数学とコンピューター

— 数学研究の現場にて —

筑波大学数学系　　森田　純

1　はじめに

近年、コンピューターと私どもの生活は、お互いにもう切り離せない関係になってきておりますし、これからの大きなテーマとしては、いかにコンピューターとつき合い、利用し、共存していくかというところにあると思います。そして、こういう波が、一般人個々にまで及んできているという訳なのです。ここでは、コンピューターとは何か？とか、コンピューターの将来は？などという大それたことではなく、数学研究の現場にいる一個人として、現在如何にコンピューターと関わっているかということを、話させて頂きたいと思います。それから、１０年以上も前に大流行した、ルービック・キューブを題材にしての群の話もしたいと思います。

2　コンピューターの利用

現在、筑波大学数学系では、各研究室等にあるパソコンとは別に、共同利用の為のワークステーションと呼ばれるコンピューターが導入されている。これは、ごく小型ではあるが大型計算機なみの性能を有しているものである。だいたい次の様な目的が主に考えられている。

コンピューター
- 数学文書作成 (特殊プログラムによる機能)
- e-mail 等のネットワーク
- 大規模計算
- プログラム開発
- 大学数学の教育
- ゲーム？（いやいやこれは余計！）

（１） 数学文書作成

主に TEX（テフとかテックとか呼ばれる）というソフトウエアーが世界的に主流である。この文章も日本語対応の LATEX で書かれている。

$$S = \frac{\max_{1<n<m} \log_2 P_n}{\lim_{x\to 0} \frac{\sin x}{x}}$$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

$$\|D_x^l D_y^k \phi\|_\beta \equiv \left(\sum_{i=1}^{s} \sum_{j=1}^{r} \int_{\gamma_i}^{\gamma_{i+1}} \int_{\mu_j}^{\mu_{j+1}} |D_x^l D_y^k \phi|^p \, dy \, dx \right)^{1/p} < \infty$$

など、基本的にはどんな数式も楽々と書けてしまう。例えば、最後の式は実際は、

```
$$\Vert D_xˆlD_yˆk\phi\Vert_\beta \equiv\left(\sum_{i = 1}ˆ{s}\sum_{j = 1}ˆ{r}
\int_{\gamma_i}ˆ{\gamma_{i + 1}}\int_{\mu_j}ˆ{\mu_{j + 1}}
\vert D_xˆlD_yˆk\phi\vertˆp\, dy\, dx \right)ˆ{1/p} < \infty$$
```

と入力すればよい。以前は、タイプライターで原稿を作り、いちいち活字指定をしていたのであるから、ずいぶん様がわりしたものである。世界的にみても数学の論文（プレプリント）はこの TEX で打たれることが圧倒的に多くなってきている。中には、「どうして数学者が活字印刷の仕事をしなければいけないのか！」と怒るむきもあるが、若手数学者にとっては、TEX は自然に覚えておかなくてはならない常識になりつつある。

（２） コンピューター・ネットワーク

これは世界各地にあるコンピューターをつないで、情報の相互交換を可能にしようとするものである。

[e-mail] 国内外の研究者との情報交換をネットワークを通じて行なう。相手のアドレスが anata@math.uni.kuni という場合に、例えば、

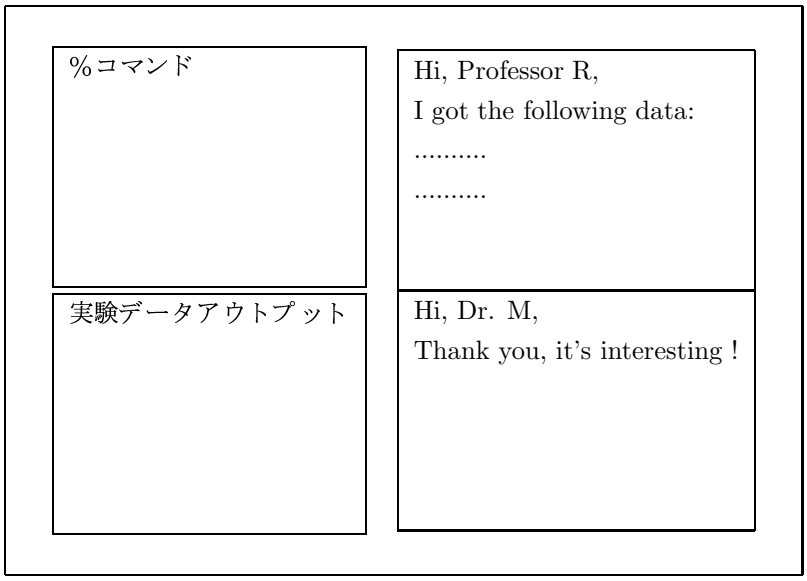
```
To: anata@math.uni.kuni
Subject: request
–text follows this line–

Dear Professor,
How are you doing ?
When will you visit Japan ?
Please send me your paper.
Sincerely yours,
J. Morita.
```

という風に画面に書いたメッセージをそのまま送ることができる。そして、外国からでも一両日中には返事が来る。さらに、TEX で書いたテキストファイルそのものを送って貰うこともできる。こちらはそれをプリントアウトすればよい。手紙でいちいちやりとりしていた頃とは、情報交換のスピードは格段の差がある。こういう意味でも TEX は、数式を記述する世界共通の言語になりつつあるといえる。

[共同研究] 離れた所にいる研究者が同時に会話しながら研究を進めることができる。例えば、

というネットワークに於いて、M氏はT大学のコンピューターで実験をしつつ、D国のコンピューターにログインして、R氏と画面上で筆談をし、さらにデータの交換もすることができる。X端末でマルチウインドウを開くことによって、全て同時に行なえるのである。



X端末画面

（３） 大規模計算

これは様々な形で以前から行なわれてたのであるが、最近そのニュアンスが少し変わってきている様に思われる。私の専門分野の代数学でもコンピューターを大きな計算に使用する人はいたが、かなり限られた一部の人達によってであった。ところが、ワークステーションの発達とともに、その上で動くソフトウェアーの開発も進み、多くの人達が使える状態になってきた。「誰でも気軽に」というのと「限られた一部の人達」とでは、その影響には大きな差がでてくる。ではここで、一つの問題を考えてみよう。

問題 椅子が６つ一列に並んでいて、端から１番、２番、… と６番まで番号がつけてあり、６人の生徒がそれぞれ椅子に座っている。先生が笛を吹いたら１番の椅子の生徒は２番へ、２番の椅子の生徒は３番へ、３番の椅子の生徒は４番へ、４番の椅子の生徒は１番へと移動し、５番と６番の椅子の生徒はそのままでいることにする。また、先生が太鼓を叩いたら３番の椅子の生徒は４番へ、４番の椅子の生徒は５番へ、５番の椅子の生徒は６番へ、６番の椅子の生徒は３番へと移動し、１番と２番の椅子の生徒はそのままでいることにする。初期の状態から、先生は笛や太鼓を色々と組み合わせて用いて、次々に生徒達を移動させていくものとする。このとき、生徒達の並び方は全部で何通りの可能性があるでしょうか？

さて、群計算用のシステム「Gap」を用いて、計算してみよう。「Gap」用に次の file を作り、 file 名を demo とする。

```
g:=Group((1,2,3,4),(3,4,5,6));;
Size(g);
gg:=CommutatorSubgroup(g,g);;
Size(gg);
IsSimple(gg);
```

ここで、ワークステーションのプロンプトで

% gap < demo RETURN

とすると、

Gap

```
gap> gap> 120
gap> gap> 60
gap> true
```

という答えが画面に出る。実は、群 g は 5 次対称群と同型で位数は 120、従ってその交換子群 gg は 5 次交代群となり、それは位数 60 の単純群であった。少し余計な計算もしたが、上の問題の答えは 120 である。こういう計算が、あたかも電卓を使う様に誰でもが、そして大量にできてしまう。手計算では限界がある様なものでも、多くの人がその結果を見ることができるのである。他に「Cayley」という群計算用のシステムや、それの後釜の「Magma」というものもあり、私は状況に応じて使い分けている。

（４）プログラム開発

これは、目的に応じて様々な計算システムを創り上げていくもので、計算機数学の専門家が日々開発を進めている。

（５） 大学数学の教育

通常、大学の２年あるいは３年の代数学の授業で群論を学ぶ。群の定義から始まり、定義・性質・例題の繰り返しのオンパレードとなる。少し複雑になると、ほとんどの学生は具体的イメージが持てないままに先に進むことになってしまう。そういう学生には、例えば、上に述べた「Gap」などが有効なのではないかと思う。基本事項を学んだあと、「Gap」の使い方を教えて、群計算の実験をさせる。自由に発見をさせながら、色々な概念を学ばせていく。こういう風にできたら面白いのになあと思っている。ただし、これを実行するには結構予算が必要だし、実際の効果がいかほどかも不明である。相当な予備実験も必要だし、取り敢えずは夢物語である。私のゼミの学生は自由に「Gap」を使いこなせるようになり、その実験データを参考にしながら、新しい構造定理を導いたりしていている。少なくとも、身近なところでは数学教育に多いに役に立っている。

3　ルービックキューブと群

ここでは、昔々に流行した「ルービック･キューブ」を、懐かしく思いだしながら、「Gap」をいじってみることにする (cf. [10],[11])。子供の玩具箱の片隅に転がっている「立体おもちゃ」を取りだしてくる。基本操作としては、６面のうちの一つを９０度回転することであり、その位数は４となる。それらで生成される、このキューブの自己同型群を cube とする。この群を、「Gap」に入れよう。

```
cube:=Group(
(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19),
(9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35),
(17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11),
(25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24),
(33,35,40,38)(34,37,39,36)(3,9,46,32)(2,12,47,29)(1,14,48,27),
(41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40)
);;
Size(cube);
Collected(Factors(last));
a:= CommutatorFactorGroup(cube);;
AbelianInvariants(a);
dcube:= DerivedSubgroup(cube);;
b:= CommutatorFactorGroup(dcube);;
AbelianInvariants(b);
IsPerfect(dcube);
c:= Centre(cube);;
Size(c);
AbelianInvariants(c);
Centre(cube);
```

これは、中心の６個以外の小面に１から４８までの番号をつけたものを考え、４８文字上の置換群として見なしたものである。これを入力すれば、

```
gap> > > > > > > > gap> gap> 43252003274489856000
gap> gap> [ [ 2, 27 ], [ 3, 14 ], [ 5, 3 ], [ 7, 2 ], [ 11, 1 ] ]
gap> gap> gap> gap> [ 2 ]
gap> gap> gap> gap> gap> gap> [  ]
gap> true
gap> gap> gap> 2
gap> [ 2 ]
gap> gap> Subgroup( Group( ( 1, 3, 8, 6)( 2, 5, 7, 4)( 9,33,25,17)(10,34,26,18)(11,35,27,19),
( 1,17,41,40)( 4,20,44,37)( 6,22,46,35)( 9,11,16,14)(10,13,15,12),
( 6,25,43,16)( 7,28,42,13)( 8,30,41,11)(17,19,24,22)(18,21,23,20),
( 3,38,43,19)( 5,36,45,21)( 8,33,48,24)(25,27,32,30)(26,29,31,28),
( 1,14,48,27)( 2,12,47,29)( 3, 9,46,32)(33,35,40,38)(34,37,39,36),
(14,22,30,38)(15,23,31,39)(16,24,32,40)(41,43,48,46)(42,45,47,44) ),
[(2, 34)(4, 10)(5, 26)(7, 18)(12, 37)(13, 20)(15, 44)(21, 28)(23, 42)(29, 36)(31, 45)(39, 47)] )
```

という出力が得られる。この群の位数は $2^{27} \cdot 3^{14} \cdot 5^3 \cdot 7^2 \cdot 11$ であり、$\mathrm{cube}^{ab} \simeq Z_2$ かつ $\mathrm{cube}' = \mathrm{cube}''$ である。さらに、中心の構造は $Z(\mathrm{cube}) \simeq Z_2$ である。具体的には、

$$\sigma = \text{ 各辺にある二面の部分だけを一斉にひっくり返す}$$

とおけば、$Z(\mathrm{cube}) = \{\ id.\ ,\ \sigma\ \}$ を得る。

上に述べたことからも、ルービックキューブの模様は、$2^{27} \cdot 3^{14} \cdot 5^3 \cdot 7^2 \cdot 11$ 通りあることが判る。最後に、「Cayley」を用いて群 cube の組成列を見てみよう。

```
cube:permutation group(48);
gens=[ (1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19),
(9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35),
(17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11),
(25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24),
(33,35,40,38)(34,37,39,36)(3,9,46,32)(2,12,47,29)(1,14,48,27),
(41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40) ];
cube.generators: gens;
print composition factors(cube);
```

と入力すれば、

```
COMPOSITION FACTORS OF GROUP CUBE

G — Cyclic(2) — Alternating(12) — Cyclic(2) — Cyclic(2) — Cyclic(2) — Cyclic(2) — Cyclic(2)
— Cyclic(2) — Cyclic(2) — Cyclic(2) — Cyclic(2) — Cyclic(2) — Cyclic(2) — Alternating(8)
— Cyclic(3) — Cyclic(3) — Cyclic(3) — Cyclic(3) — Cyclic(3) — Cyclic(3) — Cyclic(3) — 1

END OF RUN.
8.829 SECONDS
```

という出力を得る。群 cube から得られる単純群は、位数２の巡回群、位数３の巡回群、８次の交代群、１２次の交代群の４種類である。時間にして９秒で得られる。繰り返しになるが、こういう面倒な計算も気軽にできる時代になりつつあるんだなー、と思いつつ話しの終りにしたい。