

第7章： 多次元超立体魔方陣の最新研究： 摂田 寛二

多次元超立体魔方陣の展開見取り図が何通り描けるかの具体的な研究から、思いもよらぬ魔方陣構成法を発見した。これが、驚くほど簡潔で高速な構成法だ。最初は自分でも信じ難い気がしたが、テストをいろいろ重ねて、この方法で様々なタイプの魔方陣を作れることがわかった。必ずしも万能ではないが、多次元超立体魔方陣が今まで作って来たものは、大概この方法で作れる。他の方法では難しかった高次方陣も、簡単に作れる。順次報告する。

第5節： 四次元超立体二方陣の最新構成法

1. 四次元超立体二方陣の基本図として「展開見取り図」が何通り描けるか、という問題に関しては、既に第1節で概略を論じた。n1=1のタイプが24通りあることが予想される。

その具体的な描法は、4重のfor(.条件式.){...}ループを用いて二進数を発生させながら、d0,d1,d2,d3の順序を入れ替えて作るのであった。

n1に2以上の数値を来させるためには、一部for(.カウントダウン型の条件式.){...}を用いる。全然用いない場合は、n1=1になるが、全部に用いるとn1=16となる。ループが4重なので、n1が取り得る値は、 $2^4=16$ 通りあることになる。

つまり、展開見取り図は、総数が $24 \times 16 = 384$ 個描けると予想される。

専用プログラムを組んで、その展開見取り図を描かせてみた。成果を示す。今はこれに紙数をあまり使いたくないので、後半は二次元四方陣に展開した図(対称型のみ)を掲げる。

**** 四次元超立体二方陣の基本図 [展開見取り図 : 24通り+16通り] ****

**** タイプ #1 ****

1/d0	/d1	SC/	CC/
1---- 2 9---+10 5-- - 6 13----14	3---- 4 11---+12 7-- - 8 15----16	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	1 2 4 3 5 6 8 7 13 14 16 15 9 10 12 11
2/d0	/d1	SC/	CC/
1---- 3 9---+11 5-- - 7 13----15	2---- 4 10---+12 6-- - 8 14----16	1 3 2 4 5 7 6 8 9 11 10 12 13 15 14 16	1 3 4 2 5 7 8 6 13 15 16 14 9 11 12 10
3/d0	/d1	SC/	CC/
1---- 2 9---+10 3-- - 4 11----12	5---- 6 13---+14 7-- - 8 15----16	1 2 5 6 3 4 7 8 9 10 13 14 11 12 15 16	1 2 6 5 3 4 8 7 11 12 16 15 9 10 14 13
4/d0	/d1	SC/	CC/
1---- 3 9---+11 2-- - 4 10----12	5---- 7 13---+15 6-- - 8 14----16	1 3 5 7 2 4 6 8 9 11 13 15 10 12 14 16	1 3 7 5 2 4 8 6 10 12 16 14 9 11 15 13
5/d0	/d1	SC/	CC/
1---- 5 9---+13 3-- - 7 11----15	2---- 6 10---+14 4-- - 8 12----16	1 5 2 6 3 7 4 8 9 13 10 14 11 15 12 16	1 5 6 2 3 7 8 4 11 15 16 12 9 13 14 10
6/d0	/d1	SC/	CC/
1---- 5 9---+13 2-- - 6 10----14	3---- 7 11---+15 4-- - 8 12----16	1 5 3 7 2 6 4 8 9 13 11 15 10 14 12 16	1 5 7 3 2 6 8 4 10 14 16 12 9 13 15 11

7/ 1 2 3 4 9 10 11 12 5 6 7 8 13 14 15 16	8/ 1 3 2 4 9 11 10 12 5 7 6 8 13 15 14 16	9/ 1 2 5 6 9 10 13 14 3 4 7 8 11 12 15 16	10/ 1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16	11/ 1 5 2 6 9 13 10 14 3 7 4 8 11 15 12 16	12/ 1 5 3 7 9 13 11 15 2 6 4 8 10 14 12 16
13/ 1 2 9 10 3 4 11 12 5 6 13 14 7 8 15 16	14/ 1 3 9 11 2 4 10 12 5 7 13 15 6 8 14 16	15/ 1 2 9 10 5 6 13 14 3 4 11 12 7 8 15 16	16/ 1 3 9 11 5 7 13 15 2 4 10 12 6 8 14 16	17/ 1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16	18/ 1 5 9 13 3 7 11 15 2 6 10 14 4 8 12 16
19/ 1 9 2 10 3 11 4 12 5 13 6 14 7 15 8 16	20/ 1 9 3 11 2 10 4 12 5 13 7 15 6 14 8 16	21/ 1 9 2 10 5 13 6 14 3 11 4 12 7 15 8 16	22/ 1 9 3 11 5 13 7 15 2 10 4 12 6 14 8 16	23/ 1 9 5 13 2 10 6 14 3 11 7 15 4 12 8 16	24/ 1 9 5 13 3 11 7 15 2 10 6 14 4 12 8 16

**** タイプ #2 ****

1/ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	2/ 2 1 4 3 6 5 8 7 10 9 12 11 14 13 16 15	3/ 3 4 1 2 7 8 5 6 11 12 9 10 15 16 13 14	4/ 4 3 2 1 8 7 6 5 12 11 10 9 16 15 14 13
5/ 5 6 7 8 1 2 3 4 13 14 15 16 9 10 11 12	6/ 6 5 8 7 2 1 4 3 14 13 16 15 10 9 12 11	7/ 7 8 5 6 3 4 1 2 15 16 13 14 11 12 9 10	8/ 8 7 6 5 4 3 2 1 16 15 14 13 12 11 10 9
9/ 9 10 11 12 13 14 15 16 1 2 3 4 5 6 7 8	10/ 10 9 12 11 14 13 16 15 2 1 4 3 6 5 8 7	11/ 11 12 9 10 15 16 13 14 3 4 1 2 7 8 5 6	12/ 12 11 10 9 16 15 14 13 4 3 2 1 8 7 6 5
13/ 13 14 15 16 9 10 11 12 5 6 7 8 1 2 3 4	14/ 14 13 16 15 10 9 12 11 6 5 8 7 2 1 4 3	15/ 15 16 13 14 11 12 9 10 7 8 5 6 3 4 1 2	16/ 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

上の表の 24 通りに下の表の 16 通りを組み合わせると、 $24 \times 16 = 384$ 通りの展開見取り図が描ける。やり方は、4重のループの中で d_0, d_1, d_2, d_3 の順序を入れ替えることと、`for(. カウントダウン型の条件式.) {...}` を用いること。この2つをただ組み合わせるだけだ。

ただし、言うはたやすいが、プログラムに書くのは中々に難しい。小生の工夫は、

- (1) 予め d_0, d_1, d_2, d_3 の順列を計算し、具体的な順序を配列に登録しておく。
- (2) `for(. カウントダウン型の条件式.) {...}` を用いる順序も、配列に登録しておく。

それを順次読み出して4重のループを適宜制御しながら、作業を進めることにした。配列から読み出す時に「間接アドレッシング」を用いる。これは、かなり高級な技法だ。

2. 出来た展開見取り図 384 通りを見ていて、次第に不思議の感情が膨らんだ。

- (1) いったい何故にまた 384 個なのか。

384 個というのは、補数対称型の四方陣および各種の汎四方陣の「原始解」の総数だ。当然正方連結完全四方陣の総数も、完全オイラー型の汎四方陣の総数も、皆 384 個だ。

- (2) 図を並べて、思った。かって作った「原初四方陣」に何故にそっくりなのか。

原初方陣は、基本配列を模倣した方陣で、第一代表解はまさに基本配列そのものである。今回の展開図の最初の1つ (SC タイプ) もまた基本配列と同じ形をしている (「基本図」を描いているのだから、当然なのではあるが)。

全体としても、そうなる。理由の全貌はまだわからないが、確かにそうなるのだ。

ならば、適当な「ひな形」をさがしてやろう。そうすれば、この展開図から **384** 個の解を直接に構成することができるのではないか。さらに別に適当な「変換法」を考えてやれば、何のタイプの魔方陣でも作れるのではないか（原始解の総数が **384** 個のものに限って）。

かなり図々しいひらめきであるが、できる予感がした。

基本図から解を作り出す「ひな形」を見つけるにはどうするか。この探索は、決して初めての経験ではない。かって「原初方陣」で再三汗した苦勞が役に立つのではないか。

結局、元になる基本図の特徴を一番多く残す解の中から下図の例を選んで、「ひな形変換法」を考えることにした。半分の **4** 組 **8** 個の数字を「点対称移動」する。或いは、**8** 個の数を「補数変換」するのである。プログラムは、例えば次のように書く。

補数対称型四方陣を完全型四方陣に変換するには、いつもの方法が使える。縦方向にも横方向にも同時に第3行/列と第4行/列とを相互に交換する。汎対角線の定和が成り立つだけでなく、正方連結四数の定和も成り立つようになるはずだ。

```
/*
BD/d0          /d1          SC/
 1---- 2        3---- 4        1  2  3  4
 |  9---+10     | 11---+12     5  6  7  8
 5--|- 6 |     7--|- 8 |     9 10 11 12
   13----14     15----16     13 14 15 16

MD/d0          /d1          SC/          CC/
 1----15        14---- 4        1 15 14  4    1 15  4 14
 |  8---+10     | 11---+ 5     12  6  7  9    12  6  9  7
 12--|- 6 |     7--|- 9 |     8 10 11  5    13  3 16  2
   13---- 3        2----16     13  3  2 16    8 10  5 11

*/
//
CC=17;
y[1]=x[1];      y[2]=CC-x[2];      y[3]=CC-x[3];      y[4]=x[4];
y[5]=CC-x[5];  y[6]=x[6];      y[7]=x[7];      y[8]=CC-x[8];
y[9]=CC-x[9];  y[10]=x[10];     y[11]=x[11];     y[12]=CC-x[12];
y[13]=x[13];   y[14]=CC-x[14];   y[15]=CC-x[15];   y[16]=x[16];
//
```

実際のプログラムは、次のように書いた。今は対象を1つに絞り補数対称四方陣の原始解 **384** 個のみを構成する。構成部品を少なくしてわかりやすさを優先するためである。

```
/** Newest Method of Composing S-C Magic Squares of Order 4^2 **
/** Using All View-Diagrams of Developed Extra-Cubic Objects 2^4 **
/** File: 'SNewMS4.c' Dictated by Kanji Setsuda on **
/** Apr.26,'05 and Jan.24,'06 with MacOSX & Xcode 2.1 **
/** Recompiled on Mar. 5, 2015 with MacOSX & Xcode 6.1.1 **
//
#include <stdio.h>
//
short tcnt;
short vc, pc;
short tv[16][4];
short td[24][4];
short nm[17];
short pnm[385][17], tnm[385][17];
//
void mkfv16(void), prmt24(void);
void d24(short x, short y);
void mkso124(void);
```



```

        uflg[d1]=0;
    }}
    uflg[d0]=0;
}
}
//
void d24(short x, short y){
    short d0,d1,d2,d3;
    short t0,t1,t2,t3;
    short CC,c,n;
    short s[4], t[4][2];
    CC=1;
    for(n=0;n<4;n++){
        t[n][0]=tv[x][n]; t[n][1]=CC-tv[x][n];}
    //
    c=0;
    for(d0=0;d0<2;d0++){
        for(d1=0;d1<2;d1++){
            for(d2=0;d2<2;d2++){
                for(d3=0;d3<2;d3++){c++;
                    s[0]=t[0][d0]; s[1]=t[1][d1]; s[2]=t[2][d2]; s[3]=t[3][d3];
                    t0=td[y][0]; t1=td[y][1]; t2=td[y][2]; t3=td[y][3];
                    pnm[x*pc+y][c]=((s[t0]*2+s[t1])*2+s[t2])*2+s[t3]+1;
                }
            }
        }
    }
}
//
void mkso124(void){
    short m,n,CC;
    CC=17;
    for(m=0;m<tcnt;m++){
        nm[1]=pnm[m][1]; nm[2]=CC-pnm[m][2]; nm[3]=CC-pnm[m][3]; nm[4]=pnm[m][4];
        nm[5]=CC-pnm[m][5]; nm[6]=pnm[m][6]; nm[7]=pnm[m][7]; nm[8]=CC-pnm[m][8];
        nm[9]=CC-pnm[m][9]; nm[10]=pnm[m][10]; nm[11]=pnm[m][11]; nm[12]=CC-pnm[m][12];
        nm[13]=pnm[m][13]; nm[14]=CC-pnm[m][14]; nm[15]=CC-pnm[m][15]; nm[16]=pnm[m][16];
        for(n=1;n<17;n++){tnm[m][n]=nm[n];}
    }
}
//
/** Print each Prototype and Object Solutions */
void pr32ans(short x, short y){
    short m,l,l4,n,p;
    for(m=x-1;m<y;m=m+3){
        printf("      P/%3d/S          P/%3d/S          P/%3d/S\n",
            m+1,m+2,m+3);
        for(l=0;l<4;l++){l4=l*4;
            for(p=0;p<3;p++){
                printf(" ");
                for(n=0;n<4;n++){printf("%3d",pnm[m+p][l4+n+1]);}
                printf(" ");
                for(n=0;n<4;n++){printf("%3d",tnm[m+p][l4+n+1]);}
                if(p<2){printf(" ");}
            }
            printf("\n");
        }
    }
}
//
/** Sort Program for the Smart List of Solutions */
void solsort(void){
    int d0,d1;
    short mx,n,f;
    for(mx=tcnt-1;mx>0;mx--){

```

```

do{f=0;
  for(n=0;n<mx;n++){
    d0=(((pnm[n][1]*16+pnm[n][2])*16+pnm[n][3])*16+pnm[n][4])*16+pnm[n][5];
    d1=(((pnm[n+1][1]*16+pnm[n+1][2])*16+pnm[n+1][3])*16+pnm[n+1][4])*16+pnm[n+1][5];
    if(d0>d1){exc(n); f=1;}
    if((d0==d1)&&(pnm[n][6]>pnm[n+1][6])){exc(n); f=1;}
  }
}while(f>0);
}
//
void exc(short x){
  short n,d;
  for(n=1;n<17;n++){d=pnm[x][n]; pnm[x][n]=pnm[x+1][n]; pnm[x+1][n]=d;}
}
//

```

プログラムの書き方が今までと全然違うのに、気付いた方も居られよう。
 何しろ「展開見取り図」を描いてそれを補数変換するだけなのだから、簡単である。
 第一に、例の定義の連立方程式が無い。それを計算して探索する作業がどこにも無い。
 以前は最大で $16P_{16} = 16 \times 15 \times 14 \times 13 \times \dots \times 3 \times 2 \times 1$ 通りもの数値を代入して調べたもの
 なのに、今はそれが無い。そういう無駄が無いから、実行スピードも速そうだ。
 こんな不思議なプログラムで、本当に正しい解が 384 個も作れるものだろうか。

3. 実行結果は次の通りである。

**** 四次元二方陣の見取り図を利用する新構成法 ****

** Comparative List of the 384 Pairs of Primitive Solutions: **

P/ 1/S				P/ 2/S				P/ 3/S			
1 2 3 4	1 15 14 4	1 2 3 4	1 15 14 4	1 2 5 6	1 15 12 6						
5 6 7 8	12 6 7 9	9 10 11 12	8 10 11 5	3 4 7 8	14 4 7 9						
9 10 11 12	8 10 11 5	5 6 7 8	12 6 7 9	9 10 13 14	8 10 13 3						
13 14 15 16	13 3 2 16	13 14 15 16	13 3 2 16	11 12 15 16	11 5 2 16						
P/ 4/S				P/ 5/S				P/ 6/S			
1 2 5 6	1 15 12 6	1 2 9 10	1 15 8 10	1 2 9 10	1 15 8 10						
9 10 13 14	8 10 13 3	3 4 11 12	14 4 11 5	5 6 13 14	12 6 13 3						
3 4 7 8	14 4 7 9	5 6 13 14	12 6 13 3	3 4 11 12	14 4 11 5						
11 12 15 16	11 5 2 16	7 8 15 16	7 9 2 16	7 8 15 16	7 9 2 16						
P/ 7/S				P/ 8/S				P/ 9/S			
1 3 2 4	1 14 15 4	1 3 2 4	1 14 15 4	1 3 5 7	1 14 12 7						
5 7 6 8	12 7 6 9	9 11 10 12	8 11 10 5	2 4 6 8	15 4 6 9						
9 11 10 12	8 11 10 5	5 7 6 8	12 7 6 9	9 11 13 15	8 11 13 2						
13 15 14 16	13 2 3 16	13 15 14 16	13 2 3 16	10 12 14 16	10 5 3 16						
P/ 10/S				P/ 11/S				P/ 12/S			
1 3 5 7	1 14 12 7	1 3 9 11	1 14 8 11	1 3 9 11	1 14 8 11						
9 11 13 15	8 11 13 2	2 4 10 12	15 4 10 5	5 7 13 15	12 7 13 2						
2 4 6 8	15 4 6 9	5 7 13 15	12 7 13 2	2 4 10 12	15 4 10 5						
10 12 14 16	10 5 3 16	6 8 14 16	6 9 3 16	6 8 14 16	6 9 3 16						
P/ 13/S				P/ 14/S				P/ 15/S			
1 5 2 6	1 12 15 6	1 5 2 6	1 12 15 6	1 5 3 7	1 12 14 7						
3 7 4 8	14 7 4 9	9 13 10 14	8 13 10 3	2 6 4 8	15 6 4 9						
9 13 10 14	8 13 10 3	3 7 4 8	14 7 4 9	9 13 11 15	8 13 11 2						
11 15 12 16	11 2 5 16	11 15 12 16	11 2 5 16	10 14 12 16	10 3 5 16						
P/ 16/S				P/ 17/S				P/ 18/S			
1 5 3 7	1 12 14 7	1 5 9 13	1 12 8 13	1 5 9 13	1 12 8 13						
9 13 11 15	8 13 11 2	2 6 10 14	15 6 10 3	3 7 11 15	14 7 11 2						
2 6 4 8	15 6 4 9	3 7 11 15	14 7 11 2	2 6 10 14	15 6 10 3						
10 14 12 16	10 3 5 16	4 8 12 16	4 9 5 16	4 8 12 16	4 9 5 16						

P/ 19/S
 1 9 2 10 1 8 15 10
 3 11 4 12 14 11 4 5
 5 13 6 14 12 13 6 3
 7 15 8 16 7 2 9 16

P/ 20/S
 1 9 2 10 1 8 15 10
 5 13 6 14 12 13 6 3
 3 11 4 12 14 11 4 5
 7 15 8 16 7 2 9 16

P/ 21/S
 1 9 3 11 1 8 14 11
 2 10 4 12 15 10 4 5
 5 13 7 15 12 13 7 2
 6 14 8 16 6 3 9 16

P/ 22/S
 1 9 3 11 1 8 14 11
 5 13 7 15 12 13 7 2
 2 10 4 12 15 10 4 5
 6 14 8 16 6 3 9 16

P/ 23/S
 1 9 5 13 1 8 12 13
 2 10 6 14 15 10 6 3
 3 11 7 15 14 11 7 2
 4 12 8 16 4 5 9 16

P/ 24/S
 1 9 5 13 1 8 12 13
 3 11 7 15 14 11 7 2
 2 10 6 14 15 10 6 3
 4 12 8 16 4 5 9 16

P/ 25/S
 2 1 4 3 2 16 13 3
 6 5 8 7 11 5 8 10
 10 9 12 11 7 9 12 6
 14 13 16 15 14 4 1 15

P/ 26/S
 2 1 4 3 2 16 13 3
 10 9 12 11 7 9 12 6
 6 5 8 7 11 5 8 10
 14 13 16 15 14 4 1 15

P/ 27/S
 2 1 6 5 2 16 11 5
 4 3 8 7 13 3 8 10
 10 9 14 13 7 9 14 4
 12 11 16 15 12 6 1 15

P/ 28/S
 2 1 6 5 2 16 11 5
 10 9 14 13 7 9 14 4
 4 3 8 7 13 3 8 10
 12 11 16 15 12 6 1 15

P/ 29/S
 2 1 10 9 2 16 7 9
 4 3 12 11 13 3 12 6
 6 5 14 13 11 5 14 4
 8 7 16 15 8 10 1 15

P/ 30/S
 2 1 10 9 2 16 7 9
 6 5 14 13 11 5 14 4
 4 3 12 11 13 3 12 6
 8 7 16 15 8 10 1 15

P/ 31/S
 2 4 1 3 2 13 16 3
 6 8 5 7 11 8 5 10
 10 12 9 11 7 12 9 6
 14 16 13 15 14 1 4 15

P/ 32/S
 2 4 1 3 2 13 16 3
 10 12 9 11 7 12 9 6
 6 8 5 7 11 8 5 10
 14 16 13 15 14 1 4 15

P/ 33/S
 2 4 6 8 2 13 11 8
 1 3 5 7 16 3 5 10
 10 12 14 16 7 12 14 1
 9 11 13 15 9 6 4 15

P/ 34/S
 2 4 6 8 2 13 11 8
 10 12 14 16 7 12 14 1
 1 3 5 7 16 3 5 10
 9 11 13 15 9 6 4 15

P/ 35/S
 2 4 10 12 2 13 7 12
 1 3 9 11 16 3 9 6
 6 8 14 16 11 8 14 1
 5 7 13 15 5 10 4 15

P/ 36/S
 2 4 10 12 2 13 7 12
 6 8 14 16 11 8 14 1
 1 3 9 11 16 3 9 6
 5 7 13 15 5 10 4 15

P/ 37/S
 2 6 1 5 2 11 16 5
 4 8 3 7 13 8 3 10
 10 14 9 13 7 14 9 4
 12 16 11 15 12 1 6 15

P/ 38/S
 2 6 1 5 2 11 16 5
 10 14 9 13 7 14 9 4
 4 8 3 7 13 8 3 10
 12 16 11 15 12 1 6 15

P/ 39/S
 2 6 4 8 2 11 13 8
 1 5 3 7 16 5 3 10
 10 14 12 16 7 14 12 1
 9 13 11 15 9 4 6 15

P/ 40/S
 2 6 4 8 2 11 13 8
 10 14 12 16 7 14 12 1
 1 5 3 7 16 5 3 10
 9 13 11 15 9 4 6 15

P/ 41/S
 2 6 10 14 2 11 7 14
 1 5 9 13 16 5 9 4
 4 8 12 16 13 8 12 1
 3 7 11 15 3 10 6 15

P/ 42/S
 2 6 10 14 2 11 7 14
 4 8 12 16 13 8 12 1
 1 5 9 13 16 5 9 4
 3 7 11 15 3 10 6 15

P/ 43/S
 2 10 1 9 2 7 16 9
 4 12 3 11 13 12 3 6
 6 14 5 13 11 14 5 4
 8 16 7 15 8 1 10 15

P/ 44/S
 2 10 1 9 2 7 16 9
 6 14 5 13 11 14 5 4
 4 12 3 11 13 12 3 6
 8 16 7 15 8 1 10 15

P/ 45/S
 2 10 4 12 2 7 13 12
 1 9 3 11 16 9 3 6
 6 14 8 16 11 14 8 1
 5 13 7 15 5 4 10 15

P/ 46/S
 2 10 4 12 2 7 13 12
 6 14 8 16 11 14 8 1
 1 9 3 11 16 9 3 6
 5 13 7 15 5 4 10 15

P/ 47/S
 2 10 6 14 2 7 11 14
 1 9 5 13 16 9 5 4
 4 12 8 16 13 12 8 1
 3 11 7 15 3 6 10 15

P/ 48/S
 2 10 6 14 2 7 11 14
 4 12 8 16 13 12 8 1
 1 9 5 13 16 9 5 4
 3 11 7 15 3 6 10 15

P/ 49/S
 3 1 4 2 3 16 13 2
 7 5 8 6 10 5 8 11
 11 9 12 10 6 9 12 7
 15 13 16 14 15 4 1 14

P/ 50/S
 3 1 4 2 3 16 13 2
 11 9 12 10 6 9 12 7
 7 5 8 6 10 5 8 11
 15 13 16 14 15 4 1 14

P/ 51/S
 3 1 7 5 3 16 10 5
 4 2 8 6 13 2 8 11
 11 9 15 13 6 9 15 4
 12 10 16 14 12 7 1 14

.....

<p>P/190/S</p> <pre>8 16 6 14 8 1 11 14 7 15 5 13 10 15 5 4 4 12 2 10 13 12 2 7 3 11 1 9 3 6 16 9</pre>	<p>P/191/S</p> <pre>8 16 7 15 8 1 10 15 4 12 3 11 13 12 3 6 6 14 5 13 11 14 5 4 2 10 1 9 2 7 16 9</pre>	<p>P/192/S</p> <pre>8 16 7 15 8 1 10 15 6 14 5 13 11 14 5 4 4 12 3 11 13 12 3 6 2 10 1 9 2 7 16 9</pre>
<p>P/193/S</p> <pre>9 1 10 2 9 16 7 2 11 3 12 4 6 3 12 13 13 5 14 6 4 5 14 11 15 7 16 8 15 10 1 8</pre>	<p>P/194/S</p> <pre>9 1 10 2 9 16 7 2 13 5 14 6 4 5 14 11 11 3 12 4 6 3 12 13 15 7 16 8 15 10 1 8</pre>	<p>P/195/S</p> <pre>9 1 11 3 9 16 6 3 10 2 12 4 7 2 12 13 13 5 15 7 4 5 15 10 14 6 16 8 14 11 1 8</pre>
<p>P/196/S</p> <pre>9 1 11 3 9 16 6 3 13 5 15 7 4 5 15 10 10 2 12 4 7 2 12 13 14 6 16 8 14 11 1 8</pre>	<p>P/197/S</p> <pre>9 1 13 5 9 16 4 5 10 2 14 6 7 2 14 11 11 3 15 7 6 3 15 10 12 4 16 8 12 13 1 8</pre>	<p>P/198/S</p> <pre>9 1 13 5 9 16 4 5 11 3 15 7 6 3 15 10 10 2 14 6 7 2 14 11 12 4 16 8 12 13 1 8</pre>
.		
<p>P/373/S</p> <pre>16 14 8 6 16 3 9 6 12 10 4 2 5 10 4 15 15 13 7 5 2 13 7 12 11 9 3 1 11 8 14 1</pre>	<p>P/374/S</p> <pre>16 14 8 6 16 3 9 6 15 13 7 5 2 13 7 12 12 10 4 2 5 10 4 15 11 9 3 1 11 8 14 1</pre>	<p>P/375/S</p> <pre>16 14 12 10 16 3 5 10 8 6 4 2 9 6 4 15 15 13 11 9 2 13 11 8 7 5 3 1 7 12 14 1</pre>
<p>P/376/S</p> <pre>16 14 12 10 16 3 5 10 15 13 11 9 2 13 11 8 8 6 4 2 9 6 4 15 7 5 3 1 7 12 14 1</pre>	<p>P/377/S</p> <pre>16 14 15 13 16 3 2 13 8 6 7 5 9 6 7 12 12 10 11 9 5 10 11 8 4 2 3 1 4 15 14 1</pre>	<p>P/378/S</p> <pre>16 14 15 13 16 3 2 13 12 10 11 9 5 10 11 8 8 6 7 5 9 6 7 12 4 2 3 1 4 15 14 1</pre>
<p>P/379/S</p> <pre>16 15 8 7 16 2 9 7 12 11 4 3 5 11 4 14 14 13 6 5 3 13 6 12 10 9 2 1 10 8 15 1</pre>	<p>P/380/S</p> <pre>16 15 8 7 16 2 9 7 14 13 6 5 3 13 6 12 12 11 4 3 5 11 4 14 10 9 2 1 10 8 15 1</pre>	<p>P/381/S</p> <pre>16 15 12 11 16 2 5 11 8 7 4 3 9 7 4 14 14 13 10 9 3 13 10 8 6 5 2 1 6 12 15 1</pre>
<p>P/382/S</p> <pre>16 15 12 11 16 2 5 11 14 13 10 9 3 13 10 8 8 7 4 3 9 7 4 14 6 5 2 1 6 12 15 1</pre>	<p>P/383/S</p> <pre>16 15 14 13 16 2 3 13 8 7 6 5 9 7 6 12 12 11 10 9 5 11 10 8 4 3 2 1 4 14 15 1</pre>	<p>P/384/S</p> <pre>16 15 14 13 16 2 3 13 12 11 10 9 5 11 10 8 8 7 6 5 9 7 6 12 4 3 2 1 4 14 15 1</pre>

[Counts=384] OK!

** Recompiled and Listed by Kanji Setsuda on
Mar. 5, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **

全数検査したところ、全てが正しい「対称四方陣」の解であった。重複も欠落も無い。いよいよ不思議の感情が膨らむ。何故にこんなことが可能なのだろうか。

4. 次は、汎対角線定和の成り立つ**完全四方陣**を作ってみよう。今までのプログラムに幾つかの部品を足すだけで。

- (1) 補数対称型から完全型に変換するプログラム
- (2) リスト清書用のソート・プログラム

型変換のプログラムは、これまでに何度も使って来たものと同じだ。下図に示した。

<p>1/d0</p> <pre>1---- 2 9--+10 5-- - 6 13----14</pre>	<p>/d1</p> <pre>3---- 4 11--+12 7-- - 8 15----16</pre>	<p>SC/</p> <pre>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</pre>	<p>CC/</p> <pre>1 2 4 3 5 6 8 7 13 14 16 15 9 10 12 11</pre>
--	--	--	--


```

d[1]=c[1];    d[2]=c[2];    d[3]=c[4];    d[4]=c[3];
d[5]=c[5];    d[6]=c[6];    d[7]=c[8];    d[8]=c[7];
d[9]=c[13];   d[10]=c[14];   d[11]=c[16];  d[12]=c[15];
d[13]=c[9];   d[14]=c[10];   d[15]=c[12];  d[16]=c[11];

```

少し長くて恐縮だが、プログラムの全リストを示そう。

```

/** Composing 'Complete' Magic Squares 4x4 by the Newest Method **
/** Using All View-Diagrams for Developed ExtCubic Objects 2^4: **
/** List of the 384 Pairs of Prototype Squares and Complete MS44 **
/** File: 'S2NewMS4.c' Dictated by Kanji Setsuda in **
/** 2005 and 2006 with MacOSX & Xcode2.1; Recompiled on **
/** March 5, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **
//
#include <stdio.h>
//
short cnt, tcnt;
short vc, pc;
short tv[16][4];
short td[24][4];
short LSM, PSM;
short nm[17], uflg[17];
short snm[385][17];
short pnm[385][17], tnm[385][17];
short onmb[385];
//
void mkfv16(void), prmt24(void);
void d24(short x, short y);
void mksol24(void), tconv24(void);
void pr32ans(short x, short y);
void solsort(void), exc(short x);
//
void stp01(void), stp02(void), stp03(void), stp04(void);
void stp05(void), stp06(void), stp07(void), stp08(void);
void recsol(void), chkmon(void);
void prmon(void);
//
/** Main Program *
int main(){
short m,n;
LSM=34; PSM=17; cnt=0;
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
stp01();
printf("\n");
printf("** Compose 'Complete' Pan-magic Squares 4^2 by the Newest Method **\n");
printf("** Using All View-Diagrams for Developed Extra-Cubic Objects 2^4: **\n");
printf("** List of 384 Pairs of Prototype Squares and Complete MS44 Composed **\n");
vc=0; mkfv16();
pc=0; prmt24();
tcnt=pc*vc;
for(m=0;m<vc;m++){for(n=0;n<pc;n++){d24(m,n);}}
solsort();
mksol24();
tconv24();
for(n=0;n<385;n++){onmb[n]=0;}
chkmon(); // Check Monitor of Correspondence to the Old List
pr32ans(1,54);
printf(" . . . . .\n");
pr32ans(70,78);
printf(" . . . . .\n");
pr32ans(94,102);
printf(" . . . . .\n");
pr32ans(190,198);

```

```

printf(" . . . . .\n");
pr32ans(tcnt-14,tcnt);
printf("\n");
printf("*** Monitor of Solution Correspondences between Old and New **\n");
prmon();
printf("\n");
printf(" [Count(Proto/Comp1) = %d/%d] OK!\n",tcnt,cnt);
printf("\n");
printf("*** Recompiled and Listed by Kanji Setsuda on\n");
printf("   March 5, 2015 with MacOSX 10.10.2 & Xcode 6.1.1 **\n");
printf("\n");
return 0;
}
//
/* Sub-Procedures for EC02^4 *
void mkfv16(void){
short d0,d1,d2,d3;
for(d0=0;d0<2;d0++){
for(d1=0;d1<2;d1++){
for(d2=0;d2<2;d2++){
for(d3=0;d3<2;d3++){
tv[vc][0]=d0; tv[vc][1]=d1; tv[vc][2]=d2; tv[vc][3]=d3;
vc++;
}
}
}
}
}
//
void prmt24(void){
short d0,d1,d2,d3,n;
short uf1g[4];
for(n=0;n<4;n++){uf1g[n]=0;}
for(d0=0;d0<4;d0++){
uf1g[d0]=1;
for(d1=0;d1<4;d1++){
if(uf1g[d1]==0){uf1g[d1]=1;
for(d2=0;d2<4;d2++){
if(uf1g[d2]==0){uf1g[d2]=1;
for(d3=0;d3<4;d3++){
if(uf1g[d3]==0){uf1g[d3]=1;
td[pc][0]=d0; td[pc][1]=d1; td[pc][2]=d2; td[pc][3]=d3;
pc++;
uf1g[d3]=0;
}
}
}
}
uf1g[d2]=0;
}
}
uf1g[d1]=0;
}
}
uf1g[d0]=0;
}
}
//
void d24(short x, short y){
short d0,d1,d2,d3;
short t0,t1,t2,t3;
short CC,c,n;
short s[4], t[4][2];
CC=1;
for(n=0;n<4;n++){
t[n][0]=tv[x][n]; t[n][1]=CC-tv[x][n];}
//
c=0;
for(d0=0;d0<2;d0++){
for(d1=0;d1<2;d1++){

```

```

    for(d2=0;d2<2;d2++){
        for(d3=0;d3<2;d3++){c++;
            s[0]=t[0][d0]; s[1]=t[1][d1]; s[2]=t[2][d2]; s[3]=t[3][d3];
            t0=td[y][0]; t1=td[y][1]; t2=td[y][2]; t3=td[y][3];
            pnm[x*pc+y][c]=((s[t0]*2+s[t1])*2+s[t2])*2+s[t3]+1;
        }
    }
}
//
void mkso124(void){
    short m,n,CC;
    CC=17;
    for(m=0;m<tcnt;m++){
        nm[1]=pnm[m][1];    nm[2]=CC-pnm[m][2];    nm[3]=CC-pnm[m][3];    nm[4]=pnm[m][4];
        nm[5]=CC-pnm[m][5]; nm[6]=pnm[m][6];    nm[7]=pnm[m][7];    nm[8]=CC-pnm[m][8];
        nm[9]=CC-pnm[m][9]; nm[10]=pnm[m][10];    nm[11]=pnm[m][11];    nm[12]=CC-pnm[m][12];
        nm[13]=pnm[m][13]; nm[14]=CC-pnm[m][14]; nm[15]=CC-pnm[m][15]; nm[16]=pnm[m][16];
        for(n=1;n<17;n++){tnm[m][n]=nm[n];}
    }
}
//
void tconv24(void){
    short m,n;
    for(m=0;m<tcnt;m++){
        nm[1]=tnm[m][1];    nm[2]=tnm[m][2];    nm[3]=tnm[m][4];    nm[4]=tnm[m][3];
        nm[5]=tnm[m][5];    nm[6]=tnm[m][6];    nm[7]=tnm[m][8];    nm[8]=tnm[m][7];
        nm[9]=tnm[m][13]; nm[10]=tnm[m][14]; nm[11]=tnm[m][16]; nm[12]=tnm[m][15];
        nm[13]=tnm[m][9]; nm[14]=tnm[m][10]; nm[15]=tnm[m][12]; nm[16]=tnm[m][11];
        for(n=1;n<17;n++){tnm[m][n]=nm[n];}
    }
}
//
/* Print each Prototype and Object Solutions *
void pr32ans(short x, short y){
    short m,l,l4,n,p;
    short t0;
    for(m=x-1;m<y;m=m+3){
        for(n=0;n<3;n++){t0=tnm[m+n][0];
            printf("      P/%3d/C",t0);
            if(n<2){printf("      ");}
        }
        printf("\n");
        for(l=0;l<4;l++){l4=l*4;
            for(p=0;p<3;p++){
                printf(" ");
                for(n=0;n<4;n++){printf("%3d",pnm[m+p][l4+n+1]);}
                printf(" ");
                for(n=0;n<4;n++){printf("%3d",tnm[m+p][l4+n+1]);}
                if(p<2){printf(" ");}
            }
            printf("\n");
        }
    }
}
//
void solsort(void){
    int d0,d1;
    short mx,n,f;
    for(mx=tcnt-1;mx>0;mx--){
        do{f=0;
            for(n=0;n<mx;n++){
                d0=(((pnm[n][1]*16+pnm[n][2])*16+pnm[n][3])*16+pnm[n][4])*16+pnm[n][5];
                d1=(((pnm[n+1][1]*16+pnm[n+1][2])*16+pnm[n+1][3])*16+pnm[n+1][4])*16+pnm[n+1][5];
            }
        }
    }
}

```

```

        if(d0>d1){exc(n); f=1;}
        if((d0==d1)&&(pnm[n][6]>pnm[n+1][6])){exc(n); f=1;}
    }
}while(f>0);
}
}
//
void exc(short x){
    short n,d;
    for(n=1;n<17;n++){d=pnm[x][n]; pnm[x][n]=pnm[x+1][n]; pnm[x+1][n]=d;}
}
//
/* Make CCMS44 by our Old Ordinary Method *
/* Set n1 & n11=PSM-n1 *
void stp01(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[1]=a; nm[11]=b;
            uflg[a]=1; uflg[b]=1;
            stp02();
            uflg[b]=0; uflg[a]=0;}}
}
}
/* Set n2 & n12=PSM-n2 *
void stp02(){
    short a,b;
    for(a=16;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[2]=a; nm[12]=b;
            uflg[a]=1; uflg[b]=1;
            stp03();
            uflg[b]=0; uflg[a]=0;}}
}
}
/* Set n3 & n9=PSM-n3 *
void stp03(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[3]=a; nm[9]=b;
            uflg[a]=1; uflg[b]=1;
            stp04();
            uflg[b]=0; uflg[a]=0;}}
}
}
/* Set n4=LSM-n1-n2-n3 & n10=PSM-n4 *
void stp04(){
    short a,b;
    a=LSM-nm[1]-nm[2]-nm[3]; b=PSM-a;
    if((0<a)&&(a<17)){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[4]=a; nm[10]=b;
            uflg[a]=1; uflg[b]=1;
            stp05();
            uflg[b]=0; uflg[a]=0;}}
}
}
/* Set n5 & n15=PSM-n5 *
void stp05(){
    short a,b;
    for(a=16;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[5]=a; nm[15]=b;
            uflg[a]=1; uflg[b]=1;
            stp06();

```

```

        uflg[b]=0; uflg[a]=0;}
    }
}
/** Set n13=LSM-n1-n5-n9 & n7=PSM-n13 *
void stp06(){
short a,b;
a=LSM-nm[1]-nm[5]-nm[9]; b=PSM-a;
if((0<a)&&(a<17)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[13]=a; nm[7]=b;
        uflg[a]=1; uflg[b]=1;
        stp07();
        uflg[b]=0; uflg[a]=0;}}
}
/** Set n6 & n16=PSM-n6 *
void stp07(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[6]=a; nm[16]=b;
        uflg[a]=1; uflg[b]=1;
        stp08();
        uflg[b]=0; uflg[a]=0;}
}
}
/** Set n8=LSM-n5-n6-n7 & n14=PSM-n8 *
void stp08(){
short a,b,c;
a=LSM-nm[5]-nm[6]-nm[7]; b=PSM-a;
c=LSM-nm[2]-nm[6]-nm[10];
if((0<a)&&(a<17)&&(b==c)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[8]=a; nm[14]=b;
        uflg[a]=1; uflg[b]=1;
        recsol();
        uflg[b]=0; uflg[a]=0;}}
}
/** Record the Answers to the Table *
void recsol(){
short n;
snm[cnt][0]=cnt+1;
for(n=1;n<17;n++){snm[cnt][n]=nm[n];}
cnt++;
}
//
/** Find the Corresponding Number in the Old List *
void chkmon(void){
short m,n,p;
short mtc=0;
for(m=0;m<tcnt;m++){
    for(n=0;n<cnt;n++){
        mtc=0;
        for(p=1;p<17;p++){
            if(tnm[m][p]==snm[n][p]){mtc++;}else{break;}}
        if(mtc==16){
            tnm[m][0]=snm[n][0];
            onmb[n+1]++; break;}
    }
    if(mtc<16){onmb[0]++;}
}
}
//
/** Print the Monitor List of Correspondence *
void prmon(void){
short n;

```

```

printf(" ??:%3d",onmb[0]);
for(n=1;n<=cnt;n++){
  if(n%24==1){printf("\n%4d:",n);}
  printf("%3d",onmb[n]);
}
printf("\n");
}
//

```

実行結果を示す。普通の方法で求めた解と比較したモニター結果も最後に添えた。

**** List of 384 Pairs of Prototype Squares and Complete MS44 Composed ****

P/ 1/C				P/ 2/C				P/ 3/C			
1 2 3 4	1 15 4 14	1 2 3 4	1 15 4 14	1 2 5 6	1 15 6 12						
5 6 7 8	12 6 9 7	9 10 11 12	8 10 5 11	3 4 7 8	14 4 9 7						
9 10 11 12	13 3 16 2	5 6 7 8	13 3 16 2	9 10 13 14	11 5 16 2						
13 14 15 16	8 10 5 11	13 14 15 16	12 6 9 7	11 12 15 16	8 10 3 13						
P/ 4/C				P/ 5/C				P/ 6/C			
1 2 5 6	1 15 6 12	1 2 9 10	1 15 10 8	1 2 9 10	1 15 10 8						
9 10 13 14	8 10 3 13	3 4 11 12	14 4 5 11	5 6 13 14	12 6 3 13						
3 4 7 8	11 5 16 2	5 6 13 14	7 9 16 2	3 4 11 12	7 9 16 2						
11 12 15 16	14 4 9 7	7 8 15 16	12 6 3 13	7 8 15 16	14 4 5 11						
P/ 7/C				P/ 8/C				P/ 9/C			
1 3 2 4	1 14 4 15	1 3 2 4	1 14 4 15	1 3 5 7	1 14 7 12						
5 7 6 8	12 7 9 6	9 11 10 12	8 11 5 10	2 4 6 8	15 4 9 6						
9 11 10 12	13 2 16 3	5 7 6 8	13 2 16 3	9 11 13 15	10 5 16 3						
13 15 14 16	8 11 5 10	13 15 14 16	12 7 9 6	10 12 14 16	8 11 2 13						
P/ 10/C				P/ 11/C				P/ 12/C			
1 3 5 7	1 14 7 12	1 3 9 11	1 14 11 8	1 3 9 11	1 14 11 8						
9 11 13 15	8 11 2 13	2 4 10 12	15 4 5 10	5 7 13 15	12 7 2 13						
2 4 6 8	10 5 16 3	5 7 13 15	6 9 16 3	2 4 10 12	6 9 16 3						
10 12 14 16	15 4 9 6	6 8 14 16	12 7 2 13	6 8 14 16	15 4 5 10						
P/ 13/C				P/ 14/C				P/ 15/C			
1 5 2 6	1 12 6 15	1 5 2 6	1 12 6 15	1 5 3 7	1 12 7 14						
3 7 4 8	14 7 9 4	9 13 10 14	8 13 3 10	2 6 4 8	15 6 9 4						
9 13 10 14	11 2 16 5	3 7 4 8	11 2 16 5	9 13 11 15	10 3 16 5						
11 15 12 16	8 13 3 10	11 15 12 16	14 7 9 4	10 14 12 16	8 13 2 11						
P/ 16/C				P/ 17/C				P/ 18/C			
1 5 3 7	1 12 7 14	1 5 9 13	1 12 13 8	1 5 9 13	1 12 13 8						
9 13 11 15	8 13 2 11	2 6 10 14	15 6 3 10	3 7 11 15	14 7 2 11						
2 6 4 8	10 3 16 5	3 7 11 15	4 9 16 5	2 6 10 14	4 9 16 5						
10 14 12 16	15 6 9 4	4 8 12 16	14 7 2 11	4 8 12 16	15 6 3 10						
P/ 19/C				P/ 20/C				P/ 21/C			
1 9 2 10	1 8 10 15	1 9 2 10	1 8 10 15	1 9 3 11	1 8 11 14						
3 11 4 12	14 11 5 4	5 13 6 14	12 13 3 6	2 10 4 12	15 10 5 4						
5 13 6 14	7 2 16 9	3 11 4 12	7 2 16 9	5 13 7 15	6 3 16 9						
7 15 8 16	12 13 3 6	7 15 8 16	14 11 5 4	6 14 8 16	12 13 2 7						
P/ 22/C				P/ 23/C				P/ 24/C			
1 9 3 11	1 8 11 14	1 9 5 13	1 8 13 12	1 9 5 13	1 8 13 12						
5 13 7 15	12 13 2 7	2 10 6 14	15 10 3 6	3 11 7 15	14 11 2 7						
2 10 4 12	6 3 16 9	3 11 7 15	4 5 16 9	2 10 6 14	4 5 16 9						
6 14 8 16	15 10 5 4	4 12 8 16	14 11 2 7	4 12 8 16	15 10 3 6						
P/ 25/C				P/ 26/C				P/ 27/C			
2 1 4 3	2 16 3 13	2 1 4 3	2 16 3 13	2 1 6 5	2 16 5 11						
6 5 8 7	11 5 10 8	10 9 12 11	7 9 6 12	4 3 8 7	13 3 10 8						
10 9 12 11	14 4 15 1	6 5 8 7	14 4 15 1	10 9 14 13	12 6 15 1						
14 13 16 15	7 9 6 12	14 13 16 15	11 5 10 8	12 11 16 15	7 9 4 14						

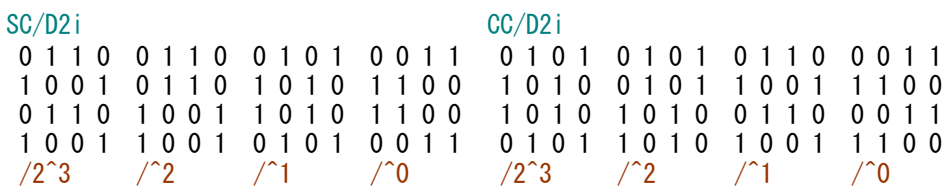
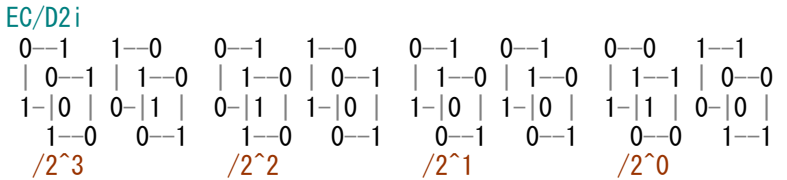
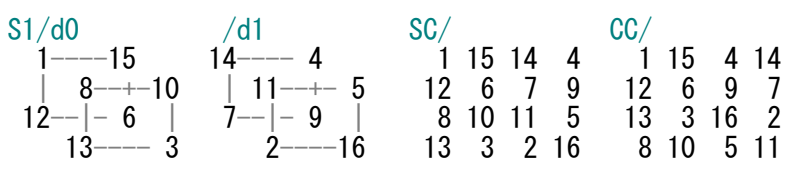
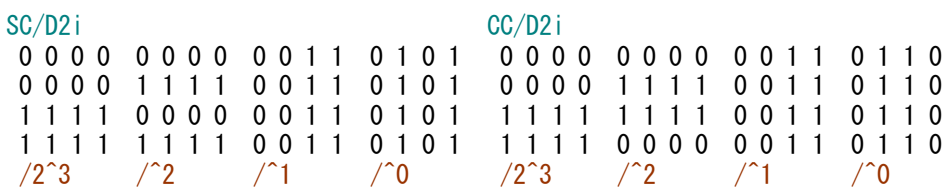
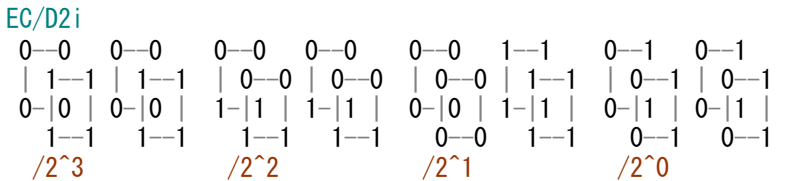
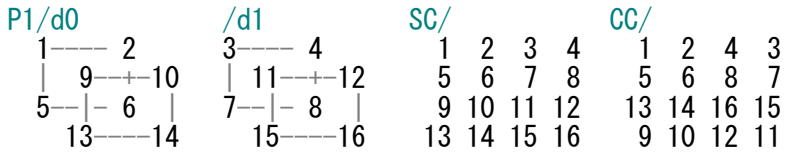
P/ 28/C				P/ 29/C				P/ 30/C															
2	1	6	5	2	16	5	11	2	1	10	9	2	16	9	7	2	1	10	9	2	16	9	7
10	9	14	13	7	9	4	14	4	3	12	11	13	3	6	12	6	5	14	13	11	5	4	14
4	3	8	7	12	6	15	1	6	5	14	13	8	10	15	1	4	3	12	11	8	10	15	1
12	11	16	15	13	3	10	8	8	7	16	15	11	5	4	14	8	7	16	15	13	3	6	12
P/ 31/C				P/ 32/C				P/ 33/C															
2	4	1	3	2	13	3	16	2	4	1	3	2	13	3	16	2	4	6	8	2	13	8	11
6	8	5	7	11	8	10	5	10	12	9	11	7	12	6	9	1	3	5	7	16	3	10	5
10	12	9	11	14	1	15	4	6	8	5	7	14	1	15	4	10	12	14	16	9	6	15	4
14	16	13	15	7	12	6	9	14	16	13	15	11	8	10	5	9	11	13	15	7	12	1	14
P/ 34/C				P/ 35/C				P/ 36/C															
2	4	6	8	2	13	8	11	2	4	10	12	2	13	12	7	2	4	10	12	2	13	12	7
10	12	14	16	7	12	1	14	1	3	9	11	16	3	6	9	6	8	14	16	11	8	1	14
1	3	5	7	9	6	15	4	6	8	14	16	5	10	15	4	1	3	9	11	5	10	15	4
9	11	13	15	16	3	10	5	5	7	13	15	11	8	1	14	5	7	13	15	16	3	6	9
P/ 37/C				P/ 38/C				P/ 39/C															
2	6	1	5	2	11	5	16	2	6	1	5	2	11	5	16	2	6	4	8	2	11	8	13
4	8	3	7	13	8	10	3	10	14	9	13	7	14	4	9	1	5	3	7	16	5	10	3
10	14	9	13	12	1	15	6	4	8	3	7	12	1	15	6	10	14	12	16	9	4	15	6
12	16	11	15	7	14	4	9	12	16	11	15	13	8	10	3	9	13	11	15	7	14	1	12
P/ 40/C				P/ 41/C				P/ 42/C															
2	6	4	8	2	11	8	13	2	6	10	14	2	11	14	7	2	6	10	14	2	11	14	7
10	14	12	16	7	14	1	12	1	5	9	13	16	5	4	9	4	8	12	16	13	8	1	12
1	5	3	7	9	4	15	6	4	8	12	16	3	10	15	6	1	5	9	13	3	10	15	6
9	13	11	15	16	5	10	3	3	7	11	15	13	8	1	12	3	7	11	15	16	5	4	9
P/ 43/C				P/ 44/C				P/ 45/C															
2	10	1	9	2	7	9	16	2	10	1	9	2	7	9	16	2	10	4	12	2	7	12	13
4	12	3	11	13	12	6	3	6	14	5	13	11	14	4	5	1	9	3	11	16	9	6	3
6	14	5	13	8	1	15	10	4	12	3	11	8	1	15	10	6	14	8	16	5	4	15	10
8	16	7	15	11	14	4	5	8	16	7	15	13	12	6	3	5	13	7	15	11	14	1	8
P/ 46/C				P/ 47/C				P/ 48/C															
2	10	4	12	2	7	12	13	2	10	6	14	2	7	14	11	2	10	6	14	2	7	14	11
6	14	8	16	11	14	1	8	1	9	5	13	16	9	4	5	4	12	8	16	13	12	1	8
1	9	3	11	5	4	15	10	4	12	8	16	3	6	15	10	1	9	5	13	3	6	15	10
5	13	7	15	16	9	6	3	3	11	7	15	13	12	1	8	3	11	7	15	16	9	4	5
P/ 49/C				P/ 50/C				P/ 51/C															
3	1	4	2	3	16	2	13	3	1	4	2	3	16	2	13	3	1	7	5	3	16	5	10
7	5	8	6	10	5	11	8	11	9	12	10	6	9	7	12	4	2	8	6	13	2	11	8
11	9	12	10	15	4	14	1	7	5	8	6	15	4	14	1	11	9	15	13	12	7	14	1
15	13	16	14	6	9	7	12	15	13	16	14	10	5	11	8	12	10	16	14	6	9	4	15
P/ 52/C				P/ 53/C				P/ 54/C															
3	1	7	5	3	16	5	10	3	1	11	9	3	16	9	6	3	1	11	9	3	16	9	6
11	9	15	13	6	9	4	15	4	2	12	10	13	2	7	12	7	5	15	13	10	5	4	15
4	2	8	6	12	7	14	1	7	5	15	13	8	11	14	1	4	2	12	10	8	11	14	1
12	10	16	14	13	2	11	8	8	6	16	14	10	5	4	15	8	6	16	14	13	2	7	12
P/ 70/C				P/ 71/C				P/ 72/C															
3	11	4	12	3	6	12	13	3	11	7	15	3	6	15	10	3	11	7	15	3	6	15	10
7	15	8	16	10	15	1	8	1	9	5	13	16	9	4	5	4	12	8	16	13	12	1	8
1	9	2	10	5	4	14	11	4	12	8	16	2	7	14	11	1	9	5	13	2	7	14	11
5	13	6	14	16	9	7	2	2	10	6	14	13	12	1	8	2	10	6	14	16	9	4	5
P/ 73/C				P/ 74/C				P/ 75/C															
4	2	3	1	4	15	1	14	4	2	3	1	4	15	1	14	4	2	8	6	4	15	6	9
8	6	7	5	9	6	12	7	12	10	11	9	5	10	8	11	3	1	7	5	14	1	12	7
12	10	11	9	16	3	13	2	8	6	7	5	16	3	13	2	12	10	16	14	11	8	13	2
16	14	15	13	5	10	8	11	16	14	15	13	9	6	12	7	11	9	15	13	5	10	3	16

<p>P/ 76/C</p> <p>4 2 8 6 4 15 6 9 12 10 16 14 5 10 3 16 3 1 7 5 11 8 13 2 11 9 15 13 14 1 12 7</p>	<p>P/ 77/C</p> <p>4 2 12 10 4 15 10 5 3 1 11 9 14 1 8 11 8 6 16 14 7 12 13 2 7 5 15 13 9 6 3 16</p>	<p>P/ 78/C</p> <p>4 2 12 10 4 15 10 5 8 6 16 14 9 6 3 16 3 1 11 9 7 12 13 2 7 5 15 13 14 1 8 11</p>
.		
<p>P/ 94/C</p> <p>4 12 3 11 4 5 11 14 8 16 7 15 9 16 2 7 2 10 1 9 6 3 13 12 6 14 5 13 15 10 8 1</p>	<p>P/ 95/C</p> <p>4 12 8 16 4 5 16 9 2 10 6 14 15 10 3 6 3 11 7 15 1 8 13 12 1 9 5 13 14 11 2 7</p>	<p>P/ 96/C</p> <p>4 12 8 16 4 5 16 9 3 11 7 15 14 11 2 7 2 10 6 14 1 8 13 12 1 9 5 13 15 10 3 6</p>
<p>P/ 97/C</p> <p>5 1 6 2 5 16 2 11 7 3 8 4 10 3 13 8 13 9 14 10 15 6 12 1 15 11 16 12 4 9 7 14</p>	<p>P/ 98/C</p> <p>5 1 6 2 5 16 2 11 13 9 14 10 4 9 7 14 7 3 8 4 15 6 12 1 15 11 16 12 10 3 13 8</p>	<p>P/ 99/C</p> <p>5 1 7 3 5 16 3 10 6 2 8 4 11 2 13 8 13 9 15 11 14 7 12 1 14 10 16 12 4 9 6 15</p>
<p>P/100/C</p> <p>5 1 7 3 5 16 3 10 13 9 15 11 4 9 6 15 6 2 8 4 14 7 12 1 14 10 16 12 11 2 13 8</p>	<p>P/101/C</p> <p>5 1 13 9 5 16 9 4 6 2 14 10 11 2 7 14 7 3 15 11 8 13 12 1 8 4 16 12 10 3 6 15</p>	<p>P/102/C</p> <p>5 1 13 9 5 16 9 4 7 3 15 11 10 3 6 15 6 2 14 10 8 13 12 1 8 4 16 12 11 2 7 14</p>
.		
<p>P/190/C</p> <p>8 16 6 14 8 1 14 11 7 15 5 13 10 15 4 5 4 12 2 10 3 6 9 16 3 11 1 9 13 12 7 2</p>	<p>P/191/C</p> <p>8 16 7 15 8 1 15 10 4 12 3 11 13 12 6 3 6 14 5 13 2 7 9 16 2 10 1 9 11 14 4 5</p>	<p>P/192/C</p> <p>8 16 7 15 8 1 15 10 6 14 5 13 11 14 4 5 4 12 3 11 2 7 9 16 2 10 1 9 13 12 6 3</p>
<p>P/193/C</p> <p>9 1 10 2 9 16 2 7 11 3 12 4 6 3 13 12 13 5 14 6 15 10 8 1 15 7 16 8 4 5 11 14</p>	<p>P/194/C</p> <p>9 1 10 2 9 16 2 7 13 5 14 6 4 5 11 14 11 3 12 4 15 10 8 1 15 7 16 8 6 3 13 12</p>	<p>P/195/C</p> <p>9 1 11 3 9 16 3 6 10 2 12 4 7 2 13 12 13 5 15 7 14 11 8 1 14 6 16 8 4 5 10 15</p>
<p>P/196/C</p> <p>9 1 11 3 9 16 3 6 13 5 15 7 4 5 10 15 10 2 12 4 14 11 8 1 14 6 16 8 7 2 13 12</p>	<p>P/197/C</p> <p>9 1 13 5 9 16 5 4 10 2 14 6 7 2 11 14 11 3 15 7 12 13 8 1 12 4 16 8 6 3 10 15</p>	<p>P/198/C</p> <p>9 1 13 5 9 16 5 4 11 3 15 7 6 3 10 15 10 2 14 6 12 13 8 1 12 4 16 8 7 2 11 14</p>
.		
<p>P/376/C</p> <p>16 14 12 10 16 3 10 5 15 13 11 9 2 13 8 11 8 6 4 2 7 12 1 14 7 5 3 1 9 6 15 4</p>	<p>P/377/C</p> <p>16 14 15 13 16 3 13 2 8 6 7 5 9 6 12 7 12 10 11 9 4 15 1 14 4 2 3 1 5 10 8 11</p>	<p>P/378/C</p> <p>16 14 15 13 16 3 13 2 12 10 11 9 5 10 8 11 8 6 7 5 4 15 1 14 4 2 3 1 9 6 12 7</p>
<p>P/379/C</p> <p>16 15 8 7 16 2 7 9 12 11 4 3 5 11 14 4 14 13 6 5 10 8 1 15 10 9 2 1 3 13 12 6</p>	<p>P/380/C</p> <p>16 15 8 7 16 2 7 9 14 13 6 5 3 13 12 6 12 11 4 3 10 8 1 15 10 9 2 1 5 11 14 4</p>	<p>P/381/C</p> <p>16 15 12 11 16 2 11 5 8 7 4 3 9 7 14 4 14 13 10 9 6 12 1 15 6 5 2 1 3 13 8 10</p>
<p>P/382/C</p> <p>16 15 12 11 16 2 11 5 14 13 10 9 3 13 8 10 8 7 4 3 6 12 1 15 6 5 2 1 9 7 14 4</p>	<p>P/383/C</p> <p>16 15 14 13 16 2 13 3 8 7 6 5 9 7 12 6 12 11 10 9 4 14 1 15 4 3 2 1 5 11 8 10</p>	<p>P/384/C</p> <p>16 15 14 13 16 2 13 3 12 11 10 9 5 11 8 10 8 7 6 5 4 14 1 15 4 3 2 1 9 7 12 6</p>

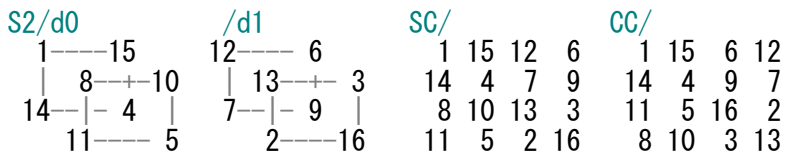
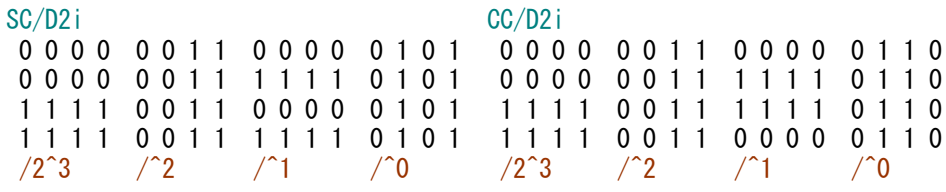
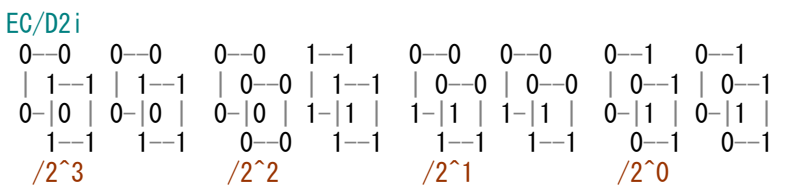
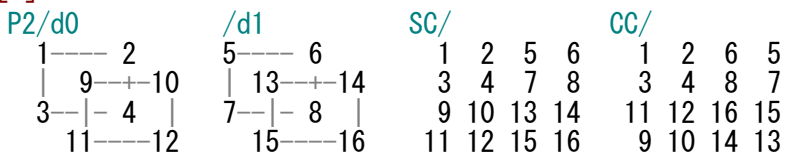
[Count (Proto/Comp) = 384/384] OK!

** 普通の方法で求めた完全四方陣の解と比較して対応関係を調べたモニター結果 **

[1]



[2]



EC/D2i

0-1	1-0	0-1	0-1	0-1	1-0	0-0	1-1
0-1 1-0 1-0 1-0 1-0 0-1 1-1 0-0							
1-0 0-1 1-0 1-0 0-1 1-0 1-1 0-0							
1-0	0-1	0-1	0-1	1-0	0-1	0-0	1-1
/2^3	/2^2	/2^1	/2^0				

SC/D2i

0 1 1 0	0 1 0 1	0 1 1 0	0 0 1 1	0 1 0 1	0 1 1 0	0 1 0 1	0 0 1 1
1 0 0 1	1 0 1 0	0 1 1 0	1 1 0 0	1 0 1 0	1 0 0 1	0 1 0 1	1 1 0 0
0 1 1 0	1 0 1 0	1 0 0 1	1 1 0 0	1 0 1 0	0 1 1 0	1 0 1 0	0 0 1 1
1 0 0 1	0 1 0 1	1 0 0 1	0 0 1 1	0 1 0 1	1 0 0 1	1 0 1 0	1 1 0 0
/2^3	/^2	/^1	/^0	/2^3	/^2	/^1	/^0

CC/D2i

[3]

P3/d0

1-2	9-10	1 2 9 10	1 2 10 9
5-6 13-14	3 4 11 12	3 4 12 11	
3-4	11-12	5 6 13 14	7 8 16 15
7-8	15-16	7 8 15 16	5 6 14 13

/d1

SC/

CC/

EC/D2i

0-0	1-1	0-0	0-0	0-0	0-0	0-1	0-1
0-0 1-1 1-1 1-1 0-0 0-0 0-1 0-1							
0-0 1-1 0-0 0-0 1-1 1-1 0-1 0-1							
0-0	1-1	1-1	1-1	1-1	1-1	0-1	0-1
/2^3	/2^2	/2^1	/2^0				

SC/D2i

0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	0 0 1 1	0 0 0 0	0 0 0 0	0 1 1 0
0 0 1 1	0 0 0 0	1 1 1 1	0 1 0 1	0 0 1 1	0 0 0 0	1 1 1 1	0 1 1 0
0 0 1 1	1 1 1 1	0 0 0 0	0 1 0 1	0 0 1 1	1 1 1 1	1 1 1 1	0 1 1 0
0 0 1 1	1 1 1 1	1 1 1 1	0 1 0 1	0 0 1 1	1 1 1 1	0 0 0 0	0 1 1 0
/2^3	/^2	/^1	/^0	/2^3	/^2	/^1	/^0

CC/D2i

S3/d0

1-15	8-10	1 15 8 10	1 15 10 8
12-6 13-3	14 4 11 5	14 4 5 11	
14-4	11-5	12 6 13 3	7 9 16 2
7-9	2-16	7 9 2 16	12 6 3 13

/d1

SC/

CC/

EC/D2i

0-1	0-1	0-1	1-0	0-1	1-0	0-0	1-1
1-0 1-0 0-1 1-0 1-0 0-1 1-1 0-0							
1-0 1-0 1-0 0-1 0-1 1-0 1-1 0-0							
0-1	0-1	1-0	0-1	1-0	0-1	0-0	1-1
/2^3	/2^2	/2^1	/2^0				

SC/D2i

0 1 0 1	0 1 1 0	0 1 1 0	0 0 1 1	0 1 1 0	0 1 0 1	0 1 0 1	0 0 1 1
1 0 1 0	1 0 0 1	0 1 1 0	1 1 0 0	1 0 0 1	1 0 1 0	0 1 0 1	1 1 0 0
1 0 1 0	0 1 1 0	1 0 0 1	1 1 0 0	0 1 1 0	1 0 1 0	1 0 1 0	0 0 1 1
0 1 0 1	1 0 0 1	1 0 0 1	0 0 1 1	1 0 0 1	0 1 0 1	1 0 1 0	1 1 0 0
/2^3	/^2	/^1	/^0	/2^3	/^2	/^1	/^0

CC/D2i

上の表は、何を意味しているか。結果から見て次のことが言えるように思う。

(1) 構成解は、いずれも「完全オイラー方陣」である。先ず小立方体の全てが、同数の {0, 1} で構成されている。これならば、重要部分の四数定和が成り立つわけだ。

変換後の二次元四方陣も、各行各列各汎対角線が {0, 0, 1, 1} で構成されている。

(2) 我々は、二進数発生プログラムを用いることによって、実は「完全オイラー方陣」を作る準備をしていたことになる。完全オイラー四方陣ならば、原始解は 384 個、標準解が 48 個出来るではないか。

(3) 「二進法で作る」という簡単な作業にそんなにも凄い力がはたしてあるのかと思うが、 $4P_4 \times 2^4 = 24 \times 16 = 384$ がそれを指し示しているように思う。単に数の表記法にとどまらない、言わば「構造を作り出す能力」が、例の 4 重ループにはあるようだ。

(4) 「展開見取り図」である基本図を多数作りながら、我々は実は「原初方陣」を作っていた。だから1個の「ひな形」を用いる変換法で、解の全数個を構成し得たのである。

かくして超立体魔方陣と原初方陣とN進法と完全オイラー方陣が、一筋につながった。今までやって来た諸研究の根源的な意義は、ここにあったのである。それを見つけ出すことができたことが、自分としては一番嬉しい。

6. その後の研究成果を付記する。

実は、小生にはまだ次のような疑問が残っていた。

最適な「ひな形」は、はたして1つに限られるものだろうか。

384個の基本形の集合からひな形変換法によって目的方陣の解384個の集合を構成する際に、我々は仮に1つの「ひな形」を選定し、それから変換法を考え出していた。

再録すると、

BD/d0	/d1	SC/基本形
1--- 2	3--- 4	1 2 3 4
9---10	11---12	5 6 7 8
5-- - 6	7-- - 8	9 10 11 12
13---14	15---16	13 14 15 16
MD/d0	/d1	SC/ひな形 ←
1---15	14--- 4	1 15 14 4
8---10	11--- 5	12 6 7 9
12-- - 6	7-- - 9	8 10 11 5
13--- 3	2---16	13 3 2 16

//* ひな形変換法 *

CC=17;

y[1]=x[1]; y[2]=CC-x[2]; y[3]=CC-x[3]; y[4]=x[4];

y[5]=CC-x[5]; y[6]=x[6]; y[7]=x[7]; y[8]=CC-x[8];

y[9]=CC-x[9]; y[10]=x[10]; y[11]=x[11]; y[12]=CC-x[12];

y[13]=x[13]; y[14]=CC-x[14]; y[15]=CC-x[15]; y[16]=x[16];

//

これは、小生のお気に入りのひな形と変換法なのだが、これしか無いのであろうか？以前の原初四方陣の研究の時に、大基本形3個の内の他の2解も適格性を備えていることがわかっていたが、それら3つ以外には無いのであろうか。それが気になっていた。試みに次の解を「ひな形」に選び、下図のような変換法を考えて実行してみよう。

1/P	8/T	ひな形		* ひな形変換法 *
1 2 3 4	1 14 15 4			y[1]=x[1]; y[2]=x[14]; y[3]=x[15]; y[4]=x[4];
5 6 7 8	8 11 10 5			y[5]=x[8]; y[6]=x[11]; y[7]=x[10]; y[8]=x[5];
9 10 11 12	12 7 6 9			y[9]=x[12]; y[10]=x[7]; y[11]=x[6]; y[12]=x[9];
13 14 15 16	13 2 3 16			y[13]=x[13]; y[14]=x[2]; y[15]=x[3]; y[16]=x[16];

実行結果は、次の通りである。基本形の集合は、順番以外に少しも変えていない。何とこのひな形でも、全数の目的方陣が構成できたのである。これには驚いた。他のひな形でも試してみたが、n1=1のタイプの24解の全てが平等に「ひな形」になり得ることがわかった。

* ひな形とひな形変換法を変えて対称四方陣を構成した結果 *

1/P	8/T	2/P	7/T	3/P	14/T
1 2 3 4	1 14 15 4	1 2 3 4	1 14 15 4	1 2 5 6	1 12 15 6
5 6 7 8	8 11 10 5	9 10 11 12	12 7 6 9	3 4 7 8	8 13 10 3
9 10 11 12	12 7 6 9	5 6 7 8	8 11 10 5	9 10 13 14	14 7 4 9
13 14 15 16	13 2 3 16	13 14 15 16	13 2 3 16	11 12 15 16	11 2 5 16

4/P 13/T	5/P 20/T	6/P 19/T
1 2 5 6 1 12 15 6	1 2 9 10 1 8 15 10	1 2 9 10 1 8 15 10
9 10 13 14 14 7 4 9	3 4 11 12 12 13 6 3	5 6 13 14 14 11 4 5
3 4 7 8 8 13 10 3	5 6 13 14 14 11 4 5	3 4 11 12 12 13 6 3
11 12 15 16 11 2 5 16	7 8 15 16 7 2 9 16	7 8 15 16 7 2 9 16
7/P 2/T	8/P 1/T	9/P 16/T
1 3 2 4 1 15 14 4	1 3 2 4 1 15 14 4	1 3 5 7 1 12 14 7
5 7 6 8 8 10 11 5	9 11 10 12 12 6 7 9	2 4 6 8 8 13 11 2
9 11 10 12 12 6 7 9	5 7 6 8 8 10 11 5	9 11 13 15 15 6 4 9
13 15 14 16 13 3 2 16	13 15 14 16 13 3 2 16	10 12 14 16 10 3 5 16
10/P 15/T	11/P 22/T	12/P 21/T
1 3 5 7 1 12 14 7	1 3 9 11 1 8 14 11	1 3 9 11 1 8 14 11
9 11 13 15 15 6 4 9	2 4 10 12 12 13 7 2	5 7 13 15 15 10 4 5
2 4 6 8 8 13 11 2	5 7 13 15 15 10 4 5	2 4 10 12 12 13 7 2
10 12 14 16 10 3 5 16	6 8 14 16 6 3 9 16	6 8 14 16 6 3 9 16
13/P 4/T	14/P 3/T	15/P 10/T
1 5 2 6 1 15 12 6	1 5 2 6 1 15 12 6	1 5 3 7 1 14 12 7
3 7 4 8 8 10 13 3	9 13 10 14 14 4 7 9	2 6 4 8 8 11 13 2
9 13 10 14 14 4 7 9	3 7 4 8 8 10 13 3	9 13 11 15 15 4 6 9
11 15 12 16 11 5 2 16	11 15 12 16 11 5 2 16	10 14 12 16 10 5 3 16
16/P 9/T	17/P 24/T	18/P 23/T
1 5 3 7 1 14 12 7	1 5 9 13 1 8 12 13	1 5 9 13 1 8 12 13
9 13 11 15 15 4 6 9	2 6 10 14 14 11 7 2	3 7 11 15 15 10 6 3
2 6 4 8 8 11 13 2	3 7 11 15 15 10 6 3	2 6 10 14 14 11 7 2
10 14 12 16 10 5 3 16	4 8 12 16 4 5 9 16	4 8 12 16 4 5 9 16
19/P 6/T	20/P 5/T	21/P 12/T
1 9 2 10 1 15 8 10	1 9 2 10 1 15 8 10	1 9 3 11 1 14 8 11
3 11 4 12 12 6 13 3	5 13 6 14 14 4 11 5	2 10 4 12 12 7 13 2
5 13 6 14 14 4 11 5	3 11 4 12 12 6 13 3	5 13 7 15 15 4 10 5
7 15 8 16 7 9 2 16	7 15 8 16 7 9 2 16	6 14 8 16 6 9 3 16
22/P 11/T	23/P 18/T	24/P 17/T
1 9 3 11 1 14 8 11	1 9 5 13 1 12 8 13	1 9 5 13 1 12 8 13
5 13 7 15 15 4 10 5	2 10 6 14 14 7 11 2	3 11 7 15 15 6 10 3
2 10 4 12 12 7 13 2	3 11 7 15 15 6 10 3	2 10 6 14 14 7 11 2
6 14 8 16 6 9 3 16	4 12 8 16 4 9 5 16	4 12 8 16 4 9 5 16
25/P 32/T	26/P 31/T	27/P 38/T
2 1 4 3 2 13 16 3	2 1 4 3 2 13 16 3	2 1 6 5 2 11 16 5
6 5 8 7 7 12 9 6	10 9 12 11 11 8 5 10	4 3 8 7 7 14 9 4
10 9 12 11 11 8 5 10	6 5 8 7 7 12 9 6	10 9 14 13 13 8 3 10
14 13 16 15 14 1 4 15	14 13 16 15 14 1 4 15	12 11 16 15 12 1 6 15
190/P 179/T	191/P 186/T	192/P 185/T
8 16 6 14 8 11 1 14	8 16 7 15 8 10 1 15	8 16 7 15 8 10 1 15
7 15 5 13 13 2 12 7	4 12 3 11 11 5 14 4	6 14 5 13 13 3 12 6
4 12 2 10 10 5 15 4	6 14 5 13 13 3 12 6	4 12 3 11 11 5 14 4
3 11 1 9 3 16 6 9	2 10 1 9 2 16 7 9	2 10 1 9 2 16 7 9
193/P 200/T	194/P 199/T	195/P 206/T
9 1 10 2 9 7 16 2	9 1 10 2 9 7 16 2	9 1 11 3 9 6 16 3
11 3 12 4 4 14 5 11	13 5 14 6 6 12 3 13	10 2 12 4 4 15 5 10
13 5 14 6 6 12 3 13	11 3 12 4 4 14 5 11	13 5 15 7 7 12 2 13
15 7 16 8 15 1 10 8	15 7 16 8 15 1 10 8	14 6 16 8 14 1 11 8
196/P 205/T	197/P 212/T	198/P 211/T
9 1 11 3 9 6 16 3	9 1 13 5 9 4 16 5	9 1 13 5 9 4 16 5
13 5 15 7 7 12 2 13	10 2 14 6 6 15 3 10	11 3 15 7 7 14 2 11

10 2 12 4 4 15 5 10 11 3 15 7 7 14 2 11 10 2 14 6 6 15 3 10
 14 6 16 8 14 1 11 8 12 4 16 8 12 1 13 8 12 4 16 8 12 1 13 8

376/P	369/T	377/P	384/T	378/P	383/T
16 14 12 10	16 5 3 10	16 14 15 13	16 2 3 13	16 14 15 13	16 2 3 13
15 13 11 9	9 4 6 15	8 6 7 5	5 11 10 8	12 10 11 9	9 7 6 12
8 6 4 2	2 11 13 8	12 10 11 9	9 7 6 12	8 6 7 5	5 11 10 8
7 5 3 1	7 14 12 1	4 2 3 1	4 14 15 1	4 2 3 1	4 14 15 1
379/P	366/T	380/P	365/T	381/P	372/T
16 15 8 7	16 9 2 7	16 15 8 7	16 9 2 7	16 15 12 11	16 5 2 11
12 11 4 3	3 6 13 12	14 13 6 5	5 4 11 14	8 7 4 3	3 10 13 8
14 13 6 5	5 4 11 14	12 11 4 3	3 6 13 12	14 13 10 9	9 4 7 14
10 9 2 1	10 15 8 1	10 9 2 1	10 15 8 1	6 5 2 1	6 15 12 1
382/P	371/T	383/P	378/T	384/P	377/T
16 15 12 11	16 5 2 11	16 15 14 13	16 3 2 13	16 15 14 13	16 3 2 13
14 13 10 9	9 4 7 14	8 7 6 5	5 10 11 8	12 11 10 9	9 6 7 12
8 7 4 3	3 10 13 8	12 11 10 9	9 6 7 12	8 7 6 5	5 10 11 8
6 5 2 1	6 15 12 1	4 3 2 1	4 15 14 1	4 3 2 1	4 15 14 1

* 新旧の解のリストを対応させたモニター結果 *

??: 0
 1: 1
 25: 1
 49: 1
 73: 1
 97: 1
 121: 1
 145: 1
 169: 1
 193: 1
 217: 1
 241: 1
 265: 1
 289: 1
 313: 1
 337: 1
 361: 1

[Count = 384/384] OK!

英国のオルレンショウ&ブラーエ両氏は、最良のひな形は1つであると言っていたように思うが、事実は違う。或いは彼らが構造的に特別な注文を付けていたせいかもしれない。目的を対称四方陣の原始解 384 個を構成するという一点に絞る限りは、ひな形にどれを選んでも良い。ひな形変換法の「可逆性」をあえてチェックする必要も無かったのである。

かくして原初四方陣と対称四方陣の間に、何本もの橋が架かることがわかった。全体として美しい関係である。このことは、原初四方陣と対称四方陣自体が持っている「対称性」の高さをまさに示しているように思われる。

それともう一つ。既にお察しの通り、小生は原初方陣の集合と基本図形の集合とを同一視している。そのことに何の問題も無いと、はたして思っていたのだろうか。

基本図の概念を拡張している。基本図形を1個に限らず、変形図まで含めて大きくまとめて考えている。その集合は原初方陣のそれと変わらず、もはや相互に区別が付かない。

その内訳の 384 個もの異形は、とりもなおさず「内部変換法」の仕様を示す概念図にな

っている。そう考えると、たった1個のひな形で全数の目的方陣を構成してしまう、あの魔法のような「ひな形変換法」が何故可能なのかの説明が付く。

小生は、今そう考えている。

別々の二者をあえて同じものと考えたと、お互いの知識や得意技のあれこれを相互に融通し合えるようになる。その絶大な効用を一度体感してしまうと、もはや元には戻れない気がする。何しろ、そのおかげで「ひな形変換法」がここでも使えたのだから。

たとえ各々の出自が違っていても、働きや規模が全く同じものは、同じカテゴリーに入れる。あえてそれらを同一視するのである。そしてそれを活用しながら、論を次に進める。

これは、なかなか賢い方法だ。生産的な方法でもある。

そういう抽象・統括能力こそが、元々**数学**の特長であり特技だったはずではないか。

(著者：撰田 寛二(Kanji Setsuda) ; jag12001@nifty.com;

初稿：執筆・計数・作表 on April 30, 2005 with MacOSX & Xcode 1.5;

改定稿：執筆・計数・作表 on January 26, 2006 with MacOSX & Xcode 2.1;

再計算清書稿： on March 5, 2015 with MacOSX 10.10.2 & Xcode 6.1.1)
