百万石 2005年春号
MMA[1]

目 次

[1] みんなで マイコンで 遊ぼう会

はじめに

百万石は春と秋に刊行している MMA 部公式の雑誌です. この記事はそれぞれの MMA 部員が活動内容に関係なく思い付いたことを好きなように書いたものです.

MMA では現在, お天気プロジェクトという名称で気象観測機器を作成し, 100 万円程する市販の気象観測機器から置き換える試みを行なっています. それに関して去年は, デジタル温湿度計その観測システムを作成したり, 監視カメラの制御を行ない, 電通大の学園祭である調布祭で展示していました. また先月に風速計の市販品が届いたので, この仕組みについてこれから調べるつもりです.

MMA の部室の大部分は PC とその関連部品で占められていて, それぞれの PC は部員が基本や応用プログラミング, システムやネットワークのオペレーション[2]をするなど自由に使えるようになっています. さらに, 次世代インターネットの規約である IPv6 が部室のマシンまで接続されているので, 実力があれば使えます. また, 部室の入出にパスワード認証の鍵システムを使っていますが, Suica や Felica など非接触カードによる認証への切替えを現在試みています. これに関連して Felica による認証システムを調布祭で展示しました.

毎年夏には合宿と称した大花火会を行なっていて, MMA の OB の人達も観に来ています. OB の人には技術者が多くいますので参加してみれば最新の成果なども聞けると思います. 去年はある人が, 腕時計型の通信装置で指の動きを読み取って遠隔の機器を操作する, という実演をされました.

これを読んでいる人がもし MMA に興味があって部室へ来るなら, mma-active@mma.gr.jp へ連絡するか, 部室に設置されたカメラの映像が

```
http://www.mma.gr.jp/~enu/mmacam.html
```

で見れますから確認してから訪ねれば確実です.

部長

[2]IT スキル標準の専門分野より

第 I 部
Pascal プログラミングの基礎 第一回

(著者) J 科 egashira

1　プログラムを書いてみる

下を見るといきなりプログラムがありますが、決して難しいことを行なっているわけではありません。一般の参考書などでは初めに出入力の方法と基本的な算術演算子の扱い方を書くと思うのですが、それだと読者の方に面白みが無いと思いますし、こちらもページの都合があるので見れば分かるところはさらっと流してより実用的な部分に力をいれてプログラム作りの面白さを知って頂きたいと思います。例題 1 では、出入力及び算術演算子、算術関数の扱い方を知ってもらいます。

(例題 1)　三辺の長さが a,b,c であることが分かっている三角形の面積 area は、$s = \frac{a+b+c}{2}$ とすると、ヘロンの公式より

$$area = \sqrt{s(s-a)(s-b)(s-c)} \tag{1}$$

となります。この公式を用いて、三辺の長さを入力したら、三角形の面積を出力するプログラムを作りなさい。

〈 **解答例** 〉

```
 1:    program Heron(input,output);
 2:    var a, b, c, s, area : real;
 3:    begin
 4:      writeln(output,'three reals ?');
 5:      write(output,' a = ');
 6:      readln(input,a);
 7:      write(output,' b = ');
 8:      readln(input,b);
 9:      write(output,' c = ');
10:      readln(input,c);
11:      s:= (a+b+c)/2;
12:      area:= sqrt(s*(s-a)*(s-b)*(s-c));
13:      writeln(output,'三角形の面積 area は ',area:8:4,'です。')
14:    end.
```

```
sigma[68]%  ./a.out
three reals ?
 a = 2.3
 b = 3.4
 c = 4.5
三角形の面積 area は  3.8165 です。
```

図 1：例題 1 の実行結果

では、解説に入りましょう。まず、1 行目の program に続く “名前” はプログラム全体を表す任意の名前です。英数字ならば何でも結構ですがハイフンは何故かエラーになりますから、入れないようにしましょう。またその

直後にある括弧内には例題にあるように通常 input,output と書きます。また、その後には ‘;’ をつけるのを忘れないで下さい。

次に、2 行目の var はその後に主プログラムで使う変数を宣言します。変数の名前には英数字を一文字以上にして割り当てることが出来ますが、数字を変数の先頭または単独で使うことは出来ません。のように値を入れます。そして、‘:’ で区切った後にその変数の型を書きます。この場合 real というのは宣言した変数にどのような値が入れられるかを指定しています。real の他にも基本的なデータ型として以下に挙げる 3 つがあります。

標準のデータ型	型の名前	型の持ち得る値の例
integer	整数型	...,-2,-1,0,1,2...
real	実数型	0.5,12.3,-4.2,3.62e-03
Boolean	論理型	false,true(2 つだけ)
char	文字型	a,V,%

表 1: データ型の種類

まず、プログラムの主体に入る前に注意点をいくつか。3 行目と 10 行目にそれぞれ begin と end と書かれています。プログラムの主体は必ずこの 2 つの間に無ければいけません。また、end. 直前の最後の行以外は文が一つ終わる度に ‘;’ で区切ります。これを抜かすとエラーになるので注意して下さい。さて、やっとプログラムの主体に入ります。まずは、小ネタで行きますが、4 行目と 5 行目とでは微妙に違うのですが、どこか分かりますか?そう、write の後に ln があるかないかです。この ln を付けとその作業が終わったときに改行するようになります。以下 10 行目まで同じことを行なっています。また、文を出力したい場合には’—文—’ と書きます。但し、変数に値が入っているときにいれる必要はなく’—文—’, 変数,’—文—’ と書き文と変数をカンマで区切ります。

11 行目からは算術演算子がでて来ているので、説明するよりも一覧表 (表 2) 作ったのでそちらを参考にして下さい。

12 行目に注意点があるとすれば、‘:=’ と ‘sqrt’ についてでしょう。変数に値を代入するときにはただ ‘=’ でつなげばよいというわけではなく、左に変数を、右に値を持つものを持って来て ‘:=’ でつながなければいけません。

次に、sqrt というのは算術関数と呼ばれるものの一つで、sqrt(x) と書いて $\sqrt{x}$ という演算結果がでます。sqrt 以外にも表 3 にあるようなものがあります。

+	整数型、実数型の足し算を行なう
-	整数型、実数型の引き算を行なう
*	整数型、実数型の掛け算を行なう
/	実数型の割算を行なう
mod	割算の余りを求める

表 2: 算術演算子一覧

13 行目が終われば Pascal プログラミングで必須と思われることは大体終わりです。ここで覚えて欲しいことは一つで area の後に ‘:8:4’ とあります。これの用途についてです。その部分が無い場合の実行結果を挙げるので比較して下さい。

見れば分かるように、変数 area を出力すると小数点以下を延々と書いていき最後に e+00 とわけの分からないものまでついてきます。一般的には ‘:w:f’ と書いて w 文字分の全体幅の中に小数点以下を f 文字分とって右揃えで出力するといった意味にるのでそれを除いたためです。そして、e+00 というのは、10 の 0 乗を意味し、e+0n と書かれていた場合、10 の n 乗を意味します。

関数名または手続き名	multicolumn1—c—機能	関数の場合は型
abs(x)	引数の絶対値を返す	integer か real
arctan(x)	引数の逆正接値を返す	real
cos(x)	引数の余弦値を返す	real
exp(x)	引数の指数関数値を返す	real
frac(x)	引数の小数部分を返す	real
int(x)	引数の整数部分を返す	real
ln(x)	引数の自然対数値を返す	real
sin(x)	引数の正弦値を返す	real
sqr(x)	引数の平方根を返す	integer か real
sqrt(x)	引数の平方根を返す	real

表 3: 算術関数一覧

```
sigma[3]%  ./a.out
three reals ?
 a = 2.3
 b = 3.4
 c = 4.5
三角形の面積 area は 3.816490534509418e+00 です。
```

表 4: 実行結果 (2)

2 制御文

プログラムを書く上で何が何でも習得したいのがこの制御文でしょう。この制御文が扱えるようになるとプログラムの幅が飛躍的に広がりまた、自分の作りたいプログラムがある程度書けるようになります。そうすると、プログラミングが楽しくなって来ますから、この機会に覚えてしまって下さい。

2.1 if 文

では、いきなりですが例題から入りましょう。

(例題 2) 2 つの実数を読み込み、それらの最大値と最小値を出力するプログラムを作りなさい。但し、二つが等しいときにはそれを知らせるようなものにすること。

〈 **解答例** 〉

```
1:  program minMax(input,output);
2:  var a, b, min, max : integer;
3:  begin
4:    writeln(output,' 二つの整数を入力して下さい');
5:    write(output,'a = '); readln(input,a);
6:    write(output,'b = '); readln(input,b);
7:    if a < b then begin min := a; max := b end
8:    else if a > b then begin min := b; max := a end
```

```
 9:     else writeln(output,' 最大値と最小値はなく二つの整数は等しい');
10:     if (a < b) or (a > b) then
11:        writeln(output,'min = ',min : 1,' max = ',max : 1)
12:   end.
```

```
sigma[80]% ./a.out
二つの整数を入力して下さい
a = 12
b = 10
min = 10 max = 12
sigma[81]% ./a.out
二つの整数を入力して下さい
a = 1
b = 1
最大値と最小値はなく二つの整数は等しい
```

表 5: if 文 実行結果

制御文の筆頭とも言える if 文です。今回の Pascal プログラミングでは著者の時間とページの都合で if 文と repeat 文しかお教えできませんが、この二つを覚えておくだけでもプログラミングの自由度をかなり高くできると思います。

では、解説に入りましょう。6 行目までは問題ないと思うので、7 行目の if 文の説明から入りたいと思います。まず、if 文の書き方から入ります。if 文には代表的なパターンが 3 つありそれらは以下の通りです。

(a) if 論理式 then 文

(b) if 論理式 then 文 1
    else 文 2

(c) if 論理式 1 then 文 1
    else if 論理式 2 then 文 2
    :
    :
    else if 論理式 (n-1) 文 (n-1)
    else 文 n

if 文の説明にはいる前にまたいくつか説明を要するものがあります。その一つ目が、論理式についてです。論理式と言うのは以下の 3 つによって定義されます。

(1) 論理型の定数、及び変数は論理式

(2) a と b を整数型、実数型、文字型のいずれかの型の式 (ただし、a、b はいず れも同じ型、または一方が整数型で他方が実数型) としたとき、$a = b$, $a <> b$ $a < b$ $a <= b$ $a > b$ $a >= b$ はいずれも論理式

(3) p と q を論理式としたとき、not(p)、(p)and(q)、(p)or(q) も論理式

ここで、=, <>, <, <=, >, >= を関係演算子と言います。それらの定義を表に示します。数学で出てくるものがほとんどなので覚える程ではないかも知れません。また、not、and、or を論理演算子と呼びます。これらはそのままの意味なので表は割愛させて頂きます。

関係演算子	定義
a = b	a と b が等しければ true、そうでなければ false
a ⟨ ⟩ b	a と b が等しくなければ true、そうでなければ false
a ⟨ b	a が b より小さければ true、そうでなければ false
a ⟨= b	a が b 以下であれば true、そうでなければ false
a ⟩ b	a が b より大きければ true、そうでなければ false
a ⟩= b	a が b 以上であれば true、そうでなければ false

表 6: 関係演算子の定義

以上を踏まえると、

- (a) は、論理式が真のときに文を実行し、偽のときには何も行なわない
- (b) は、論理式が真のときに文 1、を偽のときには文 2 を実行する
- (c) は、論理式 1、論理式 2、.... を順々に真になるまで調べていき、初めて真になる論理式が論理式 i ならば、文 i を実行しこの if 文全体を終え全ての論理式が偽のときは、文 n を実行します

となります。

ちょっとだけ注意して頂きたいのは if 文というのは (a) の場合を除けば、else 文が終わって初めて if 文全体が終わったことになるので、';' をつける場合は、else 文が終わった後にしないといけません。

さて、例題の解説に戻りたいと思います。といっても、以上のことが分かっていれば例題で分からない点はもうないと思います。ただ、2 つだけ注意して欲しいところがあります。1 つは些細なことなのですが、and や or で変数同士やその式を結ぶときには (変数の式)and(変数の) と言うように括弧でくり、not の場合も not(変数の式) として下さい。2 つ目は、if 文も文の一つなので、then や else の後にまた if 文が続くこともできます。実際 (c) のパターンは、else の後に if 文が続いている例です。

特に注意が必要なのは次の場合です。

if 論理式 1 then if 論理式 2 then 文 1 else 文 2

は、

if 論理式 1 then begin if 論理式 2 then 文 1 else 文 2 end

と解釈され、else は未対応の直前の if(または then) に対応すると定められています。そのため、もし論理式 1 が偽のときに文 2 を実行したいのなら、

if 論理式 1 then begin if 論理式 2 then 文 1 end else 文 2

と書かなくてはいけません。

2.2 repeat 文

続いては、repeat 文に入ります。

(例題 3) 正整数 S を読み込み、初めて $\sum_{k=1}^{n} 1/k \geq$ S となる n を求めるプログラムを作りなさい。

〈 **解答例** 〉

```
 1:  program sum(input,output);
 2:  var b, d : integer;
      a, c, e : real;
 3:  begin
 4:     writeln(output,' 正の整数を入力して下さい');
 5:     repeat
 6:        write(output,'S = '); readln(input,a);
 7:        if (a <> (trunc(abs(a))))
 8:           then writeln(output,'S には正の整数を入力して下さい');
 9:     until a = (trunc(abs(a)));
10:     b := 1;
11:     c := 0;
12:     d := 0;
13:     e := 0;
14:     repeat
15:        d := (1 + d);
16:        e := (1/d);
17:        c := (c + e);
18:     until c >= a;
19:     writeln(output,'1+1/2+1/3+...+1/n が S 以下になるのは n = ',d,' の時
です。')
20:  end.
```

7 行目に trunc という見慣れない演算子らしきものがあると思います。これは、変換関数の一つで、括弧内の値の整数部分を返すものです。他には repeat 文以外で特筆しなければいけないものはないので repeat 文の構造の説明をしたいと思います。

repeat 文 1; 文 2; ..... ; 文 n until 論理式

repeat 文は、文 1〜文 n を、論理式の値が true になるまで、繰り返し実行します。各繰り返しの最後に論理式の値が計算され、true であれば、repeat 文全体の実行が終了し、false であれば、繰り返しの先頭 (文 1) に戻ります。また、until の後には、

until (条件文 1) (and または or) (条件文 2) ...

というように複数の条件文を and や or でつなげることができます。

以上で今回の 'Pascal プログラミングの基礎' は終わりです。ここに記述されているものだけではまだ十分なプログラムは書けないでしょうが、その辺りはご容赦下さい。

第II部
C科1年における歩み方と100円ラジオ探求

(著者) moechar
moechar@mma.club.uec.ac.jp

概要
新入部員の皆さん入学おめでとう。と真人間ムードを醸し出しているが、下記の内容は私というつくりの悪いフィルターを通して得られた**不鮮明な記憶**に基づいて MMA らしく[3]記したものである。動作確認済みの対応環境[4]が C 科第二クラスの物理学年 1 年のみであることを断っておく。後学期の微分積分学第二, 線形代数学第二で目にする代表的な問題について、筆者が正しく理解しているか確認の意味をこめて説明した。また、100 円ラジオについても触れた。

3 前学期

3.1 関門

前学期には慣れも含めて学生として通らなければならない関門がいくつかある。

- 環境が Sun Solaris
  総合情報処理センターも C 科計算機室 IED も Solaris を積んでいる。窓使いな人やマクドな人にはカルチャーショックかもしれない。必然的に Sun Pascal を使う事になる。使用が推奨されているブラウザが Netscape であることに抵抗を示す人もいる。

- 基礎科学実験 A
  別名物理実験。**レポート**の書き方を徹底的に指南してくれる。**レポート地獄の日々**を送ることになる。大幅に時間を取られる。徹夜で書き終わらないテーマも存在する。**過去レポの写経**はバレ易い (教官はこの類は何百例と経験している) のでお薦めできない。2 人 1 組で行う実験が殆どである為パートナーとの相性や手際の良し悪しで実験時間やレポートの評価が変わる。

- レポート漬けの毎日
  レポートは物理実験だけとは限らない。溜め込まないように。本稿のように LaTeX で書いてみるのも一考。

  主張 3.1.1 レポート出したら負けかなって思ってる。

- 数学や理科だらけの日々
  本学の学生たる宿命。計算ミスで苦しんだり、楽しかったり、お腹いっぱいだったり。

  主張 3.1.2 やる気出したら負けかなって思ってる。

- 試験
  科目によって中間試験も実施されるが基本的に 7 月最終週に実施される。試験期間より前に行われる科目もある。[5]あまりにも強行軍なため、一日に 5 科目ある人もいる。殆どが記述式。低学年次における**過去問**は有効でない事が多い。

[3] 我々は一断固としてーたたかうー
[4] この環境で実行していかなる不具合が発生したとしても筆者は対応しない。自己責任で。
[5] 人文科目や外国語の科目等に見られる。数学演習第一も試験期間前に実施される。

3.2 MMA 的には

4 月の終わり位に新人歓迎飲み会があったり。hoge ちゃんねるへの誘いとか。

4 夏期休暇

追試がなければ丸二ヶ月の長期休暇となる。MMA では引きこもり部員を 4 次元ベクトル空間に連れ戻すべく毎年合宿を実施している。間が空き過ぎるため休み明けに大学に行かなくなる人もいる。盆休み辺りに**夏の聖戦**[6]が予定されている。

5 後学期

後学期はいくらか楽になるだろうと思いきや、**そうはい神崎**とばかりに必修科目, 選択必修科目の山が積まれている。一年次後学期が電通大生生活の中で最も忙しいと言われている。

5.1 鬼門の離散数学第一と同演習

講義と演習でセット。2 時限拘束される。これを取らないと論理的に 3 年になれない。毎年二桁人数の再履修者を量産している。前学期に現代数学入門 A を履修しているとわかり易くなるといわれている。

主張 5.1.1 単位とノートを取ったら負けかなって思ってる。

5.2 楽な単位

履修可能性のある科目は増えるが、前学期よりも薄くなっている感がある。微分積分学や線形代数学で顕著に見受けられる。少し垣間見るつもりで演習してみよう。

演習 5.2.1 次の積分の値を求めよ。空間の極座標を用いて変数変換してもよい。

$$\iiint_D \frac{dxdydz}{\sqrt{1-x^2-y^2-z^2}} \quad D : x^2+y^2+z^2 < 1$$

[**解**] $x = r\sin\theta\cos\varphi,\ y = r\sin\theta\sin\varphi,\ z = r\cos\theta\ (0 \leq r,\ 0 \leq \theta \leq \pi,\ 0 \leq \varphi < 2\pi)$ とおく。
ヤコビアンを求めると、

$$\left|\frac{\partial(x,y,z)}{\partial(r,\theta,\varphi)}\right| = \begin{vmatrix} x_r & x_\theta & x_\varphi \\ y_r & y_\theta & y_\varphi \\ z_r & z_\theta & z_\varphi \end{vmatrix} = \begin{vmatrix} \sin\theta\cos\varphi & r\cos\theta\cos\varphi & -r\sin\theta\sin\varphi \\ \sin\theta\sin\varphi & r\cos\theta\sin\varphi & r\sin\theta\cos\varphi \\ \cos\theta & -r\sin\theta & 0 \end{vmatrix} = r^2\sin\theta(\geq 0)^7$$

となる。

$$dxdydz = r^2\sin\theta dr d\theta d\varphi$$

[6] 全国のキモヲタや腐女子、レイヤー、カメラ小僧の方々が種々の冊子目当てに某臨海展示場に集結する行事。
[7] 行列式をサルス展開した。

ゆえに、

$$\begin{aligned}(\text{与式}) &= \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \int_0^1 \frac{r^2 \sin\theta}{\sqrt{1-r^2}} dr \\ &= \Big[\varphi\Big]_0^{2\pi} \cdot \Big[-\cos\theta\Big]_0^{\pi} \cdot \int_0^{\frac{\pi}{2}} \frac{\sin^2\delta}{\sqrt{1-\sin^2\delta}} \cos\delta d\delta \\ &= 2\pi \cdot 2 \cdot \int_0^{\frac{\pi}{2}} \frac{1-\cos 2\delta}{2} d\delta \\ &= 2\pi\Big[\delta - \frac{\sin 2\delta}{2}\Big]_0^{\frac{\pi}{2}} \\ &= \pi^2\end{aligned}$$

厳密にはこの積分は広義積分であるので文字で置き lim で飛ばさなければならないが数学上この様に記しても問題ない。x_r とは x を r について**偏微分**[8]することを意味している。つまり、

$$\frac{\partial x}{\partial r} = \frac{\partial}{\partial r} r\sin\theta\cos\varphi = \sin\theta\cos\varphi \cdot \frac{d}{dr} r = \sin\theta\cos\varphi$$

演習 5.2.2 次の *3* 次実正方行列の対角化可能性を確かめた上で**対角化**[9]せよ。

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 2 \end{pmatrix}$$

[**解**] まず、A の固有多項式から固有ベクトルを求める。t を A の固有値として、

$$\begin{aligned} g_A(t) &= |tE - A| = \begin{vmatrix} t-1 & -2 & 0 \\ 0 & t+1 & 0 \\ -1 & 0 & t-2 \end{vmatrix} = (t-1)(t+1)(t-2) = 0 \\ &\Leftrightarrow \quad t = \pm 1, 2 \end{aligned}$$

よって各固有値について $A\boldsymbol{x} = t\boldsymbol{x}$、即ち $(tE-A)\boldsymbol{x} = \boldsymbol{0}$ の解 $\boldsymbol{x}(\forall \boldsymbol{x} \in \mathbb{R}^3)$ を求める。

(1) t=1 のとき

$$E - A = \begin{pmatrix} 0 & -2 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \end{pmatrix} \xrightarrow{\text{簡約化}} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \therefore \boldsymbol{x} = s \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} (\forall s \in \mathbb{R})$$

よって固有空間は $W(1;A) = <^t (1,0,1)>$[10]

(2) t=-1 のとき

$$-E - A = \begin{pmatrix} -2 & -2 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & -3 \end{pmatrix} \xrightarrow{\text{簡約化}} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & -3 \\ 0 & 0 & 0 \end{pmatrix} \quad \therefore \boldsymbol{x} = s \begin{pmatrix} 3 \\ -3 \\ 1 \end{pmatrix} (\forall s \in \mathbb{R})$$

よって固有空間は $W(1;A) = <^t (3,-3,1)>$

[8] この場合、r 以外の文字を定数とみなして x を微分するという意味。

[9] 正方成分以外の値を 0 にする事。

[10] 実際は 3 行 1 列の列ベクトルだが左上に ^{t} と振る事で行と列の数を入れ替える事が出来る。これを**転置**といい、転置して得た行列を**転置行列**という。

(3) t=2 のとき

$$2E-A=\begin{pmatrix}1&-2&0\\0&3&0\\-1&0&0\end{pmatrix}\overset{\text{簡約化}}{\longrightarrow}\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}\quad\therefore\boldsymbol{x}=s\begin{pmatrix}0\\0\\1\end{pmatrix}(\forall s\in\mathbb{R})$$

よって固有空間は $W(1;A)=<^t(0,0,1)>$

よって $\dim(W(-1;A))+\dim(W(1;A))+\dim(W(2;A))=3=\mathrm{rank}(A)$ が成り立つので対角化可能である。上記の過程で得た 3 つの固定列ベクトルから正則行列 $P=\begin{pmatrix}3&1&0\\-3&0&0\\1&-1&1\end{pmatrix}$ を定める。また、P の逆行列 $P^{-1}=\begin{pmatrix}0&-\frac{1}{3}&0\\1&1&0\\1&\frac{2}{3}&1\end{pmatrix}$ である。求める対角化行列を B とすると、

$$\begin{aligned}B&=P^{-1}AP\\&=\begin{pmatrix}0&-\frac{1}{3}&0\\1&1&0\\1&\frac{2}{3}&1\end{pmatrix}\begin{pmatrix}1&2&0\\0&-1&0\\1&0&2\end{pmatrix}\begin{pmatrix}3&1&0\\-3&0&0\\]1&-1&1\end{pmatrix}\\&=\begin{pmatrix}-1&0&0\\0&1&0\\0&0&2\end{pmatrix}\end{aligned}$$

B のように正方成分のみに値を持つ行列を**対角行列**という。ちなみに、この対角化の流れは受験数学では A^n を求めさせる問題で用いられている。

主張 5.2.1 正解したら負けかなって思ってる。

5.3 趣味

言うまでもないが、冬期休業期間中には**冬の聖戦**[11]も予定されている。

6 キーワード

上記で 1 年次のサマリを述べたつもりだが、重要な**キーワード**が残っていたので述べておく。

- MMA
  Microcomputer **M**aking **A**ssociation の略。前年度は東大院に約 1 名[12]の部員が進学した。
  http://www.mma.club.uec.ac.jp/

- 人間不信
  アカの他人の人間性など信じてはいけない。馬鹿を見るだけである。本学の学生のうち 90% の者は非常識かつ厚顔無恥である。つまり人間性が欠落しているのである。[13]

- GPA：評価点平均。2.5 あたりから上げ辛くなる。算出式は学修要覧を参照せよ。

[11]ちなみに**春の聖戦**もある。
[12]最近頓に思うところであるが、**人の数は明らかに整数である**ので**四捨五入する意味が無い**のではないか。
[13]私も例に漏れない。

- 留年：卒業予定年度 > 入学年度+3 となること。発生の要素は再履修。
- 再履修：不可評価を頂くと来年度以降に履修して可以上の評価を取ることが義務付けられる。
- 教授：名誉教授 > 教授 > 助教授 > 講師 > 助手の序列に位置する。博士/修士/学士は学位。
- ナベジュン (本名 渡辺純子)
  昨年度前学期まで本学 H 科助教授。後学期から京大院に異動。人文科目の経済学 A, B を担当していた。一時期生協の店員に転職したのでないかという噂も流れた。
- 花火：MMA では夏期休暇中を利用して本物の花火を上げる事がある。
- リテラシー：UNIX 環境での基本操作を覚えさせることが目的。
- な〜るほど!の〜
  本学 F 科の伊東敏雄教授著作の基礎物理学シリーズ。LaTeX で記されているのが特徴。教科書として指定されることがある。
- 数学教室
  基礎数学教育に携わる教官の集団。基本的に TeX 使いだがミスプリが多い。
  http://matha.e-one.uec.ac.jp/
- 自己再実験コマンド
  文字通り自主的に再実験に臨むこと。他学科の班に紛れると成功し易いといわれている。
- 再レポ
  基礎科学実験で、レポート内容に不備があると通達されることがある。減点対象ではない。
- TSK：図書館の略 (非公式)。To Sho Kan。お分かりだろうか。CC 棟 1,2,3F にある。
- ハルモニア
  大学会館 3F にある本学が誇るレストラン。多くの教官が贔屓にしている。無論、学生も使える。食堂の 3,4 割増の料金だが心の安静を得られることがある。肉野菜炒めが同店名で商品化されている。

7 推奨環境

用意しておくと役に立つ道具になりうる物を挙げる。

- 何らかの UNIX 環境。Cygwin でも Putty[14]でも代用可。
- CC のアカウント
- TeX 環境
- 妥協のすぴりっと
- w3m パッケージ
- 濃淡に偏在する知識
- 虚言癖

[14] この場合はホストから許可されて ssh で login できるアカウントが必要になる。大抵は CC と IED で事足りる。

8　勉強の進め

余計なお世話であれば薄い内容は本章で終わりなので読み飛ばしてもらっても構わない。良い評点を取っておきたいという勉強好きなアナタに参考書を紹介する。版によって改訂されている事があるので要確認。特に全部こなす必要は無い。

(1) 単位が取れる[15]微積ノート 馬場敬之著 講談社
教科書の最低限の範囲は解説しているがこれだけで単位を取れるとは思えない。

(2) キャンパスゼミ[16] 微分積分 馬場敬之著 マセマ
上記と同じ著者の本。内容はほぼ同じ。馬場本来の味を愉しむ事が出来る。

(3) 単位が取れる線形代数ノート 齋藤寛靖著 講談社
河合塾と駿台予備学校の数学講師の齋藤氏が書いた本。微積ノートより網羅度が高い。

(4) キャンパスゼミ 線形代数 馬場敬之著 マセマ
上記の線形代数 Version。教科書に比べて教科書の内容がかなり落ちている。

(5) 演習 線形代数 寺田文行著 サイエンス社
昨年、伊東裕也助教授は課題レポートの問題にこの本から数題チョイスした。

(6) 穴埋め式 線形代数 らくらくワークブック 藤田岳彦, 石井昌宏共著 講談社
石井氏は本学T科の助手。懇切丁寧な例題の解答が参考になる。

(7) なっとくする微分方程式 小寺平治著 講談社
解析学の微分方程式の分野で重宝する。常微分方程式のみついて言及している。丁寧な説明と適量な演習問題を評価した。

(8) 物理学 小出昭一郎著 裳華房
力学第一で指定された教科書は本書の分冊版である。ハードカバーで扱い辛いがこれ一冊で一年次に習う基礎物理学の内容をカバーする。座右の書としても使える。

(9) 物理学 小出昭一郎著 東京教学社
上記の著書が A5 版なのに対して B5 版。内容はほぼ同等。

(10) 詳解物理学 原康夫著 東京教学社
B5 版なので鞄の収まりがいい。分量も適当。

(11) 単位が取れる力学ノート 橋元淳一郎著 講談社
数式のみでなく文章で本質的な内容を説明している。力学第一の内容しか扱っていない。

9　100円ラジオ

数年前某 100 円ショップで購入した Amplitude Modulus ラジオは今、店頭に置いてないという。電子工作ファンを風靡したこのラジオを取り上げる。

[15] 人気の予備校講師が執筆している。基礎科学の科目では担当教官によっては細かい内容も取り扱うためこのシリーズでは補えない事が多い。筆者としては安易にこのような色物に奔るのはお薦めできない。

[16] これも色物に部類する。無意味にカタカナが混じっていたら色物も同然。自分の頭と手で数式を組み立てて正解を導出することに意味があるのではないだろうか。

9.1 魅力

100 円ラジオの魅力にはやはり安いということ、さらには高い拡張性を有するということを挙げる事が出来る。

9.1.1 cheap 感

秋葉原で同様のパーツを揃えると 1000 円以上掛かる。スーパーヘテロダイン方式の 3 石トランジスターラジオと垂涎ものだが、中国製。そう。めいどいんちゃいなである。基盤を覗くとテキトーなハンダ付け痕[17]から哀愁漂う安っぽさが滲み出てくるのが否応無に分かる。**だがそれがいい**。

9.1.2 拡張性

ここでいう拡張性とは規格化された PC パーツのそれとは全く異なる。ケースと基盤が某キットのごとく幅広い改造のバリエーションを生む。ワイヤレスマイクや 50MHzAM、航空無線受信機などの改造例が存在する。260pF(推定) のポリバリコンもまた人気を呼んでいる。

9.2 可聴放送局

私の何年も耳鼻科にかかってない不鮮明な聴覚ではとりあえず周波数順に JOAK, JOAB, AFN, JOKR, JOQR[18], KBS, JOLF, JOUF, JOSF, JORF, JORL を捕らえる事が出来た。それなりの基準は満たしている。

9.3 品質向上

しかしこのままでは聴ける放送局が少ないうえに音声が不鮮明であるので拡張する。

9.3.1 ループアンテナについて

ループアンテナとはループ状に導線がぐるぐると輪のように巻かれたアンテナのことである。ループアンテナは基本的に通常のラジオに内蔵されているバーアンテナを大きくした構造をとる。バーアンテナはコイル (磁性体の周りに密に導線を巻いたもの) で，送信されてくる電波の変化に応じて電流が流れる (ファラデーの電磁誘導の法則による)。電波 (電磁波) の変化によって磁力線がコイルの内空を最も多く通過する向きで感度が最大になる。基本的にコイルの断面積が大きいほど刻々と変化する磁場に対する感度・効率がよくなるので内蔵の小さなバーアンテナよりも表面積の大きいループアンテナの方が微少電波を拾いやすくなると言われている。これに電気容量可変コンデンサー (バリコン) を組み合わせれば共振回路が完成する。バリコンを使ってコンデンサーの電気容量を変化させ放送局が送信している固有の周波数と一致させれば共振現象により，さらに振動が大きくなる (特定の周波数の電波を強めて拾うことが可能となる)。

9.3.2 ループアンテナの製作

市販されてはいるループアンテナ[19]であるが、あえて自作してみた。用意したものは

- $1[\mathrm{m}^2]$ のダンボール[20]

[17] もしかするとハンダのスズ比やハンダこての消費電力もテキトーなのかもしれない。
[18] 月曜の午前 2 時と午前 5 時に史上最強空前絶後の萌えソングがかかる。文化放送の歌とも言われている。
[19] コンポ等を買うと付いてくることがある。ラジオのみでなく、IC タグにも採用されている。
[20] そういえば某女性タレントの件で DANBALL Z というパロディ漫画があった気がする。

- 大量の導線。(ビニール被覆)
- 100 円ラジオのポリバリコン
- バリコンを搭載した 2 機目の 100 円ラジオ
- 50[cm] の台
- 飽きない精神

製作過程を以下に示す。

(1) バリコンの両端を一本の導線でハンダ付けし、中心部のビニール被覆を除去。

(2) ダンボールの周りに重ならないように 10 回巻き、両端をバリコンの真中の端子と (1) で被覆を取った部分にハンダ付け。

(3) ダンボールを開口し、その中心線に来るように台を置き、さらにラジオを置いた。

以上で完成である。試しに JOQR(1134[kHz]) にチューニングした結果音質が向上した。バリコンを回すとさらに感度が良くなった。そこで日ごろ訊き辛い[21]と悩んでいた JORF(1422[kHz]), JOUF(1314[kHz]), JOSF(1332[kHz]) にチューニングしてみるとこれまた音質が良い。

9.3.3 注意

ループアンテナの効果はあくまで放送電波の増幅であるため受信地に電波存在しなければ受信する事が出来ない。ただ、夜間になると電波を吸収する D 層の働きが弱くなり、電波を反射する E 層の働きが活発化するため遠隔地から発せられた電波が届く事がある。それをループアンテナで増幅するという試みである。従って、ループアンテナでの作業は夜間の方が好都合という事になる。

また屋内に設置する場合、ループアンテナ自身の指向性によって室内の電化製品からでる電磁波の影響も増幅されてしまい雑音が入りやすくなる。なるべくそれらから離れたところに置き、電源を切っておく事が望ましい。

導線の巻き数も電波の周波数と密接な関係がある。本来ならば受信する周波数に応じて巻き数を調節することが望ましいが、受信電波の種類が中波であるのでバリコンの調節で対応できる。

10 参考文献

(1) AM ラジオ遠距離受信の部屋
http://www.oyakudachi.net/amradio/amradio_top.htm

(2) ラジオの遠距離受信のテクニック
http://www.aka.ne.jp/~deguchi/hobby/radio/

(3) AM ラジオ用のループアンテナを自作しよう
http://www.geocities.jp/wepon_bafu/loop_antenna.html

(4) ループアンテナで遠距離受信
http://tvdx.hp.infoseek.co.jp/ru-pu.html

[21] 遠隔地発で低送信出力がである事が原因。

第 III 部
無限解析の基礎

概要
無限小について解説するつもりで書いたが, 読者は無限について馴染みがないし, そもそも高校 (中学後期) で無限大は数ではないと覚えさせられてきているから, 無限小を解説するまでの前書きが非常に長くなってしまった. コンピュータが数学者の内部で発生してきた事についても触れていたら, 無限小について解説する余裕がなくなってしまった.

11　世界物理年

今年は現代物理学が始まって 100 周年ということで, その記念行事が世界各地で開かれているらしい. 1 月には電通大でも 1997 年度ノーベル物理学賞受賞者 W. Phillips 氏の一般向け講演が開かれて, 実際に私も 300 人以上詰まった満員の講義室でその講演を聞いた. 彼の講演を外国人研究者が何人も聴講していて何度も質問していたのを思い出す. 彼はレーザ光で原子を冷したり捕まえたりすることに成功しノーベル物理学賞を贈られたそうだ. 100nK まで冷したとか, 現在では 50pK まで冷せてボーズ-アインシュタイン凝縮が起った, また電通大でもボーズ-アインシュタイン凝縮に成功したなどと言っていた.

100 年前に現代物理学を始めたのはもちろんアインシュタインで, その 1905 年の 3 月に光量子の論文を, 4 月と 5 月にはそれぞれブラウン運動の論文を, 6 月には特殊相対論の論文を, 9 月には質量とエネルギの関係について, いわゆる $E = mc^2$ の論文を発表したのでその年は「奇跡の年」と呼ばれているそうです. 今年 1 月のその講演によるとアインシュタインは光量子の論文にノーベル物理学賞を贈られたが, なんでもこの論文は「奇跡の年」の 5 つの論文の内で最もつまらないものだと思われているそうだ.

この 5 つの論文の内で相対論は逆に中高生にも知られている程に有名で, 時間と空間の関係について興味を持つ人なら誰でも知りたいことだと思う. ある本 [1] によれば, 特殊相対論は 2 つの仮定の上に築かれているという. 1 つは 力学におけるガリレオ=ガリレイの相対性原理を電磁気学に拡張したもの, もう 1 つは光速の不変性だという. ここでガリレオ=ガリレイの相対性原理とは

> 互いに相手に対して慣性運動している 2 人の観測者は, 自然法則を全く同じ形で定式化する. 特に, どのような観測者も, 絶対静止と絶対運動とを自然法則を用いて見分けることはできない. したがって絶対運動は存在せず, 相対運動のみ存在する.

ということで, 特殊相対論は観測者が慣性座標系に居るという意味で特殊な状況を想定しているという.

ガリレイの相対性原理について眺めてみよう. 2 人の観測者 A と B があってお互いに相対速度 v ですれ違うとする. 観測者 A は x-空間軸と s-時間軸, 観測者 B は y-空間軸と t-時間軸を使うとする. ただし, x-空間軸と y-空間軸の向きは同じ, また観測者 A,B がすれ違った時を s-時間軸と t-時間軸の原点とする. こうして観測者 A から見た座標 (s, x) と観測者 B から見た座標 (t, y) ができた. 物体 C があって観測者 A から見た物体 C の座標が (s,x) であるとすれば同時刻に観測者 B から見た物体 C の座標 (t,y) は, 観測者間の相対速度が v だから,

$$\begin{pmatrix} t \\ y \end{pmatrix} = \begin{pmatrix} s \\ x + vs \end{pmatrix} \tag{2}$$

となる. つまり,

$$G_v \ : \ \begin{pmatrix} s \\ x \end{pmatrix} \longrightarrow \begin{pmatrix} t \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ v & 1 \end{pmatrix} \begin{pmatrix} s \\ x \end{pmatrix} \tag{3}$$

である. この線型写像 G_v がガリレイ変換と呼ばれるものです. この変換が正しいとすると観測者 A から見て光速 C で動く物体は観測者 B から見ると速度 $C+v$ となったり, したがって電磁気における Maxwell の法則が観測者によって違うなどという事が起ってしまいます. これを修正した線形写像 L_v をローレンツ変換と呼び, 特殊相対論では上に挙げた 2 つの仮定から導かれます.

今見たように特殊相対論は線型代数によって作られていますが, しかし一般相対論はそうでなくてリーマン幾何学 [5] という微分幾何学の 1 つによって作られています. リーマン幾何学は微分幾何学の 1 つというより, 微分幾何学や多様体論がリーマン幾何学によって始まったという感じのものです. リーマン幾何学で何をやりたいのかというと, 世界地図を眺めれば分かるように赤道以外の水平線は直線ではありません. というのは, それらの水平線を辿ると最短距離にならないからで, 直線であれば最短距離であるはずだから. 逆に全ての垂線は直線です. このように歪みがある空間は一般に非ユークリッド空間よばれ, リーマン幾何学はこのような空間を計量という定規で統一的に扱ったものです. この計量を考えるときにベクトルや行列を拡張したテンソルと呼ばれるものを使うため, テンソル解析というものが派生しています.

12 位相とは

ある集合, 例えば平面 R^2 を考えたときに 2 点 a, b の距離を

$$d(a,b) = \sqrt{(a_1-b_1)^2+(a_2-b_2)^2} \tag{4}$$

と定めると距離 $d(a,b)$ はユークリッド距離と呼ばれて, 平面 R^2 とユークリッド距離 $d(a,b)$ を組み合わせたもの (R^2,d) をユークリッド空間と呼び, 再び記号 R^2 で表わすことは知られていると思います. さらに, 色々な集合も扱いたい, 近いかどうかだけでも分かるなら距離まで知らなくても十分だ, こういうものを一般的に扱うのが位相空間 [3] です. このように集合に近さという概念が定まっていれば収束や連続という概念も定めることができるという意味で都合がいいのです. この近さを判定するものを位相といいます. 距離が分かれば近さも分かるという意味でユークリッド空間も 1 種の位相空間です. 位相はトポロジー [4] とも呼ばれ, 物理で使われるフェーズはかつて位相とも呼ばれましたが今では相と呼ばれています.

位相空間が持っていると都合が良いものにコンパクトという性質があります. 例えば, ユークリッド空間 R^n の部分集合 M が有界な閉集合のとき集合 M はコンパクトであると言われます. コンパクトであると何が良いのかというと, コンパクトな集合 X を定義域とする連続関数 $f : M \to R$ は最大値と最小値を持つからです. つまり, 微積分の教科書に必ず載っている最大値最小値の定理を理解するには位相の知識が必要だということです.

13 無限小とは

300 年前の数学には無限小解析という分野がありました. ニュートンやライプニッツが始めた微積分学また解析学はかつて無限小解析とよばれていました. 解析学はさらにオイラー [2] によって急速に発展しています. したがって, 解析学と言ったとき解析されるものは「無限」が持っている性質です. 無限小解析の強力さを見てみましょう [6].

まず変化量 x の関数 $y = f(x)$ とします. 変化量 x が dx 増えて $x+dx$ になったときに y が dy 増えたとすれば,

$$dy = f(x+dx) - f(x) = f(x+dx) - y \tag{5}$$

つまり,

$$f(x+dx) = y + dy \tag{6}$$

です. さらに,

$$
\begin{aligned}
f(x+2dx) &= (y+dy)+d(y+dy) && (7)\\
&= y+2dy+d^2y && (8)\\
f(x+3dx) &= (y+dy)+2d(y+dy)+d^2(y+dy) && (9)\\
&= y+3dy+3d^2y+d^3y && (10)
\end{aligned}
$$

を繰替えしていけば

$$
f(x+ndx) = \sum_k^n \binom{n}{k} d^k y \tag{11}
$$

となる. ω を有限として

$$
n = \frac{\omega}{dx} \tag{12}
$$

と置き dx を無限小とみなせば, n は無限大となるから

$$
\binom{n}{k} = \frac{n^k}{k!} \tag{13}
$$

が言える. したがって

$$
\begin{aligned}
f(x+ndx) &= \sum_k^n \frac{n^k}{k!} d^k y && (14)\\
&= \sum_k^n \frac{1}{k!} \left(\frac{\omega}{dx}\right)^k d^k y && (15)\\
&= \sum_k^n \frac{\omega^k}{k!} \frac{d^k y}{dx^k} && (16)\\
&= \sum_k^n \frac{\omega^k}{k!} f^{(k)}(x) \qquad [\text{テーラーの公式}] && (17)
\end{aligned}
$$

というテーラー展開の公式が求まります.

しかし無限の性質を明らかにしていった無限小解析の方法は極めて過激で, 数学者の反発を招き, 一旦は滅ぼされてしまいました. それに代わって台頭してきたのが, 無限を直接扱わないで無限の性質を探る $\varepsilon-\delta$ 法と呼ばれる手続きです.

さらに 20 世紀に入ると無限に関する不思議な性質が明らかにされていく. 有名なものとしてラッセルのパラドックスがあります.

> 自身を要素として含まない集合の全体を集合 A とする. だから集合 A は集合 A に含まれない. したがって集合 A は集合 A に含まれる.

数学の基礎を揺がすこうした問題に対処するため, ヒルベルトは「有限の立場」に基づいたヒルベルトのプログラムを, ブラウワーは直観主義を提唱し, 数学基礎論という分野が生れました. さらに, ツェロメロやフレンケルはパラドックスが生じるような集合を集合として認めない定義を提唱し, それを基にノイマンが公理的集合論と呼ばれる ZF 集合論を形式化しています. 数学とは数学者が神の立場にたって絶対的な真理を追求する学問ですが, 直観主義とは数学者が人間の立場にたって計算可能な真理だけを追求しようという立場です. 計算機科学には計算量の理論や計算可能性の理論という分野がありますが, 計算を構成できるということは直観主義で最も重畳な問題です.

直観主義を提唱したブラウワーは初めてテューリングの論文を見たとき,「テューリングマシンこそ, 明らかに私が構成的ルールと呼ぶものである」と宣言しています [7].

さらに衝撃的な定理が表われる, それがゲーデルの完全性定理です. ゲーデルの完全性定理によればある公理を満たす数学的な構造 (モデル) は一つだけとは限らずいくらでもある, さらにスコーレムの定理によれば, 実数の性質を保ちながらも自然数と同じ大きさの集合が実数の部分集合として存在する, とまで言われてしまう. これにより 1950 年代にモデルの理論が始まっています.

このモデル理論が無限小解析が復活するきっかけでした. 1960 年代にロビンソンは実数を部分集合として含む超実数を考案し無限大や無限小の概念を厳密に定めることで, オイラーやライプニッツ龍の無限小解析をそのままの形で蘇えらせることに成功しています. このロビンソンの理論は超準解析とも呼ばれます.

無限小解析的な証明がどのようなものかを見るために, 解説しませんが最大値最小値の定理を証明してみます. つまり, 実数体 R のコンパクトな部分集合 X で定義された連続関数 f には最大値最小値が存在することを証明してみます.

実数体 R の部分集合 X はコンパクトだから, 集合 X には上限が存在し X に属する. したがって集合 X には最大元と最小元が存在する.

集合 X はコンパクトだから, 集合 X^* の任意の元 x は集合 X のある元 y に無限に近い. さらに関数 f は連続だから $f(x)$ と $f(y)$ は無限に近い. よって像 $f(X^*)$ の任意の元が有限でその標準部分が像 $f(X)$ に属する. つまり像 $f(X)$ もコンパクトである. したがって関数 f には最大値と最小値が存在する. ■

参考文献

[1] J.J. Callahan. **時空の幾何学 特殊および一般相対論の数学的基礎**. Springer-Verlag Tokyo, 2003.

[2] L. Euler. **オイラーの無限解析**. Kaimei, 2001.

[3] K. Matsuzaka. **集合=位相入門**. Iwanami, 2003.

[4] T. Okabe. **反例からみた数学**. Yuhsei, 2003.

[5] G. F. B. Riemann. **リーマン論文集**. Asakura, 2004.

[6] M. Takase. dx **と** dy **の解析学**. Nippon-Hyoryon, 2000.

[7] G. Takeuchi. **無限小解析と物理学**. Yuhsei, 2001.

第 IV 部
Suica で RFID ごっこ (PaSoRi 解析編)

(著者) oku

注意

以下の情報は嘘かもしれないし、**Windows 版 VMware と libusbwin32、snoopypro は共存不能であるため同時にインストールすると互いに問題を起こす**。この 2 点を注意。また、PaSoRi には白いの (RC-S320) とカードが立てかけられるの (RC-S310) と 2 種類ある。以下の話は全て oku の手元にある白い奴に基づいている。

14 要約

Windows でしか使えない USB デバイス解析してを UNIX でも使おうという話。USB デバイス解析のとても入門的な話。

15 動機

MMA には「鍵システム」と呼ばれる、部室の鍵を持っていなくても部員なら部室に入れるようにする仕組みが設置されている。

動作原理はすこぶる単純。外においてあるキーボードで ID とパスワードを入れるとキーボードに繋がった PC で鍵を開けるモーターを動作させるソフトが起動するようになっている。

この仕組みを使えば、原理的には PC に繋がる周辺機器はなんでも鍵にすることが出来る。そこで IC カードで近未来的に入場できたら COOL じゃないか。と。

最近流行りの RFID みたいなことをやろうと。

16 FeliCa と PaSoRi

今回は FeliCa と PaSoRi をターゲットにします。これらは普及していて安価だからです。

FeliCa はソニーが開発した非接触型 IC カードで、交通系の IC カードは殆どコレ。また電子マネーの Edy やおさいふケータイも FeliCa です。

首都圏では JR の Suica として利用されているので結構普及率は良いみたいです。

PaSoRi はソニーから発売されている FeliCa 規格の IC カードを USB 経由で読み取る装置で、これを使うと Windows 上で IC カードな電子マネーで買い物をしたり、Suica や ICOCA の残額や乗車履歴が確認できたりする。

約 3000 円で入手できるので、非接触型 IC カードリーダとしてはそれなりに安価なほうと言える。

何故か FeliCa を利用したアプリケーションは大抵 3 音節になっている。FeliCa(フェリカ)、PaSoRi(パソリ)、Suica(スイカ)、ICOCA(イコカ)、PiTaPa(ぴたぱ)、IruCa(いるか) 等。もっとも、この法則に当てはまらないものも有りますが。。

鍵システムとしては、PaSoRi を使って FeliCa の情報を読み取り、予め登録された FeliCa で鍵を開けさせるようにすれば OK。ただし問題がいくつか有ります。問題のうち最初にぶち当たるのが、

PaSoRi のドライバは Windows 向けにしか提供されていない

ということ。MMA には Wintel なマシンは一台も無いのでソニーから提供されるドライバは使えない。そればかりか、10 万円程のお金を積んで SDK を買っても、UNIX 向けのドライバは手に入らない[22]。

まぁ、無いものは作るのが基本なのでまずは作るための情報を集めましょうと。

17 解析

実は PaSoRi 自体は[23]解析が容易な部類です。今回はいわゆる使用許諾契約[24]を尊重してソフトウェアのリバースエンジニアリング[25]は一切行わずに行きます。

USB の通信を覗き見るソフトやハードウェアは各種あるが、無料で入手できてそれなりに安定しているものとして SnoopyPro[26]を使ってみます。

SnoopyPro の操作は非常に簡単。

・起動
・View → USB Devices
・(USB Devices の)File → Unpack Drivers
・(USB Devices の)File → Install Service
・リストからデバイスを探して右クリック→ Install and restart
・USB デバイスに対して何か送る
・停止ボタンをクリックしてログを眺める

一度インストールが終わった後は Restart device で始められます。

今回は、SF カードリーダーで Suica の履歴を表示させているところを覗き見して解析をします。

覗き見したデータは例えばこういう風になっています。

PC → PaSoRi

```
00 00 ff 07 f9 5c 06 00 00 03 00 00 9b 00 …A
```

PaSoRi → PC

```
00 00 ff 00 ff 00 … B
00 00 ff 13 ed 5d 12 01 01 01 05 01 cb 01 f3 1c
02 00 4b 02 4f 49 93 ff 34 00 … C
```

基本的に PC から一つ送る (A) と PaSoRi から B のパターンと長いデータ (C) という組み合わせになっていることが判りました。

みて直ぐにわかる特徴は 2 つ。`00 00 ff` で始まり、`00` で終わること。これらは共通なのでとりあえず考えないことにします。

PC → PaSoRi

```
07 f9 5c 06 00 00 03 00 00 9b …A'
```

PaSoRi → PC

```
13 ed 5d 12 01 01 01 05 01 cb 01 f3 1c 02 00 4b
02 4f 49 93 ff 34 … C'
```

[22] 10 万円程で手に入るのは Windows 上のドライバを操作するコード。リファレンスコードを買えば多分ドライバを作るだけの情報を手に入れることは出来るが、相当に高価だろう

[23] ゲームの画像ローダとか通信プロトコルとかに比べれば

[24] インストールするときに表示される契約書っぽい奴

[25] 逆アセンブルみたり逆コンパイルしたりしてプログラムのソースコードを妄想すること

[26] http://sourceforge.net/projects/usbsnoop

もっと周りを見てみると、01 01 05 01 cb 01 f3 1c のパターンが多いことが[27]解ります。似たパターンが多いということは通信が暗号化されている可能性が結構低い[28]ということとそのパターンがコマンドであったり ID であったりアドレスであったりするということを示唆します。

この辺で古き良き解析テクニックが活躍します。

ファミコンやスーパーファミコン時代のゲームは「パスワード」でゲームの経過を保存していました。パスワードをちょっと打ち間違えただけで、むちゃくちゃな HP とか MP を誇るキャラクタが出来てはゲーム的に困るので、パスワードが間違っていないかチェックする方法論が必要になります。

そういうときに利用されるのが「チェックサム」で、例えば HP と MP の合計値をパスワードの中に盛り込めば、HP と HP+MP の値の両方を打ち間違えない限りは、パスワードの打ち間違いを検出することが出来ます。

で、我々としてはパスワードを捏造したいわけで、何処が HP や MP をあらわしているのか、何処がチェックサムなのかを推理する必要が有ります。

勘の良い人はもうお気づきかと思いますが[29]、「ちょっとだけパラメータをずらして幾つもセーブすればいい」んです。ちょっとだけずらした大量のセーブデータが SnoopyPro で得られたと考えよう。

あとはセーブデータ同士を比較して法則性を探します。同じ事をこのログでやってみます。

A' の長さは 10 バイト、C' の長さは 22 バイト。A' の先頭が 7、C' の先頭が 19[30]。どちらも差が 12。つまり先頭の 1 バイトで長さを示している可能性が高いと考えられます[31]

実は 2 バイト目と 4 バイト目には特徴があります。どっちも合計すると FF。こういうバイトはチェックサムである可能性が高いです (そして実際チェックサムです)。

そして長さを示す 1 バイト目、チェックサムっぽい 2 バイト目と 4 バイト目を除いたものの総和は常に FA になっています。総和が一定になるのは偶然ではありません。総和が FA で無ければ通信エラーと判定するように出来ていると考えます。

大体このようにして通信フォーマットを探っていきます。

このセクションは長いので纏め。「差分を取れ」。

18 考察

Windows 上の PaSoRi ドライバは Program Files/Common Files/Sony Shared/FeliCaLibrary に格納されており、plugins/command にサポートされているコマンドと思しきモジュールが幾つか転がっている。このモジュールの名前をヒントに Google 検索してみると結構簡単に仕様書が見つかる。

FeliCa と通信するためにはちょうど Ethernet の MAC アドレスに当たる IDm という番号を使う。先のセクションで出てきた 01 01 05 01 cb 01 f3 1c が IDm で、世の中に 01 01 05 01 cb 01 f3 1c の IDm をもつ FeliCa は手元の 1 枚の他に存在しないので、この 8 バイトをパスワードとして使います。

なんでパスワードにするはずの 01 01 05 01 cb 01 f3 1c をこう堂々と書けるかといえば、このパスワードは「通常の方法では他人には入力できないから」です。

逆に他人に入力する方法があるならそれがそのままこの鍵システムの弱点になります。例えば部室前に設置した PaSoRi が改造されたらダメです。もっと単純にカードを落として他人に拾われたらゲームオーバーです。

そのような状況にならないように、手持ちの FeliCa を全て黒く塗ってどれが鍵かわからないようにするのは対策として弱いので、カードをなくしたら迅速に無効にできるようにする必要が有ります。また、長期的な被害を防ぐためにも、カードの持ち主を定期的に本人であるか確認する必要が有るでしょう。

---

[27] 後述するがこの数字はカードごとに異なるので実際に試してみてこのパターンが無いからといって落ち込む必要は無い。

[28] USB デバイスでプロトコルを暗号化しているものは殆どありません。逆に PaSoRi は数少ない例外で、特定の状況下で暗号通信を行います。それについては割愛。

[29] というか、これを手にとるような人がチェックサムとか知らないって事も無いような気もしてきた

[30] 十六進数で 13 は十進数で 19

[31] 実際にはもっと多くの通信に対してこういう考察を行ったうえでこのような結論を出します…かなり紆余曲折しています。

実は先のセクションのような解析をしなくても、Windows 上のアプリケーションが行った通信をそのまま取り込んで使えば OK だったりします。ただやっぱり男のロマンという事で。

19 感想

実は鍵システムの構築よりも UNIX で動く PaSoRi ライブラリの構築に傾いてしまっているので未だに設置してなかったり。。

Linux とか MacOS は世の中のスマートカード化の流れから結構取り残されていてかわいそうで。たとえば、いわゆる住基ネットで使われる IC カードを利用するソフトウェアも Windows 専用だったりします。

半分は仕様をクローズにする機器メーカのせい、もう半分はちゃんとしたスマートカード API 基盤が存在しない OS のせいなのでどっちもどっちなんですが。