

# ICUHS 数学ツアー2019

H. Suzuki (鈴木寛)

August 9, 2019

## 1 はじめに

### 1.1 数学ツアー案内

2019年8月29日・30日

タイトル：NAND回路の設計とラグランジュの補間公式  
サブタイトル：～数学的思考とその応用～

電子回路は、ゲートといわれるものを組み合わせて作ります。ゲートのひとつが NAND と呼ばれるもので、2つの入力と、1つの出力からなり、2つの入力が両方とも ON (通常1であらわします) のときのみ、OFF (通常0であらわします)、それ以外は、ON (1) を出力します。この NAND ゲートだけをいくつか使い線ですな

いで、足し算を計算する回路を作ること考えたいと思います。  
LOGIC LAB: <http://www.neuroproductions.be/logic-lab/> (ブラウザーによっては、設定で Flash Player を使えるようにする必要があります)

平面上に二点が与えられているとその二点を結ぶ直線が一本あります。三点が与えられていると、その三点を通る、二次関数は、いつでもありますか。それは、ひとつだけですか。点の数を、四点、五点と増やしていくとどうなるのでしょうか。考えてみましょう。

この2つの全くことなると思われる問題の背後にある、数学の考え方のいくつかを、一緒に学ぶことができればと考えています。

### 1.2 コンピュータ・ツールについて

ツールが必要な場合には、そのときに説明します。核となる部分は、数学ですから、コンピュータ・ツールは使わなくても学習できますが、使うことで視覚化でき、理解もしやすくなりますし、発想の幅も広がります。ただ、なれるのには、多少の時間がかかりますので、少し、触っておくことはたいせつです。二日間で、考えられることは時間の関係で限られますが、ツールにもなれておくことで、自分で問題を作って、考えることもでき、発展性もあります。得意な人もそうでない人も、少し、世界を広げるといった気持ちで、この機会に、使ってみてはいかがでしょうか。

大きく分けて三種類のツールを紹介します。1つ目は、論理回路に関係するもの、2つ目は、グラフを描くためのもの、3つ目は、文書を書くためのものです。3つ目は、みなさんは、基本的に使いませんが、わたしがどのようにして、数式などを書いているかを知るとは、有益だと思うのと、数学やデータ・サイエンスを利用する分野で文章を書く時には、基本的なツールですので、簡単に紹介しておきます。

すべてフリーのソフトで、かつインターネット上で、オンラインで使えるものです。アカウントを作ったほうが便利なものもありますが、1つ目と2つ目に関しては、それらも必要ありません。

#### 1.2.1 論理回路設計

理解を助けるツールです。たくさんありますが、二種類、紹介します。

- Logic Lab: <http://www.neuroproductions.be/logic-lab/> (<http://www.neuroproductions.be/logic-lab/>)
- logic.ly: <https://logic.ly> (<https://logic.ly>)
  - Online Trial Page: <https://logic.ly/demo> (<https://logic.ly/demo>)

英語のサイトですが、オンラインで使えますので、コンピュータにインストールする必要もなく、かつ直感的に使えます。入力 (input) は、ON または OFF (1 または 0) と考えても構いません。論理ゲート (logic gate, logic port) を通ると、入力によって変化します。出力 (output) で、実際に ON なのか、OFF なのかを確認できます。基本的にこれら 3 つのものを線でつないで作ります。

Logic Lab と Logic.ly で多少こととなりますが、殆ど同じ機能を持っています。(Logic Lab は、Adobe の Flash Player を使っていますので、特に、iOS すなわち、Apple 系の Tablet や、Cell Phone などでは、うまく動作しないかもしれません、そのときは、Logic.ly を使ってみてください。ただし、無償版では保存はできません。) Logic Lab は、古くからある Web Application で、わたしは、長く使っています。自分で設計した回路を保存することもできます。Logic.ly のほうが新しく、発展性もあるようですが、Online 版では保存はできません。わたしは、(8月に入ってから) 30日間の無償トライアル版を使っています。Windows 版と、MacOS 版があり、30日間は、すべての機能が使えるようで、保存もできます。

数学ツアーでは、NAND ゲートだけを使って、加算器を作ることを目標としていますが、他のゲート (Logic Lab では Port と呼ばれています) もぜひ使ってみてください。

Flip-Flops も使えるようになっていきます。今回は、時間の関係で触れることはできないと思いますが、基本的なものですので、Flip-Flop とはどのようなもので、何に使われるのか、調べてみるのもよいでしょう。

論理ゲートや、Flip-Flop や、実際のゲートが電子的にどのような構造になっているのかなどの情報は、インターネットで調べることができます。二箇所だけ上げておきます。

- TOSHIBA e-Learning: [https://toshiba.semicon-storage.com/jp/design-support/e-learning/micro\\_intro/chap1/1274740.html](https://toshiba.semicon-storage.com/jp/design-support/e-learning/micro_intro/chap1/1274740.html) ([https://toshiba.semicon-storage.com/jp/design-support/e-learning/micro\\_intro/chap1/1274740.html](https://toshiba.semicon-storage.com/jp/design-support/e-learning/micro_intro/chap1/1274740.html))
- 頼れるエンジニアの知恵袋: <https://www.chip1stop.com/sp/knowledge/top> (<https://www.chip1stop.com/sp/knowledge/top>)

## 1.2.2 グラフ・ツール

関数のグラフを描画するツールです。二つ上げておきます。

- Desmos: <https://www.desmos.com/calculator> (<https://www.desmos.com/calculator>)
- GeoGebra: <https://www.geogebra.org/graphing> (<https://www.geogebra.org/graphing>)

Desmos は、関数のグラフの描画に特化したもので、GeoGebra は幾何にも使えるようになっていきます。今回は、Desmos をほんの少し使う予定ですが、殆どのは、どちらでも可能です。機能も充実しており、学習にも、教育にも有効だと思います。

## 1.2.3 文書作成ツール

数学の文章は、数式を整えて見やすく書く必要があること、文書全体の論理構造がきちりしていることがたいせつですが、数学者の D. Knuth が開発し、無償公開した、 $T_E X$  (テック) という組版システムが、数学関連の論文や書籍の殆どで使われています。 $T_E X$  は、本体だけでなく、フォントを作成し管理するシステムや、 $T_E X$  プログラム自体を改善し使い易くする、マクロと呼ばれるものが最初の設計段階から整備されていたり、説明をも加えられる、web という形式で書かれたこともあり、Reproducible Research (研究の再現性) や、Literate Programming (プログラムの文書化) という、研究や、ソフトウェア開発にも欠かせない分野に大きな影響を与えました。

$TeX$  を使いやすくするために、やはり、数学者の L. Lamport によるマクロを付け加えた、 $\LaTeX$  (ラテック) の名前で呼ばれることもあります。

この文章は、web の考え方を発展させ、データ・サイエンスで主要なソフトとして使われている、R などとの連携を考えて開発された、Rmarkdown という形式で書かれています。この形式で書くと、ホームページなどを記述する、HTML (hypertext markup language) 形式のほか、Microsoft Word や、PDF にも簡単に出力できるようになっています。Rmarkdown は、中にプログラムを書き込み、その実行結果とともに、文書を作成できるようになっており、プログラムもいろいろな言語に対応しています。プログラムを文書に加え、その結果を出力するだけでなく、そのプログラムの解説も、同時に作成することができ、データを書き換えても、グラフなども同時に書き換えることができるので、データ・サイエンスにおける、再現性 (Reproducibility) にはなくてはならないものになっています。今回は、データ分析も、プログラミングも行いませんが、いろいろな形式の文書として、書き出すことができる利点を考えて、Rmarkdown で書き始めています。Rmarkdown は、テキストで文書を作成しますが、RStudio と連携させると、特に、データ分析のときには、便利です。他に似たものとしては、Python というコンピュータ言語に付随して開発された、Jupyter と呼ばれる、Python Notebook が有名です。

- RStudio: <https://www.rstudio.com> (<https://www.rstudio.com>)
- RMarkdown: <https://rmarkdown.rstudio.com> (<https://rmarkdown.rstudio.com>)
- $TeX$ : <https://texwiki.texjp.org> (<https://texwiki.texjp.org>)
- Online  $\LaTeX$ : <https://cloudlatex.io/ja> (<https://cloudlatex.io/ja>)
- Cocalc: <https://cocalc.com> (<https://cocalc.com>)
- SageCell: <https://sagecell.sagemath.org> (<https://sagecell.sagemath.org>)

Cocalc は、 $\LaTeX$  も Rmarkdown も Jupiter も使えますし、数式の計算もできるもので、最初は、SageMath の一部として開発されたものです。Cocalc は、Collaborative Calculation からとったもので、共同作業に適した環境を提供しています。最後の SageCell は、電卓のようにして使える、SageMath (商用の Mathematica や Maple などの代わりになるものとして開発されました) です。

Rmarkdown の中でも、数式は、 $TeX$  形式で書いています。

## 1.3 問題について

### 1.3.1 論理回路の問題

The Logic Lab で説明してみます。ページの下に、Logic Ports と書いてあるものがあります。それは、この[リンク] (<http://www.neuroproductions.be/logic-lab/index.php?id=104983>) のようになっています。すべてにスイッチがついていて、白になっていれば、OFF、赤になっていれば ON です。それを動かして、AND, NOR, NOT, NAND, XOR, OR, XNOR それぞれで、入力がどうなっているときに、電気が赤くつくかを調べてみてください。案内に書いたように、NAND は、両方が、OFF のときのみ ON になっていることが確かめられると思います。他のものは、それぞれどうなっていますか。

Logic.ly の場合には、ある回路を選択すると、Truth Table (真理表) を作ってくれる機能もあります。上で書いた、ON または 1 が TRUE で、OFF または 0 が FALSE に対応しています。入力の値によって、出力がどうなるかが読み取れるようになっています。

加算器を作りたいのですが、2進演算を考えます。簡単なものから始めます。

- $0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 10_2$  です。最後だけ二桁になっています。添字の2は、2進であることを示したものです。繰り上がりを Carry と言います。ですから、みな二桁だと考えて、
- $0 + 0 = 00_2, 0 + 1 = 01_2, 1 + 0 = 01_2, 1 + 1 = 10_2$  と出力を二桁にしてみましょう。

- これを、いくつかのゲート（ポート）を組み合わせて作ってみてください。
- Output に2進4桁のカウンターもありますから、数字で答えを出すことも可能です。Input も数字で出せるとわかりやすいですね。
- 時間があれば、最終的には、2進3桁+2進3桁を計算できる加算器を作りたいと思います。2進3桁は、10進では0,1,2,3,4,5,6,7ですね。
- ちょっと難しいのは、これを、NAND ゲート（Port）だけで作ってみようという部分ですが、まずは、他のゲートも使って、作ってみてください。
- NAND だけでできたら、NAND ゲート を使う数を一番少なくするにはどうしてらよいか考えてみてください。

どのように考えていったらよいのでしょうか。ひとに説明できるように、考え方も整理できるとよいですね。

### 1.3.2 いくつかの点を通る多項式関数

- 平面上の二点  $(a, b)$  と  $(c, d)$  を通る直線は、いつでも一本ありますね。その方程式はどうなりますか。
- 三点  $(a, b)$  と  $(c, d)$  と  $(e, f)$  のときは、どうでしょうか。二次関数をつかうと、この三点を通るものがいつでも見つけられますか。

$$y = \alpha x^2 + \beta x + \gamma$$

として、 $\alpha, \beta, \gamma$  はいつでも見つかりますか。

- 二次関数が見つかる条件は何でしょうか。それは、ただ一つに決まるのでしょうか。
- 二点を通る直線は、1次関数に関係しており、三点を通る直線は、2次関数に関係しているとする、点の数を、4点、5点、6点とどんどん増やしていくとどうなると思いますか。
- すっきりした形で記述する方法はあるでしょうか。
- $y = \alpha x^3 + \beta x^2 + \gamma x + \delta$  は、 $\alpha \neq 0$  のとき、三次関数と呼びます。三点  $(a, b)$  と  $(c, d)$  と  $(e, f)$  を通る、三次関数をすべて求めることはできるでしょうか。
- どのように結果を整理（定理として記述）したら良いでしょうか。
- $y = \alpha x^3 + \beta x^2 + \gamma x + \delta$  は、 $\alpha \neq 0$  の左辺は、右辺の  $x$  の値によって決まりますから、 $x$  の関数とよび、 $f(x) = \alpha x^3 + \beta x^2 + \gamma x + \delta$  と書きます。 $f(x)$  は三次関数です。 $c$  を実数としたとき、

$$f(x) = \alpha x^3 + \beta x^2 + \gamma x + \delta = (\alpha' x^2 + \beta' x + \gamma')(x - c) + r$$

と書けることが知られてます（数学II）。 $\alpha' = \alpha$  と書けることはわかりますね。上のように書けている時、 $f(x)$  を  $x - c$  で割って、商が  $\alpha' x^2 + \beta' x + \gamma'$  余りが、 $r$  であるともいいます。

- 上の式で、 $r = f(c)$  となっていることに注意してください。特に、 $f(c) = 0$  のときは、

$$f(x) = (\alpha' x^2 + \beta' x + \gamma')(x - c)$$

と書くことができます。

- Desmos で遊んでみましょう。
  - 多項式関数 (<https://www.desmos.com/calculator/kgbvajslwl>) 係数の部分をスライドさせると、グラフの形が変わります。最初の表の値を変えて、それを通る関数を見つけることができますか。

- 。左上に三本線がありますが、それをクリックすると、例が出てきます。少しずつ変更してどんなことができるか、やってみましょう。

## 1.4 数学を復習しておきたい人のために

### 1.4.1 2進数の演算

1. 10進数の 0-15 を2進数で表すと
2. 10進数を2進数で表す方法
3. 2進数を10進数で表す方法
4. 2進数の足し算 例： $101_2 + 110_2$
5. 2進数の掛け算 例： $101_2 \times 110_2$
6. 2進数で1より小さい小数を表す方法

### 1.4.2 直線と二次関数

1.  $(0, -2)$  を通り、傾きが 3 の直線の方程式。
2.  $(0, b)$  を通り、傾きが  $a$  の直線の方程式。
3. 平面上の直線で、 $y = ax + b$  とは表せないもの。
4.  $ax + by + c = 0$  をみたす平面上の点  $(x, y)$  全体が直線となる条件。
5.  $(a, 0)$  と  $(0, b)$  を通る直線の方程式。
6.  $(a, b)$  と  $(c, d)$  を通る直線の方程式。
7.  $(0, 0)$ ,  $(1, -1)$ ,  $(3, 3)$  を通る二次関数。
8.  $(1, a)$ ,  $(2, b)$ ,  $(3, c)$  と通る二次関数 (か、1 次関数か...)

## 2 NAND ゲートの加算器

最初に、1.3.1 にあることから確認していききたいと思います。

### 2.1 クラスで考えてみたいこと

1.  $p, q, r$  がすべて 0 か 1 とすると、三組  $(p, q, r)$  は

$$(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)$$

全部で8通りあります。入力が、8種類の  $(p, q, r)$  のときに、出力が、それぞれ、たとえば、

$$(0, 1, 1, 0, 1, 0, 0, 1), \quad (0, 0, 0, 1, 0, 1, 1, 1)$$

となるような回路は作れますか。(( $0, 0, 0$ ) のとき 0, ( $0, 0, 1$ ) のとき 1, ( $0, 1, 0$ ) のとき、1 などです。) 加算器の問題とは、どのように関係しているのでしょうか。

2. 1で出力の8個組がなんであっても、それを出力する回路は作れるでしょうか。
3. 2の回路は、NAND だけでつくれるでしょうか。
4. 基本的なものをまず作って、それを組み合わせてできないだろうか。

## 3 いつくかの点を通る多項式関数

最初に 1.3.2 に書いてあることから確認していききたいと思います。

## 3.1 クラスで考えてみたいこと

1. たくさんの点を考える時、どのような記号がよいだろうか。
2. 基本的なものをまず作って、それを組み合わせることはできないだろうか。
3. すべての点で、0 という値をとるものは、どのようなになっているだろうか。
4. 一つの点では、1で他では、0 という値をとるものは、どのようなものが考えられるだろうか。
5. ただ一つしかないということは、どのようにしたら示すことができるでしょうか。

## 3.2 R and SageMath Code

下のプログラムと結果は、何をあらわしているのでしょうか。

```
# R Program
library(polynom)
x<-c(0,2,3,4)
y<-c(7,11,28,63)
poly.calc(x,y)
```

```
## 7 - 2*x + x^3
```

“SageMathCell” へのリンク (<https://sagecell.sagemath.org>)

次の二行を copy して、SageMathCell の窓に paste して、Evaluate くだみてさい。

```
R = PolynomialRing(QQ, 'x')
R.lagrange_polynomial([(0,7),(2,11),(3,28),(4,63)])
```

(0, 7), (2, 11), (3, 28), (4, 63) の部分を変更するとどうなるでしょうか。

## 4 まとめ

- 背景にある数学の考え方について、確認したいと思います。

## 5 おわりに

- 関連問題として、どのようなことが考えられるでしょうか。
- 疑問におもったことや、考えたいことを話し合えればと思います。

数学の問題を理解し、考え、英語を利用し、コンピュータ・ツールも利用することは、どのような、分野に進んでも、世界を広げる大きな力となると思います。みなさんと、一緒に学びを楽しむことを願っています。