結果を見ると先ほどよりは差の平均が大きくなってしまい、目標としている所までは下がりきっていないが、これは先ほど見つけたノッチフィルターの性質からこの後十分調整できるであろうと予想しこれは成功であると考えた。

## 7.2 13000 Hz の調整

そして次の下げたい谷である 13000 Hz に目星をつけた。先ほどの結果からノッチフィルターは同様に 0.5 を与えればちょうど良く下がるだろうと予想し、検証した。その結果を以下に示す。

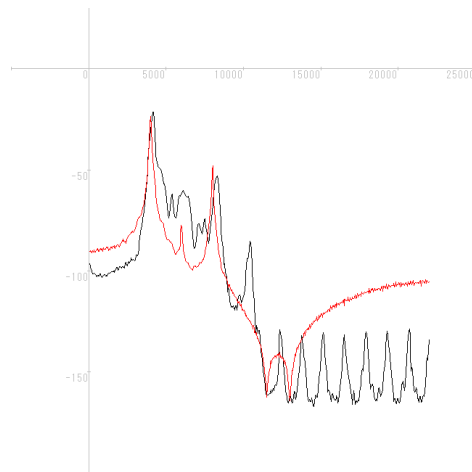


図 18 13000 Hz にノッチフィルターを 0.5 与えた差の平均

この実験では、差の平均 = 23.8509421369863 という結果となった。予想通り先ほどの 11500 Hz もつられて共に下がってくれた。しかし想像以上に 0.5 で 13000Hz が下がってしまったため、次のノッチフィルターの数値は調整しなくてはならないといけないという予想が立った。

### 7.3 14500 Hz の調整

そこで次は調整して次の谷である 14500 Hz のノッチフィルターは 0.7 を与えることにした。その結果を以下に示す。

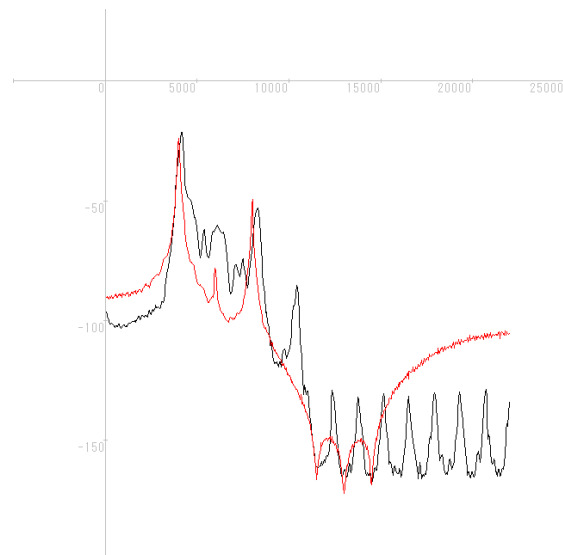


図 19 14500 Hz にノッチフィルターを 0.7 与えた差の平均

この実験では、差の平均 = 21.2791430058708 という結果となった。やはり予想していた通りに 13000 Hz もつられて下がってしまい、目標よりも少し小さくなってしまっ

た。これでも悪くはないのだが、今後も下がっていくと考えるともう少し余裕は持たせておきたいと考え、次はノッチフィルターを 1 与えた。その結果を以下に示す。

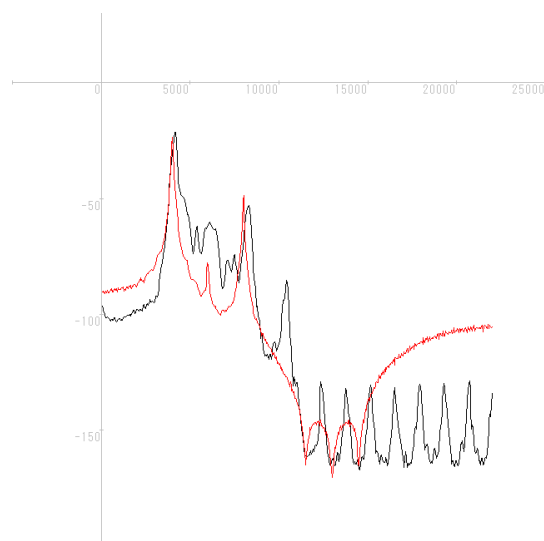


図 20 14500 Hz にノッチフィルターを 1 与えた差の平均

この実験では、差の平均 = 21.6806349178082 という結果となった。先ほどよりは少し上がり、ちょうど良いくらいの所になったと考える。

#### 7.4 16000 Hz の調整

次の谷として 16000 Hz に、これ以上他の周波数は出来れば下げたくないためノッチフィルターは 1 を与えることにする。その結果を以下に示す。

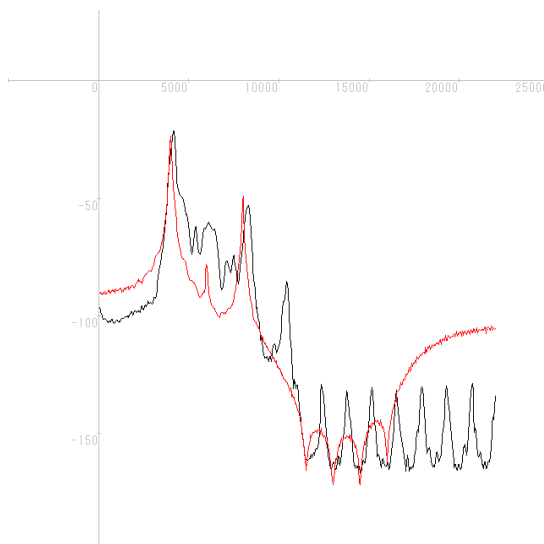


図 21 16000 Hz にノッチフィルターを 1 与えた差の平均

この実験では、差の平均 = 19.7198407318982 となった。16000 Hz の部分は良いのだが、やはり他の周波数が段々つられて下がってしまうのが懸念点となってしまう。しかしここまで差の平均は着実に小さくなっていくため、これを続けて次の谷の調整を行っていく。

## 7.5 17000 Hz の調整

次の谷は 17000 Hz でノッチフィルターは同様に 1 を与えていく。その結果を以下に示す。

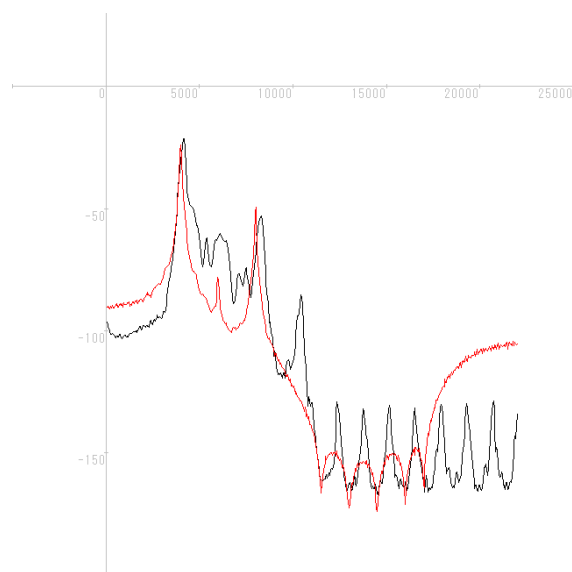


図 22 17000 Hz にノッチフィルターを 1 与えた差の平均

この実験では、差の平均 = 18.2220256634051 となった。17000 Hz は予想通りちょうどこの所まで下がっている。この調子で残りの谷の調整も順に行っていく。次の谷は 18500 Hz でこれも同様にノッチフィルターを 1 与えていった。その結果を以下に示す。

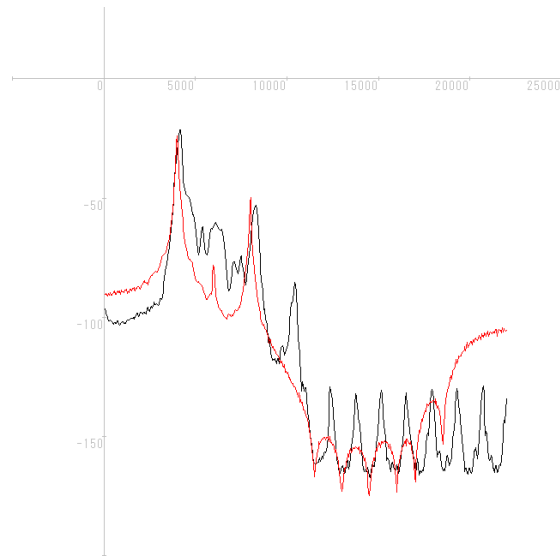


図 23 18500 Hz にノッチフィルターを 1 与えた差の平均

この実験では、差の平均 = 16.8579886223092 となった。ここで少し想定外のことが起こった。今まではノッチフィルターを 1 与えていたら大体ちょうど良いくらいまでさがってくれたのだが、この周波数ではあまり下がらなかった。しかしこれは嬉しい誤算でこの後の調整で一緒に下がって、ちょうど目標くらいまで下がってくれそうな状態であるため、良い結果であったと考える。

## 7.6 20000 Hz の調整

次は 20000 Hz で先ほどのことも考慮して少し強めに下げたほうが良いと想定して、ノッチフィルターを 0.5 与えた。その結果を以下に示す。

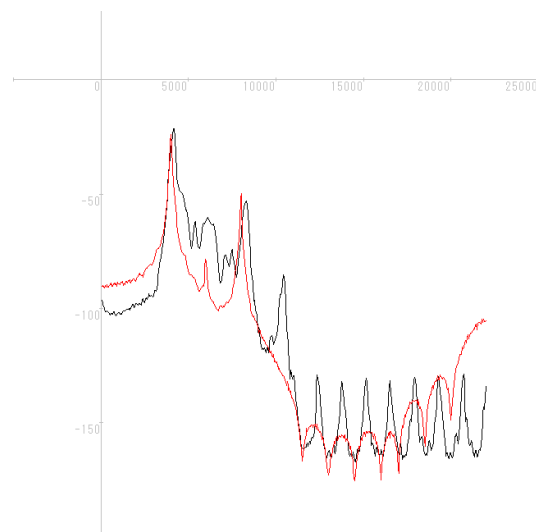


図 24 20000 Hz にノッチフィルターを 0.5 与えた差の平均



この実験では、差の平均 = 15.0915445851272 となった。ここでも想定していた以上に 20000 Hz が下がらなかった。しかし 18500 Hz の部分は想定通りにちょうど目標くらいまで下がってくれたため、これも次で調整できると考えた。

## 7.7 21500 Hz の調整

最後の谷である 21500 Hz には、さらに強めに下げるためにノッチフィルターを 0.1 与えることにした。その結果を以下に示す。

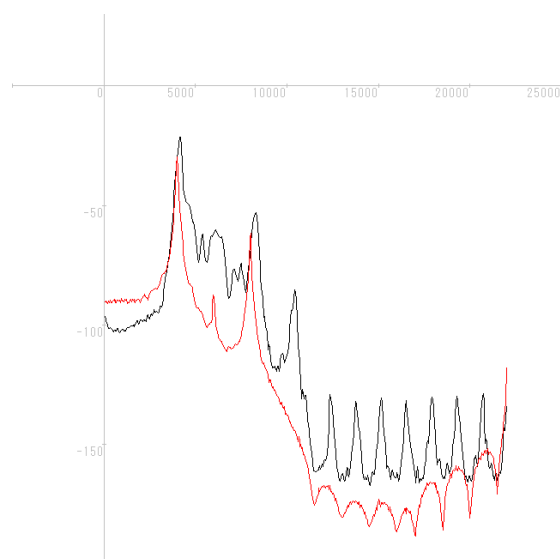


図 25 21500 Hz にノッチフィルターを 0.1 与えた差の平均

この実験では、差の平均 = 19.2564332152642 となった。さすがにこれもやりすぎてしまった。21500 Hz は飛び抜けて下がってしまい、そのほかの周波数もつられて下がり、差の平均も大きくなってしまったためこれは失敗である。このためノッチフィルターは 0.5 だと足りないと想定されるため、今度は 0.25 を与えることにした。その結果を以下に示す。

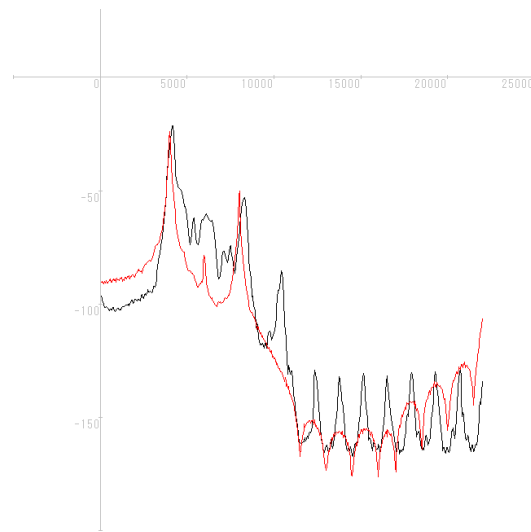


図 26 21500 Hz にノッチフィルターを 0.25 与えた差の平均

この実験では、差の平均 = 13.4907234011742 となった。先ほどと比べると確実に差の平均を小さくすることが出来た。そして 21500 Hz は下がりきってはいないが、他の周波数は概ね本来の鈴虫の音に近づけることができたと考えられる。また先ほどのノッチフィルターを 0.1 与えた際の結果から、4000 Hz や 8000 Hz までもが下がってしまっていたためこれ以上上げるのは全体的なバランスから見ても難しいように感じられる。

## 8. おわりに

最後にこの研究を行なって、当初の目標であった鈴虫の音を実物で再現するというのは出来なかったが、一からパソコン上で人の手で鈴虫の音を再現することはできた。そして鈴虫の音がどのようにしてあの様な音を出していたのかを理解することができ、なぜ鈴虫と呼ばれるようになったのかも改めて知ることが出来た。また残った課題点としては、やはり鈴虫の音を実物で再現することと、まだ改良の余地が残っているパソコン上で再現した鈴虫の音である。パソコン上での鈴虫の再現の改良点は、まだ足りていなかった 6000 Hz 付近の振幅を大きくすることや、4000 Hz よりも低い周波数部分をもう少し抑えること、8000 Hz の次のピーク 10000 Hz のピークを大きくすることなどが挙げられる。そうして鈴虫の音を人工的に再現することが出来れば、どんな虫の音も人工的に再現することは可能であるといえるだろう。

### 【参考文献】

1. <https://lab-brains.as-1.co.jp/enjoy-learn/2023/09/53986/>
2. <https://www.ctv.co.jp/hapiene/lovelabo/2007/1007/index.html>
3. <https://data-viz-lab.com/correlation-coefficient>

【付録】

[スペクトルの相関]

！ スペクトルの相関

PRINT " 2 つの音の波形のスペクトルの相関"

PRINT

PRINT "グラフは 2 つの波形のスペクトル"

LET n=1000

DIM x(n),f(n),g(n),u(n)

MAT U = CON

OPEN #1:NAME "C:\Users\Yrikita\OneDrive\デスクトップ\分析ソフト\見本  
spectrum.CSV", RECTYPE INTERNAL

LET i=1

DO

    READ #1,IF MISSING THEN EXIT DO: x(i),f(i),g(i)

    LET i=i+1

LOOP

CLOSE #1

LET dm=i-1

LET af=DOT(f,u)/dm

LET ag=DOT(g,u)/dm

PRINT dm;af;gf

LET dp=0

LET nf=0

LET ng=0

FOR i=1 TO dm

    LET dp=dp+(f(i)-af)\*(g(i)-ag)

    LET nf=nf+(f(i)-af)^2

    LET ng=ng+(g(i)-ag)^2

NEXT I

```

LET nf=SQR(nf)
LET ng=SQR(ng)
LET r=dp/(nf*ng)
PRINT
PRINT "相関係数 r : ";r

```

```

SET WINDOW -5000,25000,-150,30
DRAW axes(5000,50)
FOR i=1 TO dm
    PLOT LINES: x(i), f(i);
NEXT I
PLOT LINES

```

```

SET LINE COLOR 4
FOR i=1 TO dm
    PLOT LINES: x(i), g(i);
next I

```

```

END

```

[スペクトルの差の平均]  
 ! スペクトルの差の平均

```

PRINT " 2つの音の波形のスペクトルの差の平均"
PRINT
PRINT "グラフは2つの波形のスペクトル"

```

```

LET n=1000
DIM x(n),f(n),g(n),u(n)
MAT U = CON

```

```

OPEN #1:NAME "C:\Users\¥rikit¥OneDrive¥デスクトップ¥分析ソフト¥spectrum.CSV",

```

```

RECTYPE INTERNAL
LET i=1
DO
    READ #1,IF MISSING THEN EXIT DO: x(i),f(i),g(i)
    LET i=i+1
LOOP
CLOSE #1
LET dm=i-1

```

```

LET sa=0
FOR i=1 TO dm
    LET si=ABS(f(i)-g(i))
    LET sa=sa+si
NEXT I
LET m=sa/dm
PRINT
PRINT "差の平均 m : ";m

```

```

SET WINDOW -5000,25000,-200,30
DRAW axes(5000,50)
FOR i=1 TO dm
    PLOT LINES: x(i), f(i);
NEXT I
PLOT LINES

```

```

SET LINE COLOR 4
FOR i=1 TO dm
    PLOT LINES: x(i), g(i);
next I

```

```

END

```



卒業研究

# 交代完全数の研究

先進科学系

澤田晴丈

指導教員 松田修

令和7年1月20日

## 概要

本研究は、完全数の性質を基盤として、真の約数に着目し、新たな法則を探す試みである。完全数とは正の真の約数の和が自分自身になる自然のことである。たとえば、 $6 = 1 + 2 + 3$ ,  $28 = 1 + 2 + 4 + 7 + 14$ より6と28は完全数である。これに対して約数の符号を変更して、 $15 = 5 \times (1 - 3 + 5)$ ,  $105 = 3 \times (1 + 3 - 5 + 7 + 15 - 21 + 35)$ などを考える。この論文で、私たちは、15を(1,5)交代完全数、105を(2,3)交代完全数と呼び、一般に $(k, p)$ 交代完全数を定義した。そして、 $(k, p)$ 交代完全数に関する様々な定理を証明した。

## Abstract

The purpose of this study is to find a new law focusing on true divisors based on the properties of perfect numbers. A perfect number is a nature in which the sum of positive proper divisors is itself. For example, 6 and 28 are perfect numbers from  $6 = 1 + 2 + 3$ ,  $28 = 1 + 2 + 4 + 7 + 14$ . In contrast, changing the sign of the divisor, we have  $15 = 5 \times (1 - 3 + 5)$ ,  $105 = 3 \times (1 + 3 - 5 + 7 + 15 - 21 + 35)$ , and so on. In this paper, we call 15 a (1,5)alternating perfect number and 105 a (2,3)alternating perfect number, and we define  $(k, p)$  alternating perfect numbers in general. In this paper, we shall proof of various theorems on  $(k, p)$  alternating perfect numbers.



## 目次

|                                          |    |
|------------------------------------------|----|
| 1. 本研究の目的と動機.....                        | 1  |
| 2 完全数について.....                           | 1  |
| 3. 交代完全数について.....                        | 1  |
| 4. $(1,p)$ 交代完全数.....                    | 2  |
| 4.1 $(1,3)$ 交代完全数.....                   | 2  |
| 4.2 $(1,5)$ 交代完全数.....                   | 4  |
| 5. $(2,p)$ 交代完全数.....                    | 5  |
| 5.1 $(2,3)$ 交代完全数.....                   | 6  |
| 6. $(k,1)$ 交代完全数 ( $k \geq 1$ ) .....    | 6  |
| 6.2 $n$ 完全数について.....                     | 7  |
| 6.3 完全数と $(k,1)$ 交代完全数と $n$ 完全数の関係 ..... | 8  |
| 7. $(u,1)$ 交代完全数の素数倍の性質.....             | 9  |
| 8. $(1,-3)$ 交代完全数について .....              | 9  |
| 付録 使用したプログラム(10進 BASIC).....             | 11 |

## 1. 本研究の目的と動機

完全数は古代から多くの数学者によって研究されており、その美しさや数論的な得性が注目されてきた。しかし、その膨大な研究の歴史ゆえに、新たな視点や独自性を持った研究が求められている。本研究では従来の枠組みを変え、完全数を派生させた新たな概念を提案し、その性質や応用可能性を探索し完全数に対する理解を深めることを目的としている。

## 2 完全数について

完全数 $n$ とは、 $n$ の正の真の約数の和が $n$ に等しいものをいう。すなわち、自然数 $n$ が、 $d_1 < d_2 < \dots < d_r \neq n$ を  $n$ の正の真の約数に対して、

$$n = d_1 + d_2 + \dots + d_r$$

を満たすとき、 $n$ を完全数というのである。

完全数に関しては、以下のユークリッド・オイラーの定理が有名である。

ユークリッド・オイラーの定理

偶数の完全数は、 $2^{p-1}(2^p - 1)$ の形に限る。ここで、 $2^p - 1$  はメルセンヌ素数である。

## 3. 交代完全数について

完全数の定義に対して、本研究では、考えている自然数の正の真の約数に+と-の符号を交互につけて、その和（交代和という）を考える。そして、この交代和のパターンを解析し、その中に潜む規則性や法則を明らかにする。

まず、自然数 $n$ に対して、 $n$ の正の真の約数を小さい順にならべ、まず+の符号を最初の約数の $k$ 個までつけ、その次の約数の符号を-とし、これを繰り返して符号をつける。そして、符号をつけた約数の和を $\tau_k(n)$ とする。正確な定義は以下である。

**$\tau_k(n)$ の定義**  $n$ : 自然数,  $1 = d_0 < d_1 < \dots < d_r \neq n$ :  $n$ の正の真の約数とする。このとき、

$$\tau_k(n) := u_0 d_0 + u_1 d_1 + u_2 d_2 + \dots + u_r d_r$$

とする。ただし

$$u_j = \begin{cases} -1 & (j \equiv k \pmod{k+1}) \\ 1 & (j \not\equiv k \pmod{k+1}) \end{cases}$$

である。

例)  $\tau_1(6) = 1 - 2 + 3 = 2$ ,  $\tau_2(105) = 1 + 3 - 5 + 7 + 15 - 21 + 35 = 35$

次に,  $(k, p)$ 交代完全数を定義する.  $(k, p)$ 交代完全数 $n$ とは $n = p \cdot \tau_k(n)$ を満たすものである. 正確な定義は以下である.

$(k, p)$ 交代完全数の定義  $n, p$ : 自然数とする.  $n$ が

$$n = p \cdot \tau_k(n)$$

を満たすとき,  $n$ を $(k, p)$ 交代完全数という.

例) 6 は  $(1, 3)$  交代完全数:  $6 = 3 \times (1 - 2 + 3)$

105 は  $(2, 3)$  交代完全数:  $105 = 3 \times (1 + 3 - 5 + 7 + 15 - 21 + 35)$

以下,  $(k, p)$ の値を変えて得られたデータから法則性の推測や定理の証明を行った.

#### 4. $(1, p)$ 交代完全数

$k = 1$ の場合, すなわち $(1, p)$ 交代完全数の研究を行った. 定義を確認すると以下となる.

$n, p$ : 自然数,  $d_0 < d_1 < d_2 < \dots < d_r \neq n$

$$n = p \cdot \tau_1(n) = p \cdot \{d_0 - d_1 + d_2 - \dots + (-1)^{r-1} d_r\}$$

例) 6 は  $(1, 3)$  交代完全数:  $6 = 3 \times (1 - 2 + 3)$

15 は  $(1, 5)$  交代完全数:  $15 = 5 \times (1 - 3 + 5)$

##### 4.1 $(1, 3)$ 交代完全数

$(1, 3)$ 交代完全数の調査結果を述べる.

$(1, 3)$  交代完全数を調べると以下のリストが得られた.

3, 6, 12, 24, 48, 60, 96, 180, 192, 300, 384, 540, 768, 924, 1500, 1512, 1536, 1620, 3072, 4104, 4860, ...

これらを分類したものが以下の表である.

|       |                                                                         |
|-------|-------------------------------------------------------------------------|
| type1 | 3, 6, 12, 24, 48, 96, 192, 384, 768, 1536, 3072, ..., $3 \cdot 2^{n-1}$ |
| type2 | 60, 180, 540, 1620, 4860, ..., $60 \cdot 3^{n-1}$                       |
| type3 | 300, 1500, ..., $300 \cdot 5^{n-1}$                                     |
| type4 | 924, 1512, 4104, ..., ?                                                 |

分類表から以下の定理 1 を発見し証明した

**定理 1**

$n$  を自然数とする. このとき,  $3 \cdot 2^{n-1}$ ,  $60 \cdot 3^{n-1}$ ,  $300 \cdot 5^{n-1}$  は, いずれも (1,3)交代完全数である.

(証明) まず,  $3 \cdot 2^{n-1}$  が (1,3) 交代完全数であることを証明する.  $3 \cdot 2^{n-1}$  の真の約数の交代

和は,

$$\begin{aligned}\tau_1(3 \cdot 2^{n-1}) &= 1 - 2 + 3 - 2^2 + (2 \cdot 3) - 2^3 + \cdots - 2^{n-1} + (2^{n-2} \cdot 3) \\ &= \{1 + 3 + (2 \cdot 3) + \cdots + (2^{n-2} \cdot 3)\} - \{(2 + 2^2 + \cdots + 2^{n-2})\} \\ &= 1 + 3(2^{n-1} - 1) - (2^n - 2) = 2^{n-1}\end{aligned}$$

よって,  $3 \cdot 2^{n-1} = 3\tau_1(3 \cdot 2^{n-1})$  である. したがって,  $3 \cdot 2^{n-1}$  が (1,3) 交代完全数である.

次に,  $60 \cdot 3^{n-1}$  が (1,3) 交代完全数であることを証明する. そのために, 以下の補題 1 と補題 2 が必要である.

**補題 1.**  $n \geq 4$  のとき,  $a_n = 3^n \cdot 20$  と  $a_{n-1} = 3^{n-1} \cdot 20$  の約数は,  $a_{n-3} = 3^{n-3} \cdot 20$  まで同じである.

(補題 1 の証明)  $a_n$  と  $a_{n-1}$  の約数の中に  $a_{n-3}$  の約数が全て含まれることは明らかである. しかし,  $a_n$  の約数である  $3^n$  は,  $a_{n-3} < 3^n$  であるが  $a_{n-1}$  の約数ではない. よって, 示された.

**補題 2.**  $n \geq 3$  のときの  $a_n = 3^n \cdot 20$  の約数について  $a_{n-3} = 3^{n-3} \cdot 20$  までの約数の交代和は  $-3^{n-1}$  である.

(補題 2 の証明)  $n = 3$  のとき,  $a_3$  の約数について  $a_0 = 20$  までの約数の交代和  $\tau$  は

$$\tau = 1 - 2 + 3 - 4 + 5 - 6 + 9 - 10 + 12 - 15 + 18 - 20 = -9 = -3^2$$

であるので成り立つ.

$n \geq 4$  のとき,  $n = k - 1$  まで成り立つと仮定する. すなわち  $a_{k-4} = 3^{k-4} \cdot 20$  までの約数の交代和は  $-3^{k-2}$  であるとする. このとき,  $a_{k-4}$  の約数の個数は  $6(k-3)$  なので, 約数  $a_{k-4}$  の符号は  $-$  であることに注意する.  $a_k$  の  $a_{k-3} = 3^{k-3} \cdot 20$  までの交代和  $\tau$  を考える.  $\tau_1$  を  $a_{k-4}$  までの交代和とすると帰納法の仮定より  $\tau_1 = -3^{k-2}$  である.  $a_k$  の約数で  $a_{k-4}$  以降の約数の交代和を  $\tau_2$  とすると,  $\tau_2$  で使う約数の個数は 6 個で,

$$3^{k-1} < 2 \cdot 5 \cdot 3^{k-3} < 2^2 \cdot 3^{k-2} < 5 \cdot 3^{k-2} < 2 \cdot 3^{k-1} < 2^2 \cdot 5 \cdot 3^{k-3}$$

である. したがって,

$$\begin{aligned}\tau_2 &= 3^{k-1} - 2 \cdot 5 \cdot 3^{k-3} + 2^2 \cdot 3^{k-2} - 5 \cdot 3^{k-2} + 2 \cdot 3^{k-1} - 2^2 \cdot 5 \cdot 3^{k-3} \\ &= 3^{k-1}(1 + 2) + 3^{k-2}(2^2 - 5) - 3^{k-3}(2 \cdot 5 + 2^2 \cdot 5) = 3 \cdot 3^{k-1} - 3^{k-2} - 30 \cdot 3^{k-3} \\ &= -2 \cdot 3^{k-2}\end{aligned}$$

よって,  $\tau = \tau_1 + \tau_2 = -3^{k-2} - 2 \cdot 3^{k-2} = -3^{k-1}$ . ゆえに数学的帰納法により補題 2 は成り立つ.

$60 \cdot 3^{n-1}$  が (1,3) 交代完全数であることの証明を続ける.

①  $60$  と  $60 \cdot 3$  と  $60 \cdot 3^2$  が (1,3) 交代完全数であることは実際の計算からわかる.

②  $n \geq 4$  とする. 補題 2 より,  $60 \cdot 3^{n-1}$  の約数の  $60 \cdot 3^{n-3}$  までの交代和  $\tau_1$  は  $-3^{n-1}$  である. したがって,  $60 \cdot 3^{n-1}$  の  $60 \cdot 3^{n-3}$  以後の真の約数の交代和  $\tau_2$  は

$$\begin{aligned}\tau_2 &= 3^n - 2 \cdot 5 \cdot 3^{n-2} + 2^2 \cdot 3^{n-1} - 5 \cdot 3^{n-1} + 2 \cdot 3^n - 2^2 \cdot 5 \cdot 3^{n-2} + 2 \cdot 5 \cdot 3^{n-1} \\ &\quad - 2^2 \cdot 3^n + 5 \cdot 3^n - 2^2 \cdot 5 \cdot 3^{n-1} + 2 \cdot 5 \cdot 3^n \\ &= 3^n(1 + 2 - 2^2 + 5 + 2 \cdot 5) + 3^{n-1}(2^2 - 5 + 2 \cdot 5 - 2^2 \cdot 5) - 3^{n-2}(2 \cdot 5 + 2^2 \cdot 5) \\ &= 14 \cdot 3^n - 11 \cdot 3^{n-1} - 30 \cdot 3^{n-2} = 21 \cdot 3^{n-1}\end{aligned}$$

よって,  $60 \cdot 3^{n-1}$  の真の約数の交代和  $\tau_1(60 \cdot 3^{n-1})$  は

$$\tau_1(60 \cdot 3^{n-1}) = \tau_1 + \tau_2 = -3^{n-1} + 21 \cdot 3^{n-1} = 20 \cdot 3^{n-1}$$

従って,  $60 \cdot 3^{n-1} = 3 \cdot \tau_1(60 \cdot 3^{n-1})$  が成り立つので,  $60 \cdot 3^{n-1}$  は (1,3) 交代完全数である.

$300 \cdot 5^{n-1}$  が (1,3) 交代完全数であることも同様に証明できる.

## 4.2 (1.5)交代完全数

(1,5)交代完全数の調査結果を述べる.

(1,5) 交代完全数を調べると以下のリストが得られた.

5, 15, 45, 135, 315, 405, 495, 1215, 2205, 3645, 5445, 10935, 15435,  
32805, 59895, 108045, 658845, 98415, 295245, 885735, 2657205,  
7247295, 7971615, ...

これらを分類したものが以下の表である.

|        |                                                                       |
|--------|-----------------------------------------------------------------------|
| Type1  | 5, 15, 45, 135, 405, 1215, 3645, 10935, 32805, ..., $5 \cdot 3^{n-1}$ |
| Type2  | 315, 2205, 15435, 108045, 756315, 5294205, ..., $315 \cdot 7^{n-1}$   |
| Type3  | 495, 5445, 59895, 658845, 7247295, ..., $495 \cdot 11^{n-1}$          |
| others | ?                                                                     |

分類表から以下の定理 2 を発見し, 証明した.

**定理 2**

$n$  を自然数とする. このとき,  $5 \cdot 3^{n-1}$ ,  $315 \cdot 7^{n-1}$ ,  $495 \cdot 11^{n-1}$  は, いずれも  $(1,5)$  交代完全数である.

定理 2 は定理 1 と同様な方法で証明できる.

さて,  $3 \cdot 2^{n-1}$  は  $(1,3)$  交代完全数,  $5 \cdot 3^{n-1}$  は  $(1,5)$  交代完全数である. このとき,  $2 = \frac{3+1}{2}$ ,  $3 = \frac{5+1}{2}$  と考える.

**定理 3**

$p$  を素数,  $a = \frac{p+1}{2}$  も素数とする. このとき,  $pa^{n-1}$  は,  $(1,p)$  交代完全数である.

(証明)  $pa^{n-1}$  の約数は,  $1, a, p, a^2, pa, \dots, pa^{n-2}, a^{n-1}$  であり

$$\begin{aligned}\tau_1(pa^{n-1}) &= 1 - a + p - a^2 + pa - \dots + pa^{n-2} - a^{n-1} \\ &= 1 + \frac{p(a^{n-1} - 1)}{a - 1} - \frac{a(a^{n-1} - 1)}{a - 1} \\ &= 1 + \frac{(2a - 1)(a^{n-1} - 1)}{a - 1} - \frac{a(a^{n-1} - 1)}{a - 1} = a^{n-1}\end{aligned}$$

である. よって,  $pa^{n-1}$  は  $(1,p)$  交代完全数である.

**5.  $(2,p)$  交代完全数**

まず,  $(2,p)$  交代完全数の定義を確認する.

$n, p$  : 自然数,  $d_1 < d_2 < \dots < d_r \neq n$  :  $n$  の正の約数とする.

$$n = p \cdot \tau_2(n) = p \cdot \{u_0 d_0 + u_1 d_1 + u_2 d_2 + \dots + u_r d_r\}$$

ここで

$$u_j = \begin{cases} -1 & (j \equiv 2 \pmod{3}) \\ 1 & (j \not\equiv 2 \pmod{3}) \end{cases}$$

例) 105 は  $(2,3)$  交代完全数 :  $105 = 3 \times (1+3-5+7+15-21+35)$

189 は  $(2,3)$  交代完全数 :  $189 = 3 \times (1+3-7+9+11-27+63)$

## 5.1 (2,3)交代完全数

(2,3)交代完全数の調査結果をみると、双子素数に関係することがわかった。以下にその例を挙げる。

$$3 \cdot 5 \cdot 7 = 105 = 3 \times (1 + 3 - 5 + 7 + 15 - 21 + 35)$$

$$3^3 \cdot 7 = 189 = 3 \times (1 + 3 - 7 + 9 + 21 - 27 + 63)$$

$$3^3 \cdot 11 = 297 = 3 \times (1 + 3 - 9 + 11 + 27 - 33 + 99)$$

$$3 \cdot 11 \cdot 13 = 429 = 3 \times (1 + 3 - 11 + 13 + 33 - 39 + 143)$$

$$3 \cdot 17 \cdot 19 = 969 = 3 \times (1 + 3 - 17 + 19 + 51 - 57 + 323)$$

$$3 \cdot 29 \cdot 31 = 2697 = 3 \times (1 + 3 - 17 + 19 + 51 - 57 + 323)$$

(5, 7), (11, 13), (17, 19), (29, 31)は双子素数であるが、これら以降の(2,3)交代完全数もその多くが3・双子素数であった。これより、以下の定理4を発見し、証明した。

### 定理4

$(p, p+2)$  を双子素数で、 $p \neq 2, 3$ とする。このとき、 $3p(p+2)$  は (2,3)交代完全数である。

(証明)  $3p(p+2)$ の約数は、 $1, 3, p, 3p, (p+2), 3(p+2), p(p+2)$ であり

$$\begin{aligned}\tau_2(3p(p+2)) &= 1 + 3 - p + (p+2) + 3p - 3(p+2) + p(p+2) \\ &= p(p+2)\end{aligned}$$

である。よって、 $3p(p+2)$ は(2,3)交代完全数である。

## 6. (k,1)交代完全数 ( $k \geq 1$ )

まず、(k,1)交代完全数の定義を確認する。

$n, p$  : 自然数,  $d_1 < d_2 < \dots < d_r \neq n$  :  $n$ の正の約数とする。

$$n = p \cdot \tau_k(n) = p \cdot \{u_0 d_0 + u_1 d_1 + u_2 d_2 + \cdots + u_r d_r\}$$

ここで

$$u_j = \begin{cases} -1 & (j \equiv k \bmod k+1) \\ 1 & (j \not\equiv 0 \bmod k+1) \end{cases}$$

例) 6 は(3,1) 交代完全数:  $6 = 1+2+3$

40 は(3,1) 交代完全数:  $40 = 1+2+4+5+8+10+20$

### 6.1 (3,1)交代完全数について

(3,1)交代完全数のリストは以下である.

6, 40, 42, 54, 66, 78, 102, 114, 138, 174, 186, 222, 246, 258,  
282, 294, 318, 354, 366, 402, 426, 438, 474, 486, ...

このリストから  $p$  を 7 以上の奇素数としたとき,  $6p$  が(3,1)交代完全数になっていることがわかり, 以下の定理 5 を作り証明した.

#### 定理 5

$p \geq 7$  を素数とする.このとき,  $6p$  は (3,1) 交代完全数である.

(証明)  $6p$  の約数は  $1, 2, 3, 6, p, 2p, 3p$  であり,

$$\tau_3(6p) = 1 + 2 + 3 + 6 + p + 2p + 3p = (1 + 2 + 3)p = 6p$$

よって,  $6p$  は (3,1) 交代完全数である.

### 6.2 $\sqrt{n}$ 完全数について

(3,1) 交代完全数のリストを別の視点から見ると, それは  $\sqrt{n}$  完全と関係することが分かった.

$\sqrt{n}$ 完全数の定義  $n$  を自然数,  $d_i$  を  $n$  の真の約数とし,  $\sqrt{n} < d_i$  とする. このとき

$$\sum_i d_i = n$$

を満たすならば,  $n$  を  $\sqrt{n}$  完全数と呼ぶ.

(例) 42 は, 真の約数で  $\sqrt{42} \sim 6.5$  以上であるものの和は,  $7 + 14 + 21 = 42$  なので,  $\sqrt{n}$  完全数である.



### 研究結果

(3,1) 交代完全数の中の以下のリストは, 全て $\sqrt{n}$ 完全数であった.

42, 54, 66, 78, 102, 114, 138, 174, 186, 222, 246, 258,  
282, 318, 354, 366, 402, 426, 438, 474, ...

### 6.3 完全数と $(k, 1)$ 交代完全数と $\sqrt{n}$ 完全数の関係

$p$ を素数,  $a_r$ を $r+1$ 個の約数をもつ完全数とする. 以下の定理が成り立つ.

#### 定理 6

$a_r$  を $r$ 個の真の約数をもつ完全数,  $p \geq a_r$  を素数とする. このとき,  $a_r p$  は  $(r, 1)$ 交代完全数である.

(証明)  $a_r p$ の約数は,  $1, d_1, \dots, d_{r-1}, a_r, p, pd_1, \dots, pd_{r-1}$

$$\begin{aligned}\tau_r(a_r p) &= 1 + d_1 + \dots + d_{r-1} - a_r + p + pd_1 + \dots + pd_{r-1} \\ &= (1 + d_1 + \dots + d_{r-1})p = a_r p\end{aligned}$$

よって,  $a_r p$  は  $(r, 1)$  交代完全数である.

さらに,  $a_r p$ は $\sqrt{n}$ 完全数であることを以下の定理 7 で証明した.

#### 定理 7

$a_r$  を $r$ 個の真の約数をもつ完全数,  $p \geq a_r$  を素数とする. このとき,  $a_r p$  は  $\sqrt{n}$ 完全数である.

(証明)  $d_i$ を完全数 $a_r$ の約数とする. つまり

$$a_r = 1 + d_1 + \dots + d_{r-1}$$

ただし,  $d_0 = 1$ とした.  $a_r \leq p$ より $a_r p$ の $\sqrt{a_r p}$ より大きい約数は,

$$p, pd_1, \dots, pd_{r-1}$$

である. よって,

$$\tau_r(a_r p) = (1 + d_1 + \dots + d_{r-1})p = a_r p$$

したがって,  $a_r p$  は  $\sqrt{n}$ 完全数である.

例) 完全数6に対して,  $6p$ は $(3,1)$ 交代完全数であり $\sqrt{n}$ 完全数でもある. (ただし $p > 6$ , 6 は $(3,1)$ 交代完全数)

完全数28に対して,  $28p$ は $(5,1)$ 交代完全数であり $\sqrt{n}$ 完全数でもある. (ただし $p > 28$ , 28 は $(5,1)$ 交代完全数)

## 7. $(u, 1)$ 交代完全数の素数倍の性質

$(u, 1)$ 交代完全数 $a_r$ を考える. つまり,  $a_r$ は $d_i$ を $a_r$ の約数としたとき

$$a_r = \tau_u(a_r)$$

が成り立つ.

### 定理 8

$u + 1$ を $r + 1$ の約数とする. そして,  $a_r$  を $r$ 個の真の約数をもつ $(u, 1)$ 交代完全数, さらに,  $p > a_r$  を素数とする. このとき,  $a_r p$  は  $(u, 1)$ 交代完全数である.

(証明)  $a_r p$ の約数は,  $1, d_1, \dots, d_{r-1}, a_r, p, p d_1, \dots, p d_{r-1}$ で,  $a_r$ は $(u, 1)$ 交代完全数なので,  $\tau_u(a_r) = a_r$ である.  $u + 1$ は $r + 1$ の約数なので,

$$\tau_u(a_r p) = \tau_u(a_r) - a_r + \tau_u(a_r) p = a_r p$$

したがって,  $a_r p$  は  $(u, 1)$  交代完全数である.

例)  $40p$ は $(3, 1)$ 交代完全数 (ただし $p > 40$ ,  $40$  は $c$   $(3, 1)$ 交代完全数)

より一般に, 次の定理を証明することができた.

### 定理 9

$a_r p$  は定理 8 から得られた  $(u, 1)$ 交代完全数とし,  $q > a_r p$  を素数とする. このとき,  $a_r p q$  は  $(u, 1)$ 交代完全数である.

(証明)  $\tau_u(a_r p) = a_r p$ より

$$\tau_u(a_r p q) = \tau_u(a_r p) - a_r p + \tau_u(a_r p) q = a_r p q$$

したがって,  $a_r p q$  は  $(u, 1)$  交代完全数である.

例)  $40pq$ は $(3, 1)$ 交代完全数 (ただし $q > 40p$ ,  $p > 40$ ,  $40$  は $(3, 1)$ 交代完全数)

## 8. $(1, -3)$ 交代完全数について

$3 \cdot 2^n$ は $(3, 1)$ 交代完全数である. なぜならば,

$$\tau_1(3 \cdot 2^n) = 1 - 2 + 3 - 4 + 6 - 8 + \dots + 3 \cdot 2^{n-1} = 2^n$$

なので,  $3 \cdot 2^n = 3 \cdot \tau_1(3 \cdot 2^n)$  が成り立つからである. そこで,  $p > 3 \cdot 2^n$  として

$3 \cdot 2^n p$  を考える. このとき  $3 \cdot 2^n p$  はどのような交代完全数なのかを研究した.

**定理 10**  $n$  を自然数,  $p$  を素数,  $3 \cdot 2^n$ ,  $p > x$  とする. このとき  $3 \cdot 2^n p$  は  $(1, \frac{3p}{p-2})$  交代完全数である.

(証明)  $p > 3 \cdot 2^n$  であることに注意する.  $S_0$  を  $3 \cdot 2^n$  を含む約数の交代和とおく

と,

$$S_0 = 1 - 2 + 3 - 4 + \cdots + 3 \cdot 2^{n-1} - 3 \cdot 2^n = -2^{n+1}$$

よって,

$$\tau_1(x) = S_0 + pS_0 + 3 \cdot 2^n p = -2^{n+1} - 2^{n+1}p + 3 \cdot 2^n p = -2^{n+1} + 2^n p$$

したがって,

$$\begin{aligned} \{x - 3\tau_1(x)\}p &= \{3 \cdot 2^n p - 3(-2^{n+1} + 2^n p)\}p \\ &= \{3 \cdot 2^{n+1} + 3 \cdot 2^n p - 3 \cdot 2^n p\}p \\ &= 3 \cdot 2^{n+1}p = 2x \end{aligned}$$

よって,  $x = \frac{3p}{p-2} \tau_1(x)$  が成り立つ.

(証明終)

付録 使用したプログラム(10 進 BASIC)

### 8.1( $k, p$ )交代完全数

```
FOR N=1 TO 10000
! PRINT "n=";n
  LET n1=INT(n/2)
  LET S=0
  LET j=1
  FOR i=1 TO n1 STEP 1
    LET i1=i
    IF MOD(n,i)=0 THEN
      IF MOD(j,Q)=0 THEN !Q に計算したい k+1 を入れ
る
        LET i1=i1*(-1)
      ELSE
        LET i1=i1
      END IF
      LET S=S+i1
      LET j=j+1
    END IF
  NEXT I
  LET S=ABS(S)
  ! PRINT
  IF p*S=n THEN !p に計算したい数を入れる
    PRINT n;"(k,p)交代完全数"
  END IF
NEXT N
END
```

## 8.2 $\sqrt{n}$ 完全数

```
FOR N=1 TO 10000
! PRINT "n=";n
  LET n1=INT(n/2)
  LET S=0
  LET A=0
  LET j=1
  FOR i=1 TO n1 STEP 1
    LET i1=i
    IF MOD(n,i)=0 THEN

      LET S=S+i1
      LET S=ABS(S)
      LET K=SQR(N)
      LET K=ABS(K)
      IF K<i1 THEN
        LET A=A+i1
      END IF
      LET j=j+1

    END IF
  NEXT I

  IF A=n THEN
    PRINT n;"SQR(N)完全数"
  ELSE
    ! PRINT n;"×"
  END IF
NEXT N

END
```

### 8.3 $\sqrt{n}$ 完全数から(3,1)交代完全数を探すプログラム

```
FOR N=1 TO 1000
!  PRINT "n=";n
    LET n1=INT(n/2)
    LET S=0
    LET A=0
    LET j=1

    FOR i=1 TO n1 STEP 1
        LET i1=i
        IF MOD(n,i)=0 THEN
            LET K=SQR(N)
            LET K=ABS(K)
            IF K<i1 THEN
                LET A=A+i1
            END IF
            IF MOD(j,4)=0 THEN
                LET i1=i1*(-1)
            ELSE
                LET i1=i1
            END IF
            LET S=S+i1
            LET S=ABS(S)

            LET j=j+1

        END IF
    NEXT I

    IF A=n THEN
        IF A=S THEN
            PRINT n;"SQR(N)(3,1)交代完全数"
        ELSE
            PRINT n;"×"
```

```
        END IF
    END IF
NEXT  N

END
```

#### 8.4(3,1)交代完全数から $\sqrt{n}$ 完全数を探すプログラム

```
FOR N=1 TO 10000
!  PRINT "n=";n
    LET n1=INT(n/2)
    LET S=0
    LET A=0
    LET j=1
    FOR i=1 TO n1 STEP 1
        LET i1=i
        IF MOD(n,i)=0 THEN
            IF MOD(j,4)=0 THEN
                LET i1=i1*(-1)
            ELSE
                LET i1=i1
            END IF

            LET S=S+i1
            LET S=ABS(S)
            LET K=SQR(N)
            LET K=ABS(K)
            IF K<i1 THEN
                LET A=A+i1
            END IF
            LET j=j+1

        END IF
    NEXT I

    IF S=n THEN

        IF A=S THEN
            PRINT n;"(3,1)SQR(N)交代完全数"
        ELSE
            PRINT n;"×"
```



```
        END IF
    END IF
NEXT  N

END
```

### 8.5 6p 完全数

```
LET N=7
FOR I=N TO 2000
  LET PRIME=1
  FOR j=2 TO SQR(I)
    IF MOD(i , j) =0 THEN
      LET prime=0
      EXIT FOR
    END IF
  NEXT j
  IF prime=1 THEN PRINT 6*I
NEXT I
END
```

### 8.6 (3,1)交代友愛数

```
FOR N=1 TO 10000
  LET N1=INT(N/2)
  LET S=0
  LET j=1
  FOR i=1 TO N1 STEP 1
    LET i1=i
    IF MOD(N,i)=0 THEN
      IF MOD(j,4)=0 THEN      !(k,1)
        LET i1=i1*(-1)
      ELSE
        LET i1=i1
      END IF
      LET S=S+i1
      LET j=j+1
    END IF
  NEXT i
  LET S=ABS(S)

  LET S1=INT(S/2)
  LET V=0
  LET q=1
  FOR a=1 TO S1 STEP 1
    LET a1=a
    IF MOD(S,a)=0 THEN
      IF MOD (q,4)=0 THEN      !(k,1)
        LET a1=a1*(-1)
      ELSE
        LET a1=a1
      END IF
      LET V=V+a1
      LET q=q+1
    END IF
  NEXT a
```

LET V=ABS(V)

IF V=n THEN

IF n=S THEN

! PRINT n;S;"(3,1)交代完全数"

ELSE

PRINT n;S;"(3,1)交代友愛数"

END IF

END IF

NEXT N

END