

# 第3章：五方陣の最新研究： 摂田 寛二

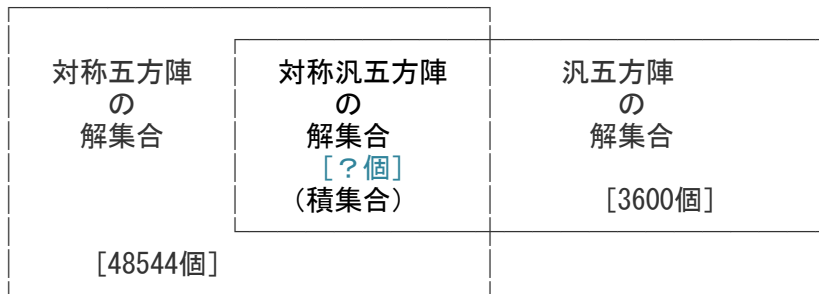
## 第3節： 連立型対称汎五方陣の最新研究

### 0. 連立型の「対称汎五方陣」とは

汎五方陣に対称性を付加したものを求める。あるいは対称五方陣に汎性を付加したものを研究する。つまり、対称性と汎性の2つを兼ね備えた5次の魔方陣を追求するのである。

集合論の言葉を使って言えば、対称五方陣の解集合と汎五方陣の解集合との間の「論理的積集合」を求めるのである。

\*\*\* 論理的積集合を求める \*\*\*



その対称汎五方陣を直接に求めるには、基本条件をどう与えたら良いだろうか。次の図で示された変数の配置と、21本の連立方程式を立てる。そして、連続する自然数の1~25を全て重複無く欠落無く各1度ずつ用いて、与条件を満たすものとする。

### 1. 対称汎五方陣の基本図



### 2. 連立方程式

行列の定和式：

$$\begin{aligned} n_1 + n_2 + n_3 + n_4 + n_5 &= K \quad \dots (1) \\ n_6 + n_7 + n_8 + n_9 + n_{10} &= K \quad \dots (2) \\ n_{11} + n_{12} + n_{13} + n_{14} + n_{15} &= K \quad \dots (3) \\ n_{16} + n_{17} + n_{18} + n_{19} + n_{20} &= K \quad \dots (4) \\ n_{21} + n_{22} + n_{23} + n_{24} + n_{25} &= K \quad \dots (5) \\ n_1 + n_6 + n_{11} + n_{16} + n_{21} &= K \quad \dots (6) \\ n_2 + n_7 + n_{12} + n_{17} + n_{22} &= K \quad \dots (7) \\ n_3 + n_8 + n_{13} + n_{18} + n_{23} &= K \quad \dots (8) \\ n_4 + n_9 + n_{14} + n_{19} + n_{24} &= K \quad \dots (9) \\ n_5 + n_{10} + n_{15} + n_{20} + n_{25} &= K \quad \dots (10) \end{aligned}$$

汎対角線の定和式：

$$\begin{aligned} n_1 + n_7 + n_{13} + n_{19} + n_{25} &= K \quad \dots (11) \\ n_1 + n_{10} + n_{14} + n_{18} + n_{22} &= K \quad \dots (12) \\ n_2 + n_8 + n_{14} + n_{20} + n_{21} &= K \quad \dots (13) \\ n_2 + n_6 + n_{15} + n_{19} + n_{23} &= K \quad \dots (14) \\ n_3 + n_7 + n_{11} + n_{20} + n_{24} &= K \quad \dots (16) \\ n_4 + n_8 + n_{12} + n_{16} + n_{25} &= K \quad \dots (18) \\ n_5 + n_9 + n_{13} + n_{17} + n_{21} &= K \quad \dots (20) \end{aligned}$$

対称的に配置された補数の定和式：

$$\begin{aligned} n_1 + n_{25} = n_2 + n_{24} = n_3 + n_{23} = n_4 + n_{22} = n_5 + n_{21} = n_6 + n_{20} = n_7 + n_{19} \\ = n_8 + n_{18} = n_9 + n_{17} = n_{10} + n_{16} = n_{11} + n_{15} = n_{12} + n_{14} (=n_{13} + n_{13}) = C \quad \dots (21) \end{aligned}$$

まずは (1)+(2)+(3)+(4)+(5)  $\{n_1 + n_2 + n_3 + \dots + n_{24} + n_{25}\} = 5 * K$   
 $26 \times 25 \div 2 = 5 * K$   
 ゆえに  $K = 65 \quad \dots (22)$

### 3. 代数計算

これらの方程式群を関わらせて計算し、意味のある関係式をさがそう。先ず(21)式を次のように組み合わせる



同様に	13 * 15 * 12
$n_5 + n_{20} + n_{21} + n_{24} + n_{25} = 65$	* 19 20 16 *
$n_5 + n_{21} = 26$ から	23 24 <b>25</b> 21 22
$n_{20} + n_{24} + n_{25} = 65 - 26 = 39$	* 4 5 1 *
	8 * 10 * 7

18 * 20 * 17	また	
* 24 25 21 *	$n_1 + n_4 + n_5 + n_{10} + n_{25} = 65$	
3 4 <b>5</b> 1 2	$n_1 + n_{25} = 26$ から	
* 9 10 6 *	$n_4 + n_5 + n_{10} = 65 - 26 = 39$	
13 * 15 * 12		

まとめると,  
 $n_1 + n_2 + n_6 = n_4 + n_5 + n_{10} = n_{16} + n_{21} + n_{22} = n_{20} + n_{24} + n_{25} = 39 \dots (27)$

などという面白い関係式も見つかる。4頂点の三角連結隅3数の和が39なのである。

$n_1, n_2$  が決まれば,  $n_6 = 39 - n_1 - n_2$  とすぐに決まる。

$n_4, n_5$  が決まれば,  $n_{10} = 39 - n_4 - n_5$  とすぐに決まる。

$n_{16}, n_{21}$  が決まれば,  $n_{22} = 39 - n_{16} - n_{21}$  とすぐに決まる。

$n_{20}, n_{25}$  が決まれば,  $n_{24} = 39 - n_{20} - n_{25}$  とすぐに決まる。

これは便利だ。調べる場合の数をずいぶんと減らせそう。

例えば  $\{n_1, n_2, n_6\}$  の場合, 和が39になる自然数の組合せは, そんなに多くない。

$n_1 = 1$  の場合を調べる。 $\{1, 13, 25\}$  から始める。だが, これは不可能だ。13は  $n_{13}$  に既に使われているから。 $\{1, 19, 19\}$  も不可能だ。同じ数の重複使用は禁じられているから。

$\{1, 14, 24\}, \{1, 15, 23\}, \{1, 16, 22\}, \{1, 17, 21\}, \{1, 18, 20\},$

$\{1, 20, 18\}, \{1, 21, 17\}, \{1, 22, 16\}, \{1, 23, 15\}, \{1, 24, 14\}$  の場合だけ調べれば良い。

#### 4. 方程式が多いので, 方陣構成は比較的楽である。先ず $n_{13} = 13$ がわかっている。

次に(21)式の補数群によって, 調べる手間を半分に減らすことができる。

$n_1$  が決まれば,  $n_{25}$  も自動的に決まる ( $n_{25} = 26 - n_1$ )。

$n_2$  が決まれば,  $n_{24}$  も自動的に決まる ( $n_{24} = 26 - n_2$ )。

$n_3$  が決まれば,  $n_{23}$  も自動的に決まる ( $n_{23} = 26 - n_3$ )。

$n_{12}$  が決まれば,  $n_{14}$  も自動的に決まる ( $n_{14} = 26 - n_{12}$ )。

あとは, 代数計算でわかった新条件式をそのまま利用する。正方連結五数の定和やひし形連結五数の定和, あるいは三角連結隅3数の定和を適宜積極的に活用できる。最初の定義に無いものでも, 定義から直に導き出される内在的性質は, 定義と同様にして使えるのだ。

はたして解は幾つあるであろうか。

#### 5. 最初は, 原始解を求めよう。

単純な回転形や反転形を省略せずに, 先ず解の「母集合」のリストを作っておこう。

#### \*\* 連立型五方陣 : 補数対称型かつ汎対角線型 \*\*

##### \*\* 原始解128個の全リスト \*\*

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

5#					6#					7#					8#				
1	23	10	14	17	1	15	24	8	17	1	23	10	12	19	1	15	22	8	19
15	19	2	21	8	23	7	16	5	14	15	17	4	21	8	23	9	16	5	12
22	6	13	20	4	20	4	13	22	6	24	6	13	20	2	20	2	13	24	6
18	5	24	7	11	12	21	10	19	3	18	5	22	9	11	14	21	10	17	3
9	12	16	3	25	9	18	2	11	25	7	14	16	3	25	7	18	4	11	25
9#					10#					11#					12#				
2	23	19	15	6	2	14	25	18	6	2	23	19	11	10	2	14	21	18	10
14	10	1	22	18	23	16	7	4	15	14	6	5	22	18	23	20	7	4	11
21	17	13	9	5	9	5	13	21	17	25	17	13	9	1	9	1	13	25	17
8	4	25	16	12	11	22	19	10	3	8	4	21	20	12	15	22	19	6	3
20	11	7	3	24	20	8	1	12	24	16	15	7	3	24	16	8	5	12	24
13#					14#					15#					16#				
2	23	9	15	16	2	14	25	8	16	2	23	9	11	20	2	14	21	8	20
14	20	1	22	8	23	6	17	4	15	14	16	5	22	8	23	10	17	4	11
21	7	13	19	5	19	5	13	21	7	25	7	13	19	1	19	1	13	25	7
18	4	25	6	12	11	22	9	20	3	18	4	21	10	12	15	22	9	16	3
10	11	17	3	24	10	18	1	12	24	6	15	17	3	24	6	18	5	12	24
17#					18#					19#					20#				
4	23	17	15	6	4	12	25	18	6	4	23	17	11	10	4	12	21	18	10
12	10	1	24	18	23	16	9	2	15	12	6	5	24	18	23	20	9	2	11
21	19	13	7	5	7	5	13	21	19	25	19	13	7	1	7	1	13	25	19
8	2	25	16	14	11	24	17	10	3	8	2	21	20	14	15	24	17	6	3
20	11	9	3	22	20	8	1	14	22	16	15	9	3	22	16	8	5	14	22
21#					22#					23#					24#				
4	23	7	15	16	4	12	25	8	16	4	23	7	11	20	4	12	21	8	20
12	20	1	24	8	23	6	19	2	15	12	16	5	24	8	23	10	19	2	11
21	9	13	17	5	17	5	13	21	9	25	9	13	17	1	17	1	13	25	9
18	2	25	6	14	11	24	7	20	3	18	2	21	10	14	15	24	7	16	3
10	11	19	3	22	10	18	1	14	22	6	15	19	3	22	6	18	5	14	22
25#					26#					27#					28#				
5	23	16	14	7	5	11	24	18	7	5	23	16	12	9	5	11	22	18	9
11	9	2	25	18	23	17	10	1	14	11	7	4	25	18	23	19	10	1	12
22	20	13	6	4	6	4	13	22	20	24	20	13	6	2	6	2	13	24	20
8	1	24	17	15	12	25	16	9	3	8	1	22	19	15	14	25	16	7	3
19	12	10	3	21	19	8	2	15	21	17	14	10	3	21	17	8	4	15	21
29#					30#					31#					32#				
5	23	6	14	17	5	11	24	8	17	5	23	6	12	19	5	11	22	8	19
11	19	2	25	8	23	7	20	1	14	11	17	4	25	8	23	9	20	1	12
22	10	13	16	4	16	4	13	22	10	24	10	13	16	2	16	2	13	24	10
18	1	24	7	15	12	25	6	19	3	18	1	22	9	15	14	25	6	17	3
9	12	20	3	21	9	18	2	15	21	7	14	20	3	21	7	18	4	15	21
33#					34#					35#					36#				
6	18	25	14	2	6	15	19	23	2	6	18	25	12	4	6	15	17	23	4
15	4	7	16	23	18	22	1	10	14	15	2	9	16	23	18	24	1	10	12
17	21	13	5	9	5	9	13	17	21	19	21	13	5	7	5	7	13	19	21
3	10	19	22	11	12	16	25	4	8	3	10	17	24	11	14	16	25	2	8
24	12	1	8	20	24	3	7	11	20	22	14	1	8	20	22	3	9	11	20

37#					38#					39#					40#				
6	18	5	14	22	6	15	19	3	22	6	18	5	12	24	6	15	17	3	24
15	24	7	16	3	18	2	21	10	14	15	22	9	16	3	18	4	21	10	12
17	1	13	25	9	25	9	13	17	1	19	1	13	25	7	25	7	13	19	1
23	10	19	2	11	12	16	5	24	8	23	10	17	4	11	14	16	5	22	8
4	12	21	8	20	4	23	7	11	20	2	14	21	8	20	2	23	9	11	20
41#					42#					43#					44#				
7	18	24	15	1	7	14	20	23	1	7	18	24	11	5	7	14	16	23	5
14	5	6	17	23	18	21	2	9	15	14	1	10	17	23	18	25	2	9	11
16	22	13	4	10	4	10	13	16	22	20	22	13	4	6	4	6	13	20	22
3	9	20	21	12	11	17	24	5	8	3	9	16	25	12	15	17	24	1	8
25	11	2	8	19	25	3	6	12	19	21	15	2	8	19	21	3	10	12	19
45#					46#					47#					48#				
7	18	4	15	21	7	14	20	3	21	7	18	4	11	25	7	14	16	3	25
14	25	6	17	3	18	1	22	9	15	14	21	10	17	3	18	5	22	9	11
16	2	13	24	10	24	10	13	16	2	20	2	13	24	6	24	6	13	20	2
23	9	20	1	12	11	17	4	25	8	23	9	16	5	12	15	17	4	21	8
5	11	22	8	19	5	23	6	12	19	1	15	22	8	19	1	23	10	12	19
49#					50#					51#					52#				
9	18	22	15	1	9	12	20	23	1	9	18	22	11	5	9	12	16	23	5
12	5	6	19	23	18	21	4	7	15	12	1	10	19	23	18	25	4	7	11
16	24	13	2	10	2	10	13	16	24	20	24	13	2	6	2	6	13	20	24
3	7	20	21	14	11	19	22	5	8	3	7	16	25	14	15	19	22	1	8
25	11	4	8	17	25	3	6	14	17	21	15	4	8	17	21	3	10	14	17
53#					54#					55#					56#				
9	18	2	15	21	9	12	20	3	21	9	18	2	11	25	9	12	16	3	25
12	25	6	19	3	18	1	24	7	15	12	21	10	19	3	18	5	24	7	11
16	4	13	22	10	22	10	13	16	4	20	4	13	22	6	22	6	13	20	4
23	7	20	1	14	11	19	2	25	8	23	7	16	5	14	15	19	2	21	8
5	11	24	8	17	5	23	6	14	17	1	15	24	8	17	1	23	10	14	17
57#					58#					59#					60#				
10	18	21	14	2	10	11	19	23	2	10	18	21	12	4	10	11	17	23	4
11	4	7	20	23	18	22	5	6	14	11	2	9	20	23	18	24	5	6	12
17	25	13	1	9	1	9	13	17	25	19	25	13	1	7	1	7	13	19	25
3	6	19	22	15	12	20	21	4	8	3	6	17	24	15	14	20	21	2	8
24	12	5	8	16	24	3	7	15	16	22	14	5	8	16	22	3	9	15	16
61#					62#					63#					64#				
10	18	1	14	22	10	11	19	3	22	10	18	1	12	24	10	11	17	3	24
11	24	7	20	3	18	2	25	6	14	11	22	9	20	3	18	4	25	6	12
17	5	13	21	9	21	9	13	17	5	19	5	13	21	7	21	7	13	19	5
23	6	19	2	15	12	20	1	24	8	23	6	17	4	15	14	20	1	22	8
4	12	25	8	16	4	23	7	15	16	2	14	25	8	16	2	23	9	15	16
65#					66#					67#					68#				
16	15	9	23	2	16	8	25	14	2	16	15	7	23	4	16	8	25	12	4
8	22	1	20	14	15	4	17	6	23	8	24	1	20	12	15	2	19	6	23
5	19	13	7	21	7	21	13	5	19	5	17	13	9	21	9	21	13	5	17
12	6	25	4	18	3	20	9	22	11	14	6	25	2	18	3	20	7	24	11
24	3	17	11	10	24	12	1	18	10	22	3	19	11	10	22	14	1	18	10

69#	70#	71#	72#
16 15 9 3 22	16 8 5 14 22	16 15 7 3 24	16 8 5 12 24
8 2 21 20 14	15 24 17 6 3	8 4 21 20 12	15 22 19 6 3
25 19 13 7 1	7 1 13 25 19	25 17 13 9 1	9 1 13 25 17
12 6 5 24 18	23 20 9 2 11	14 6 5 22 18	23 20 7 4 11
4 23 17 11 10	4 12 21 18 10	2 23 19 11 10	2 14 21 18 10
73#	74#	75#	76#
17 14 10 23 1	17 8 24 15 1	17 14 6 23 5	17 8 24 11 5
8 21 2 19 15	14 5 16 7 23	8 25 2 19 11	14 1 20 7 23
4 20 13 6 22	6 22 13 4 20	4 16 13 10 22	10 22 13 4 16
11 7 24 5 18	3 19 10 21 12	15 7 24 1 18	3 19 6 25 12
25 3 16 12 9	25 11 2 18 9	21 3 20 12 9	21 15 2 18 9
77#	78#	79#	80#
17 14 10 3 21	17 8 4 15 21	17 14 6 3 25	17 8 4 11 25
8 1 22 19 15	14 25 16 7 3	8 5 22 19 11	14 21 20 7 3
24 20 13 6 2	6 2 13 24 20	24 16 13 10 2	10 2 13 24 16
11 7 4 25 18	23 19 10 1 12	15 7 4 21 18	23 19 6 5 12
5 23 16 12 9	5 11 22 18 9	1 23 20 12 9	1 15 22 18 9
81#	82#	83#	84#
19 12 10 23 1	19 8 22 15 1	19 12 6 23 5	19 8 22 11 5
8 21 4 17 15	12 5 16 9 23	8 25 4 17 11	12 1 20 9 23
2 20 13 6 24	6 24 13 2 20	2 16 13 10 24	10 24 13 2 16
11 9 22 5 18	3 17 10 21 14	15 9 22 1 18	3 17 6 25 14
25 3 16 14 7	25 11 4 18 7	21 3 20 14 7	21 15 4 18 7
85#	86#	87#	88#
19 12 10 3 21	19 8 2 15 21	19 12 6 3 25	19 8 2 11 25
8 1 24 17 15	12 25 16 9 3	8 5 24 17 11	12 21 20 9 3
22 20 13 6 4	6 4 13 22 20	22 16 13 10 4	10 4 13 22 16
11 9 2 25 18	23 17 10 1 14	15 9 2 21 18	23 17 6 5 14
5 23 16 14 7	5 11 24 18 7	1 23 20 14 7	1 15 24 18 7
89#	90#	91#	92#
20 11 9 23 2	20 8 21 14 2	20 11 7 23 4	20 8 21 12 4
8 22 5 16 14	11 4 17 10 23	8 24 5 16 12	11 2 19 10 23
1 19 13 7 25	7 25 13 1 19	1 17 13 9 25	9 25 13 1 17
12 10 21 4 18	3 16 9 22 15	14 10 21 2 18	3 16 7 24 15
24 3 17 15 6	24 12 5 18 6	22 3 19 15 6	22 14 5 18 6
93#	94#	95#	96#
20 11 9 3 22	20 8 1 14 22	20 11 7 3 24	20 8 1 12 24
8 2 25 16 14	11 24 17 10 3	8 4 25 16 12	11 22 19 10 3
21 19 13 7 5	7 5 13 21 19	21 17 13 9 5	9 5 13 21 17
12 10 1 24 18	23 16 9 2 15	14 10 1 22 18	23 16 7 4 15
4 23 17 15 6	4 12 25 18 6	2 23 19 15 6	2 14 25 18 6
97#	98#	99#	100#
21 15 4 18 7	21 3 20 14 7	21 15 2 18 9	21 3 20 12 9
3 17 6 25 14	15 9 22 1 18	3 19 6 25 12	15 7 24 1 18
10 24 13 2 16	2 16 13 10 24	10 22 13 4 16	4 16 13 10 22
12 1 20 9 23	8 25 4 17 11	14 1 20 7 23	8 25 2 19 11
19 8 22 11 5	19 12 6 23 5	17 8 24 11 5	17 14 6 23 5

21 15 4 8 17	21 3 10 14 17	21 15 2 8 19	21 3 10 12 19
3 7 16 25 14	15 19 22 1 8	3 9 16 25 12	15 17 24 1 8
20 24 13 2 6	2 6 13 20 24	20 22 13 4 6	4 6 13 20 22
12 1 10 19 23	18 25 4 7 11	14 1 10 17 23	18 25 2 9 11
9 18 22 11 5	9 12 16 23 5	7 18 24 11 5	7 14 16 23 5
105#	106#	107#	108#
22 14 5 18 6	22 3 19 15 6	22 14 1 18 10	22 3 19 11 10
3 16 7 24 15	14 10 21 2 18	3 20 7 24 11	14 6 25 2 18
9 25 13 1 17	1 17 13 9 25	9 21 13 5 17	5 17 13 9 21
11 2 19 10 23	8 24 5 16 12	15 2 19 6 23	8 24 1 20 12
20 8 21 12 4	20 11 7 23 4	16 8 25 12 4	16 15 7 23 4
109#	110#	111#	112#
22 14 5 8 16	22 3 9 15 16	22 14 1 8 20	22 3 9 11 20
3 6 17 24 15	14 20 21 2 8	3 10 17 24 11	14 16 25 2 8
19 25 13 1 7	1 7 13 19 25	19 21 13 5 7	5 7 13 19 21
11 2 9 20 23	18 24 5 6 12	15 2 9 16 23	18 24 1 10 12
10 18 21 12 4	10 11 17 23 4	6 18 25 12 4	6 15 17 23 4
113#	114#	115#	116#
24 12 5 18 6	24 3 17 15 6	24 12 1 18 10	24 3 17 11 10
3 16 9 22 15	12 10 21 4 18	3 20 9 22 11	12 6 25 4 18
7 25 13 1 19	1 19 13 7 25	7 21 13 5 19	5 19 13 7 21
11 4 17 10 23	8 22 5 16 14	15 4 17 6 23	8 22 1 20 14
20 8 21 14 2	20 11 9 23 2	16 8 25 14 2	16 15 9 23 2
117#	118#	119#	120#
24 12 5 8 16	24 3 7 15 16	24 12 1 8 20	24 3 7 11 20
3 6 19 22 15	12 20 21 4 8	3 10 19 22 11	12 16 25 4 8
17 25 13 1 9	1 9 13 17 25	17 21 13 5 9	5 9 13 17 21
11 4 7 20 23	18 22 5 6 14	15 4 7 16 23	18 22 1 10 14
10 18 21 14 2	10 11 19 23 2	6 18 25 14 2	6 15 19 23 2
121#	122#	123#	124#
25 11 4 18 7	25 3 16 14 7	25 11 2 18 9	25 3 16 12 9
3 17 10 21 14	11 9 22 5 18	3 19 10 21 12	11 7 24 5 18
6 24 13 2 20	2 20 13 6 24	6 22 13 4 20	4 20 13 6 22
12 5 16 9 23	8 21 4 17 15	14 5 16 7 23	8 21 2 19 15
19 8 22 15 1	19 12 10 23 1	17 8 24 15 1	17 14 10 23 1
125#	126#	127#	128#
25 11 4 8 17	25 3 6 14 17	25 11 2 8 19	25 3 6 12 19
3 7 20 21 14	11 19 22 5 8	3 9 20 21 12	11 17 24 5 8
16 24 13 2 10	2 10 13 16 24	16 22 13 4 10	4 10 13 16 22
12 5 6 19 23	18 21 4 7 15	14 5 6 17 23	18 21 2 9 15
9 18 22 15 1	9 12 20 23 1	7 18 24 15 1	7 14 20 23 1

[Count = 128] OK!

\*\* Calculated and Listed by Kanji Setsuda on May 3, 2017  
working with macOS Sierra 10.12.4 and Xcode ver. 7.3 \*\*

6. 次は、「標準解」を求めよう。単純な回転・反転による異形・派生形を抑えるために、今度も不等式の設定によることにしよう。

その時間題になるのは、規則的な行列の交換などによる異形の発生である。

しかし、ゴールとするものが対称型で同時に汎対角線型であるので、どちらか一方にしか通用しない不等式の立て方は避けなければならない。いろいろ調べて、結局  $n1 < n25; n1 < n5 < n21$ ; とゆるく立てて、攻略するしかないことがわかった。

標準解は、以下のごとくわずかに 16 個得られた。随分と少ない宝物のような存在だ。しかし、これらは、確かに対称五方陣でありかつ汎五方陣でもある「連立型」になっている。

いずれの解も、前に求めておいた対称型および汎型のリストに確かに存在する。次のリストでは、参照した古い番号をわざわざ付記しておいた。

**\*\* 対称汎五方陣の解のリスト(対称型/汎型の参照番号付き) \*\***

[ 1] 4804/ 19/ [ 2] 3314/ 1/ [ 3] 5403/ 20/ [ 4] 3845/ 2/  
 1 23 20 12 9 1 23 20 14 7 1 15 22 18 9 1 15 24 18 7  
 15 7 4 21 18 15 9 2 21 18 23 19 6 5 12 23 17 6 5 14  
 24 16 13 10 2 22 16 13 10 4 10 2 13 24 16 10 4 13 22 16  
 8 5 22 19 11 8 5 24 17 11 14 21 20 7 3 12 21 20 9 3  
 17 14 6 3 25 19 12 6 3 25 17 8 4 11 25 19 8 2 11 25

[ 5] 12658/ 631/ [ 6] 9786/ 613/ [ 7] 13071/ 632/ [ 8] 10389/ 614/  
 2 23 19 11 10 2 23 19 15 6 2 14 21 18 10 2 14 25 18 6  
 14 6 5 22 18 14 10 1 22 18 23 20 7 4 11 23 16 7 4 15  
 25 17 13 9 1 21 17 13 9 5 9 1 13 25 17 9 5 13 21 17  
 8 4 21 20 12 8 4 25 16 12 15 22 19 6 3 11 22 19 10 3  
 16 15 7 3 24 20 11 7 3 24 16 8 5 12 24 20 8 1 12 24

[ 9] 25584/1819/ [10] 22629/1801/ [11] 26051/1820/ [12] 23126/1802/  
 4 23 17 11 10 4 23 17 15 6 4 12 21 18 10 4 12 25 18 6  
 12 6 5 24 18 12 10 1 24 18 23 20 9 2 11 23 16 9 2 15  
 25 19 13 7 1 21 19 13 7 5 7 1 13 25 19 7 5 13 21 19  
 8 2 21 20 14 8 2 25 16 14 15 24 17 6 3 11 24 17 10 3  
 16 15 9 3 22 20 11 9 3 22 16 8 5 14 22 20 8 1 14 22

[13] 30410/2431/ [14] 28638/2413/ [15] 30969/2432/ [16] 29389/2414/  
 5 23 16 12 9 5 23 16 14 7 5 11 22 18 9 5 11 24 18 7  
 11 7 4 25 18 11 9 2 25 18 23 19 10 1 12 23 17 10 1 14  
 24 20 13 6 2 22 20 13 6 4 6 2 13 24 20 6 4 13 22 20  
 8 1 22 19 15 8 1 24 17 15 14 25 16 7 3 12 25 16 9 3  
 17 14 10 3 21 19 12 10 3 21 17 8 4 15 21 19 8 2 15 21

[Count: Sml = 16, SC = 48544, PD = 3600] OK!

7. なるほど奇跡のような宝物だが、老婆心で注釈すると、対称汎五方陣は、あくまでこの形でのみ同時成立しているのもであって、これを各個に対称型変換するとたちまち汎性は崩れて無くなり、汎型変換するとたちまち対称性が崩れて無くなる。

その例を以下に示す。変換法の内容については、第1節・第2節の該当項をお読みいただきたい。

**\*\* 対称型変換と汎型変換の結果 \*\***

0/ [原型]	1/SC	2/SC	3/SC
1 15 24 18 7	15 1 24 7 18	23 17 6 5 14	17 23 6 14 5
23 17 6 5 14	17 23 6 14 5	1 15 24 18 7	15 1 24 7 18
10 4 13 22 16	4 10 13 16 22	10 4 13 22 16	4 10 13 16 22
12 21 20 9 3	21 12 20 3 9	19 8 2 11 25	8 19 2 25 11
19 8 2 11 25	8 19 2 25 11	12 21 20 9 3	21 12 20 3 9



[原型]	0/PD	1/PD	2/PD	3/PD	4/PD
1 15 24 18 7	1 7 18 24 15	2 11 25 19 8	2 8 19 25 11	7 1 15 24 18	
23 17 6 5 14	23 14 5 6 17	20 9 3 12 21	20 21 12 3 9	14 23 17 6 5	
10 4 13 22 16	10 16 22 13 4	13 22 16 10 4	13 4 10 16 22	16 10 4 13 22	
12 21 20 9 3	12 3 9 20 21	6 5 14 23 17	6 17 23 14 5	3 12 21 20 9	
19 8 2 11 25	19 25 11 2 8	24 18 7 1 15	24 15 1 7 18	25 19 8 2 11	

実例を見ると、確かに原型1個から変換されて、対称型3個と汎型五方陣が4個作られているが、対称汎五方陣は原型のみで、他はもはや連立型の方陣ではない。

したがって、対称型や汎型がそれぞれ固有の変換に耐え得る本来ダイナミックな概念であったのに対して、今回の五方陣は1つの状態でのみ成り立つスタティックな現象であると言える。これは致し方が無いことで、連立型はそれほど珍しくも貴重な存在なのである。文学的に表現すれば、2つの音の一瞬の共鳴あるいはハーモニーであると言っても良いだろう。

8. この小論は、鈴木睦教授の『補数変換に対して自己相似な完全五方陣(=汎五方陣)』の内容とオーバーラップするものがある。実際に追試・確認の報告になっている。

「補数変換に対して自己相似な」という教授のハイカラな表現は、日本では古来「対称型方陣」と言われて来たものと内容的にまったく同じだ。だから「対称汎五方陣」という題で、今回小生は追求した。

たしか今から15年くらい前に阿部楽方先生から「対称型で同時に完全型(=汎型)の方陣を全数さがせ」という宿題をいただいた記憶がある。それが五方陣だったかどうかはいま定かではないが、もったいないと言おうか、生意気と言おうか、小生は即座にお断りしてしまった覚えがある(先生の周辺では既に幾つか実例が得られていたのではあるまいか。それはおっしゃらなかったが)。

四方陣で連立型が不可能であることはわかっていた。小生は、五方陣では汎魔方陣そのものが不可能であると思っていた(独断論的誤謬)。また「対称方陣」も「汎魔方陣」も、もともと変化が自在のダイナミックな概念のはずであった。たとえ万が一にも対称型と汎型が同時に実現したとしても、それはあくまで一瞬のことであって、行列をひとたび動かせばもう両立は壊れる。変化には耐えられないスタティックな現象のはずだ。元々ダイナミックな概念とは相容れないのではないか。そんな危ういものを何故求めるのか、と。

その後の経緯は、小生は知らない。誰かが見つけていたかも知れないが、小生は既に研究を止めていた。それに何の貢献もしなかったので、コミュニケーションが途絶えてしまった。

ところが、最近になってホームページを見たら、教授御自身の報告があり、可能だと言う。非常に珍しい現象だが、16個もの解全数を発見したと言う。いささかショックを受けた。「五方陣では、何でもありか」と、正直そう思った。

汎五方陣の存在確認で既に眼を開かされていたから、今度は教授の報告を最初から疑うことはしなかった。小生がやるべきことは、自分の方法で追試して、この発見を確認することである。確かに存在した。そしてそれは、十分に美しかった。たとえ一瞬の虹あるいはハーモニーであったとしても。

鈴木教授に敬意を表し、あらためて全数確定者の名誉を讃えたいと思う。

まさか2つの解集合の間に論理的積集合が存在するとは、小生は夢にも思わなかった。

夢想家の夢も、時には実現することがあるようだ。(on September 20, 2000)

9. その後の研究でわかった大切なことを付け加える。

対称汎五方陣のリストをあらためて五進法分解してみたら、どれもが同じ特徴を有することに気が付いた。上位面も下位面も、

- (1) 各行の内容に (0, 1, 2, 3, 4) の数がそれぞれ1度ずつ現れていて重複が無い。
- (2) 各列の内容に (0, 1, 2, 3, 4) の数がそれぞれ1度ずつ現れていて重複が無い。
- (3) 各汎対角線の内容に (0, 1, 2, 3, 4) の数がそれぞれ1度ずつ現れていて重複が無い。

さらに、

- (4) 上位と下位の数の組み合わせにも、(00, 01, 02, 03, 04, 10, 11, 12, ..., 43, 44) の全数がそれぞれ1回ずつ現れていて重複が無い。

この特徴によって、各行列・各汎対角線の和を求める式は、すべて

$$(0+1+2+3+4) \times 5^1 + (0+1+2+3+4) \times 5^0 = 220_{(五)} = 60_{(十)}$$

と同一になり、定和（1～25を用いる古典的表記では、65に相当する）を実現している。

その特徴に気付けば、もう次からは計算するまでもなく、各行列・各汎対角線5数の定和が直に見て取れるのである。鈴木教授に教わったところ、こういう特長を持つ方陣を「オイラー方陣」と言うそうだ。あの伝説の大数学者オイラー教授が、もう2百年以上も前に研究していたと言う。

われらの珍品は、汎対角線までもこの特徴が成り立つ。感服した小生は、あえてこれを『完全オイラー方陣』と呼びたい。

五進法分解図付きのリストを以下に示そう。

**\* 対称汎五方陣の標準解：16個（五進法分解図/D5i付き）\***

〈古典的表記# [数学的表記] チェック・サム 五進法分解図 上位H 下位L チェック・サム〉

1#	[Math]	Rw CI\P1/P2	/D5i	H Rw CI\P1/P2	L Rw CI\P1/P2
1 23 20 12 9	0 22 19 11	8 60 60 60 60	0 4 3 2 1	10 10 10 10	0 2 4 1 3 10 10 10 10
15 7 4 21 18	14 6 3 20 17	60 60 60 60	2 1 0 4 3	10 10 10 10	4 1 3 0 2 10 10 10 10
24 16 13 10 2	23 15 12 9 1	60 60 60 60	4 3 2 1 0	10 10 10 10	3 0 2 4 1 10 10 10 10
8 5 22 19 11	7 4 21 18 10	60 60 60 60	1 0 4 3 2	10 10 10 10	2 4 1 3 0 10 10 10 10
17 14 6 3 25	16 13 5 2 24	60 60 60 60	3 2 1 0 4	10 10 10 10	1 3 0 2 4 10 10 10 10
2#	[Math]	Rw CI\P1/P2	/D5i	H Rw CI\P1/P2	L Rw CI\P1/P2
1 23 20 14 7	0 22 19 13 6	60 60 60 60	0 4 3 2 1	10 10 10 10	0 2 4 3 1 10 10 10 10
15 9 2 21 18	14 8 1 20 17	60 60 60 60	2 1 0 4 3	10 10 10 10	4 3 1 0 2 10 10 10 10
22 16 13 10 4	21 15 12 9 3	60 60 60 60	4 3 2 1 0	10 10 10 10	1 0 2 4 3 10 10 10 10
8 5 24 17 11	7 4 23 16 10	60 60 60 60	1 0 4 3 2	10 10 10 10	2 4 3 1 0 10 10 10 10
19 12 6 3 25	18 11 5 2 24	60 60 60 60	3 2 1 0 4	10 10 10 10	3 1 0 2 4 10 10 10 10
3#	[Math]	Rw CI\P1/P2	/D5i	H Rw CI\P1/P2	L Rw CI\P1/P2
1 15 22 18 9	0 14 21 17 8	60 60 60 60	0 2 4 3 1	10 10 10 10	0 4 1 2 3 10 10 10 10
23 19 6 5 12	22 18 5 4 11	60 60 60 60	4 3 1 0 2	10 10 10 10	2 3 0 4 1 10 10 10 10
10 2 13 24 16	9 1 12 23 15	60 60 60 60	1 0 2 4 3	10 10 10 10	4 1 2 3 0 10 10 10 10
14 21 20 7 3	13 20 19 6 2	60 60 60 60	2 4 3 1 0	10 10 10 10	3 0 4 1 2 10 10 10 10
17 8 4 11 25	16 7 3 10 24	60 60 60 60	3 1 0 2 4	10 10 10 10	1 2 3 0 4 10 10 10 10
4#	[Math]	Rw CI\P1/P2	/D5i	H Rw CI\P1/P2	L Rw CI\P1/P2
1 15 24 18 7	0 14 23 17 6	60 60 60 60	0 2 4 3 1	10 10 10 10	0 4 3 2 1 10 10 10 10
23 17 6 5 14	22 16 5 4 13	60 60 60 60	4 3 1 0 2	10 10 10 10	2 1 0 4 3 10 10 10 10
10 4 13 22 16	9 3 12 21 15	60 60 60 60	1 0 2 4 3	10 10 10 10	4 3 2 1 0 10 10 10 10
12 21 20 9 3	11 20 19 8 2	60 60 60 60	2 4 3 1 0	10 10 10 10	1 0 4 3 2 10 10 10 10
19 8 2 11 25	18 7 1 10 24	60 60 60 60	3 1 0 2 4	10 10 10 10	3 2 1 0 4 10 10 10 10
5#	[Math]	Rw CI\P1/P2	/D5i	H Rw CI\P1/P2	L Rw CI\P1/P2
2 23 19 11 10	1 22 18 10 9	60 60 60 60	0 4 3 2 1	10 10 10 10	1 2 3 0 4 10 10 10 10
14 6 5 22 18	13 5 4 21 17	60 60 60 60	2 1 0 4 3	10 10 10 10	3 0 4 1 2 10 10 10 10
25 17 13 9 1	24 16 12 8 0	60 60 60 60	4 3 2 1 0	10 10 10 10	4 1 2 3 0 10 10 10 10
8 4 21 20 12	7 3 20 19 11	60 60 60 60	1 0 4 3 2	10 10 10 10	2 3 0 4 1 10 10 10 10
16 15 7 3 24	15 14 6 2 23	60 60 60 60	3 2 1 0 4	10 10 10 10	0 4 1 2 3 10 10 10 10
6#	[Math]	Rw CI\P1/P2	/D5i	H Rw CI\P1/P2	L Rw CI\P1/P2
2 23 19 15 6	1 22 18 14 5	60 60 60 60	0 4 3 2 1	10 10 10 10	1 2 3 4 0 10 10 10 10
14 10 1 22 18	13 9 0 21 17	60 60 60 60	2 1 0 4 3	10 10 10 10	3 4 0 1 2 10 10 10 10
21 17 13 9 5	20 16 12 8 4	60 60 60 60	4 3 2 1 0	10 10 10 10	0 1 2 3 4 10 10 10 10
8 4 25 16 12	7 3 24 15 11	60 60 60 60	1 0 4 3 2	10 10 10 10	2 3 4 0 1 10 10 10 10
20 11 7 3 24	19 10 6 2 23	60 60 60 60	3 2 1 0 4	10 10 10 10	4 0 1 2 3 10 10 10 10
7#	[Math]	Rw CI\P1/P2	/D5i	H Rw CI\P1/P2	L Rw CI\P1/P2
2 14 21 18 10	1 13 20 17 9	60 60 60 60	0 2 4 3 1	10 10 10 10	1 3 0 2 4 10 10 10 10
23 20 7 4 11	22 19 6 3 10	60 60 60 60	4 3 1 0 2	10 10 10 10	2 4 1 3 0 10 10 10 10
9 1 13 25 17	8 0 12 24 16	60 60 60 60	1 0 2 4 3	10 10 10 10	3 0 2 4 1 10 10 10 10
15 22 19 6 3	14 21 18 5 2	60 60 60 60	2 4 3 1 0	10 10 10 10	4 1 3 0 2 10 10 10 10
16 8 5 12 24	15 7 4 11 23	60 60 60 60	3 1 0 2 4	10 10 10 10	0 2 4 1 3 10 10 10 10

8#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
2	14 25 18 6	1	13 24 17 5	60	60	60	60	0	2 4 3 1	10	10	10	10	1	3 4 2 0	10	10	10	10
23	16 7 4 15	22	15 6 3 14	60	60	60	60	4	3 1 0 2	10	10	10	10	2	0 1 3 4	10	10	10	10
9	5 13 21 17	8	4 12 20 16	60	60	60	60	1	0 2 4 3	10	10	10	10	3	4 2 0 1	10	10	10	10
11	22 19 10 3	10	21 18 9 2	60	60	60	60	2	4 3 1 0	10	10	10	10	0	1 3 4 2	10	10	10	10
20	8 1 12 24	19	7 0 11 23	60	60	60	60	3	1 0 2 4	10	10	10	10	4	2 0 1 3	10	10	10	10
9#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
4	23 17 11 10	3	22 16 10 9	60	60	60	60	0	4 3 2 1	10	10	10	10	3	2 1 0 4	10	10	10	10
12	6 5 24 18	11	5 4 23 17	60	60	60	60	2	1 0 4 3	10	10	10	10	1	0 4 3 2	10	10	10	10
25	19 13 7 1	24	18 12 6 0	60	60	60	60	4	3 2 1 0	10	10	10	10	4	3 2 1 0	10	10	10	10
8	2 21 20 14	7	1 20 19 13	60	60	60	60	1	0 4 3 2	10	10	10	10	2	1 0 4 3	10	10	10	10
16	15 9 3 22	15	14 8 2 21	60	60	60	60	3	2 1 0 4	10	10	10	10	0	4 3 2 1	10	10	10	10
10#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
4	23 17 15 6	3	22 16 14 5	60	60	60	60	0	4 3 2 1	10	10	10	10	3	2 1 4 0	10	10	10	10
12	10 1 24 18	11	9 0 23 17	60	60	60	60	2	1 0 4 3	10	10	10	10	1	4 0 3 2	10	10	10	10
21	19 13 7 5	20	18 12 6 4	60	60	60	60	4	3 2 1 0	10	10	10	10	0	3 2 1 4	10	10	10	10
8	2 25 16 14	7	1 24 15 13	60	60	60	60	1	0 4 3 2	10	10	10	10	2	1 4 0 3	10	10	10	10
20	11 9 3 22	19	10 8 2 21	60	60	60	60	3	2 1 0 4	10	10	10	10	4	0 3 2 1	10	10	10	10
11#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
4	12 21 18 10	3	11 20 17 9	60	60	60	60	0	2 4 3 1	10	10	10	10	3	1 0 2 4	10	10	10	10
23	20 9 2 11	22	19 8 1 10	60	60	60	60	4	3 1 0 2	10	10	10	10	2	4 3 1 0	10	10	10	10
7	1 13 25 19	6	0 12 24 18	60	60	60	60	1	0 2 4 3	10	10	10	10	1	0 2 4 3	10	10	10	10
15	24 17 6 3	14	23 16 5 2	60	60	60	60	2	4 3 1 0	10	10	10	10	4	3 1 0 2	10	10	10	10
16	8 5 14 22	15	7 4 13 21	60	60	60	60	3	1 0 2 4	10	10	10	10	0	2 4 3 1	10	10	10	10
12#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
4	12 25 18 6	3	11 24 17 5	60	60	60	60	0	2 4 3 1	10	10	10	10	3	1 4 2 0	10	10	10	10
23	16 9 2 15	22	15 8 1 14	60	60	60	60	4	3 1 0 2	10	10	10	10	2	0 3 1 4	10	10	10	10
7	5 13 21 19	6	4 12 20 18	60	60	60	60	1	0 2 4 3	10	10	10	10	1	4 2 0 3	10	10	10	10
11	24 17 10 3	10	23 16 9 2	60	60	60	60	2	4 3 1 0	10	10	10	10	0	3 1 4 2	10	10	10	10
20	8 1 14 22	19	7 0 13 21	60	60	60	60	3	1 0 2 4	10	10	10	10	4	2 0 3 1	10	10	10	10
13#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
5	23 16 12 9	4	22 15 11 8	60	60	60	60	0	4 3 2 1	10	10	10	10	4	2 0 1 3	10	10	10	10
11	7 4 25 18	10	6 3 24 17	60	60	60	60	2	1 0 4 3	10	10	10	10	0	1 3 4 2	10	10	10	10
24	20 13 6 2	23	19 12 5 1	60	60	60	60	4	3 2 1 0	10	10	10	10	3	4 2 0 1	10	10	10	10
8	1 22 19 15	7	0 21 18 14	60	60	60	60	1	0 4 3 2	10	10	10	10	2	0 1 3 4	10	10	10	10
17	14 10 3 21	16	13 9 2 20	60	60	60	60	3	2 1 0 4	10	10	10	10	1	3 4 2 0	10	10	10	10
14#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
5	23 16 14 7	4	22 15 13 6	60	60	60	60	0	4 3 2 1	10	10	10	10	4	2 0 3 1	10	10	10	10
11	9 2 25 18	10	8 1 24 17	60	60	60	60	2	1 0 4 3	10	10	10	10	0	3 1 4 2	10	10	10	10
22	20 13 6 4	21	19 12 5 3	60	60	60	60	4	3 2 1 0	10	10	10	10	1	4 2 0 3	10	10	10	10
8	1 24 17 15	7	0 23 16 14	60	60	60	60	1	0 4 3 2	10	10	10	10	2	0 3 1 4	10	10	10	10
19	12 10 3 21	18	11 9 2 20	60	60	60	60	3	2 1 0 4	10	10	10	10	3	1 4 2 0	10	10	10	10
15#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
5	11 22 18 9	4	10 21 17 8	60	60	60	60	0	2 4 3 1	10	10	10	10	4	0 1 2 3	10	10	10	10
23	19 10 1 12	22	18 9 0 11	60	60	60	60	4	3 1 0 2	10	10	10	10	2	3 4 0 1	10	10	10	10
6	2 13 24 20	5	1 12 23 19	60	60	60	60	1	0 2 4 3	10	10	10	10	0	1 2 3 4	10	10	10	10
14	25 16 7 3	13	24 15 6 2	60	60	60	60	2	4 3 1 0	10	10	10	10	3	4 0 1 2	10	10	10	10
17	8 4 15 21	16	7 3 14 20	60	60	60	60	3	1 0 2 4	10	10	10	10	1	2 3 4 0	10	10	10	10
16#		[Math]	Rw C1\P1/P2	/D5i	H Rw C1\P1/P2	L Rw C1\P1/P2													
5	11 24 18 7	4	10 23 17 6	60	60	60	60	0	2 4 3 1	10	10	10	10	4	0 3 2 1	10	10	10	10
23	17 10 1 14	22	16 9 0 13	60	60	60	60	4	3 1 0 2	10	10	10	10	2	1 4 0 3	10	10	10	10
6	4 13 22 20	5	3 12 21 19	60	60	60	60	1	0 2 4 3	10	10	10	10	0	3 2 1 4	10	10	10	10
12	25 16 9 3	11	24 15 8 2	60	60	60	60	2	4 3 1 0	10	10	10	10	1	4 0 3 2	10	10	10	10
19	8 2 15 21	18	7 1 14 20	60	60	60	60	3	1 0 2 4	10	10	10	10	3	2 1 4 0	10	10	10	10

[Count of Solution Counts = 16]

\*\* Calculated by Kanji Setsuda on Apr. 3, 2016 with MacOSX 10.11.4 & Xcode 7.3 \*\*

五進法分解図の各パターンを、上位／下位の区別無く取り出して分類してみると、相互に良く似てはいるが、相異なる16個のパターンが使われていることがわかった。

しかも、この16個には、面白い特徴が幾つもある。

上位には、2/, 4/ という2つのパターンしか使われていないのである。

また以下のように、16パターンを4つのグループに分けてみた。

そう、どのグループでも、内部で90度ずつの回転が連続して起きている。

#### [五進法分解図のパターン : 16面]

1/	6/	16/	11/
0 2 4 1 3	1 2 3 4 0	4 2 0 3 1	3 2 1 0 4
4 1 3 0 2	3 4 0 1 2	0 3 1 4 2	1 0 4 3 2
3 0 2 4 1	0 1 2 3 4	1 4 2 0 3	4 3 2 1 0
2 4 1 3 0	2 3 4 0 1	2 0 3 1 4	2 1 0 4 3
1 3 0 2 4	4 0 1 2 3	3 1 4 2 0	0 4 3 2 1
2/	12/	15/	5/
0 2 4 3 1	3 2 1 4 0	4 2 0 1 3	1 2 3 0 4
4 3 1 0 2	1 4 0 3 2	0 1 3 4 2	3 0 4 1 2
1 0 2 4 3	0 3 2 1 4	3 4 2 0 1	4 1 2 3 0
2 4 3 1 0	2 1 4 0 3	2 0 1 3 4	2 3 0 4 1
3 1 0 2 4	4 0 3 2 1	1 3 4 2 0	0 4 1 2 3
3/	8/	14/	9/
0 4 1 2 3	1 3 4 2 0	4 0 3 2 1	3 1 0 2 4
2 3 0 4 1	2 0 1 3 4	2 1 4 0 3	2 4 3 1 0
4 1 2 3 0	3 4 2 0 1	0 3 2 1 4	1 0 2 4 3
3 0 4 1 2	0 1 3 4 2	1 4 0 3 2	4 3 1 0 2
1 2 3 0 4	4 2 0 1 3	3 2 1 4 0	0 2 4 3 1
4/	10/	13/	7/
0 4 3 2 1	3 1 4 2 0	4 0 1 2 3	1 3 0 2 4
2 1 0 4 3	2 0 3 1 4	2 3 4 0 1	2 4 1 3 0
4 3 2 1 0	1 4 2 0 3	0 1 2 3 4	3 0 2 4 1
1 0 4 3 2	0 3 1 4 2	3 4 0 1 2	4 1 3 0 2
3 2 1 0 4	4 2 0 3 1	1 2 3 4 0	0 2 4 1 3

しかも、各グループ先頭の4個どうしは、{1/, 4/}, {2/, 3/} が相互に反転形になっている。ということは、本質的には、16個が2個の「基本パターン」で出来ていることになる。その基本の2個には {2/, 4/} を指名したら良いであろう。

美しい構造が見えて来た。嬉しくなった。

Sep. 20, 2000(初稿); Mar. 26, 2001(二稿); Sep. 12, 2002(三稿);

Feb. 22, 2006(改定清書稿); Apr. 3, 2016 and May 3, 2017(改定清書稿);

macOS Sierra 10.12.4 & Xcode 7.3 による計数・作表 : by 摂田 寛二(Kanji Setsuda)

E-Mail Address: jag12001@nifty.com

#### \* 付録 : プログラム・リスト \*

```

/** Simultaneous Type of Magic Squares of Order 5: **
/** Both Self-Complementary and Pan-Diagonal; Built by **
/** the ordinary method with Decomposed Diagrams by 5i System **
/** File: 'MS55SmlD5i.c' Dictated by Kanji Setsuda **
/** in 2003, 2006, 2011, 2012; Revised on Apr. 3, 2016 **
/** Working with MacOSX 10.11.4 & Xcode 7.3 **

```

```

//
#include <stdio.h>
//
short int cnt;
short nm[26], uflg[26], tn[26];
short CC, LSM;
short anm[7][27];
short cs[5][21];
//
void stp01(void), stp02(void), stp03(void), stp04(void), stp05(void), stp06(void);
void stp07(void), stp08(void), stp09(void), stp10(void), stp11(void), stp12(void);
void stp13(void), stp14(void), stp15(void), stp16(void);
void solprint(void), prlsol(void);
void chksms(short x);
//
/* Main Program *
int main(void){
short n;
printf("\n");
printf("** Simultaneous Type of MS55: Both Self-Complementary & Pan-Diagonal;\n");
printf("  Built by the Ordinary Method and Listed with /D5i **\n");
for(n=0;n<26;n++){uflg[n]=0;}
nm[13]=13; uflg[13]=1;
CC=26; LSM=65; cnt=0;
printf(" [List of Standard Solutions with Decomposed Units /D5i and with Check-sums]\n");
stp01(); // Make them by the Ordinary Method
printf(" [Count of Solution Counts = %d]\n",cnt);
printf("\n");
printf("** Calculated and Listed by Kanji Setsuda on\n");
printf("  Apr. 3, 2016 with MacOSX 10.11.4 & Xcode 7.3 **\n");
printf("\n");
return 0;
}
//
/** Basic Form for the Simultaneous MS55: SC & PD; **
//
// 2 3 4 5 | 1 | 2 | 3 | 4 | 5 | 1 2 3 4
// 7 8 9 10 | 6 | 7 | 8 | 9 | 10 | 6 7 8 9
// 12 13 14 15 | 11 | 12 | 13 | 14 | 15 | 11 12 13 14
// 17 18 19 20 | 16 | 17 | 18 | 19 | 20 | 16 17 18 19
// 22 23 24 25 | 21 | 22 | 23 | 24 | 25 | 21 22 23 24
//
/** Define Conditions for Rows and Columns: C=65 **
// n1+n2+n3+n4+n5=C ... rw1 | n1+n6+n11+n16+n21=C ... cl1
// n6+n7+n8+n9+n10=C ... rw2 | n2+n7+n12+n17+n22=C ... cl2
// n11+n12+n13+n14+n15=C ... rw3 | n3+n8+n13+n18+n23=C ... cl3
// n16+n17+n18+n19+n20=C ... rw4 | n4+n9+n14+n19+n24=C ... cl4
// n21+n22+n23+n24+n25=C ... rw5 | n5+n10+n15+n20+n25=C ... cl5
/** Pan-Diagonal Conditions of the Object: C=65 **
// n1+n7+n13+n19+n25=C ... pd1 | n1+n10+n14+n18+n22=C ... pd6
// n2+n8+n14+n20+n21=C ... pd2 | n2+n6+n15+n19+n23=C ... pd7
// n3+n9+n15+n16+n22=C ... pd3 | n3+n7+n11+n20+n24=C ... pd8
// n4+n10+n11+n17+n23=C ... pd4 | n4+n8+n12+n16+n25=C ... pd9
// n5+n6+n12+n18+n24=C ... pd5 | n5+n9+n13+n17+n21=C ... pd10
/** Self-Complementary Conditions of the Object: SC=26 **
// n1+n25=n2+n24=n3+n23=n4+n22=n5+n21=n6+n20=n7+n19=
// n8+n18=n9+n17=n10+n16=n11+n15=n12+n14=n13+n13=SC ... scc
/** List-forming Inequality Conditions for the Standard Solutions **
// n1<n5; n5<n21; and n1<n25;
//

```

```

// Level #1:
/** Set n1 & n25 & (n1<n25) *
void stp01(){
  short a, b;
  for(a=1;a<13;a++){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[1]=a; nm[25]=b;
      uflg[a]=1; uflg[b]=1;
      stp02();
      uflg[b]=0; uflg[a]=0;}
  }
}
/** Set n2 & n24 *
void stp02(){
  short a, b;
  for(a=25;a>0;a--){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[2]=a; nm[24]=b;
      uflg[a]=1; uflg[b]=1;
      stp03();
      uflg[b]=0; uflg[a]=0;}
  }
}
/** Set n6=39-n1-n2 & n20 *
void stp03(){
  short a, b;
  a=39-nm[1]-nm[2];
  if((0<a)&&(a<26)){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[6]=a; nm[20]=b;
      uflg[a]=1; uflg[b]=1;
      stp04();
      uflg[b]=0; uflg[a]=0;}}
}
/** Set n3 & n23 *
void stp04(){
  short a, b;
  for(a=1;a<26;a++){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[3]=a; nm[23]=b;
      uflg[a]=1; uflg[b]=1;
      stp05();
      uflg[b]=0; uflg[a]=0;}
  }
}
/** Set n4 & n22 *
void stp05(){
  short a, b;
  for(a=1;a<26;a++){b=CC-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[4]=a; nm[22]=b;
      uflg[a]=1; uflg[b]=1;
      stp06();
      uflg[b]=0; uflg[a]=0;}
  }
}
/** Set n5=LSM-n1-n2-n3-n4 & n21 & (n1<n5<n21) *
void stp06(){
  short a, b;
  a=LSM-nm[1]-nm[2]-nm[3]-nm[4];
  b=LSM-nm[25]-nm[24]-nm[23]-nm[22];
  if((nm[1]<a)&&(a<b)&&(a+b==CC)){
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[5]=a; nm[21]=b;

```

```

        uflg[a]=1; uflg[b]=1;
        stp07();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set n8=LSM-n2-n6-n21-n23 & n18 *
void stp07(){
short a, b;
a=LSM-nm[2]-nm[6]-nm[21]-nm[23];
if((0<a)&&(a<26)){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[8]=a; nm[18]=b;
uflg[a]=1; uflg[b]=1;
stp08();
uflg[b]=0; uflg[a]=0;}}}
}
/* Set n10=39-n4-n5 & n16 *
void stp08(){
short a, b;
a=39-nm[4]-nm[5];
if((0<a)&&(a<26)){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[10]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
stp09();
uflg[b]=0; uflg[a]=0;}}}
}
/* Set n11=LSM-n1-n6-n16-n21 & n15 *
void stp09(){
short a, b;
a=LSM-nm[1]-nm[6]-nm[16]-nm[21];
if((0<a)&&(a<26)){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[11]=a; nm[15]=b;
uflg[a]=1; uflg[b]=1;
stp10();
uflg[b]=0; uflg[a]=0;}}}
}
/* Set n7=LSM-n1-n3-n11-n13 & n19 *
void stp10(){
short a, b;
a=LSM-nm[1]-nm[3]-nm[11]-nm[13];
if((0<a)&&(a<26)){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[7]=a; nm[19]=b;
uflg[a]=1; uflg[b]=1;
stp11();
uflg[b]=0; uflg[a]=0;}}}
}
/* Set n9=LSM-n6-n7-n8-n10 & n17 *
void stp11(){
short a, b;
a=LSM-nm[6]-nm[7]-nm[8]-nm[10];
if((0<a)&&(a<26)){b=CC-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[9]=a; nm[17]=b;
uflg[a]=1; uflg[b]=1;
stp12();
uflg[b]=0; uflg[a]=0;}}}
}
/* Set n12=LSM-n2-n7-n17-n22 *
void stp12(){
short a, b;
a=LSM-nm[2]-nm[7]-nm[17]-nm[22];
if((0<a)&&(a<26)){b=CC-a;

```

```

    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[12]=a; nm[14]=b;
        uflg[a]=1; uflg[b]=1;
        stp13();
        uflg[b]=0; uflg[a]=0;}}
}
/* Checks n11+n12+n13+n14+n15=65 & Others *
void stp13(void){
    short sm1, sm2;
    sm1=nm[11]+nm[12]+nm[13]+nm[14]+nm[15];
    sm2=nm[3]+nm[8]+nm[13]+nm[18]+nm[23];
    if((sm1==LSM)&&(sm2==LSM)){stp14();}
}
/* Checks n1+n10+n14+n18+n22=65 & Others *
void stp14(void){
    short sm1, sm2, sm3;
    sm1=nm[1]+nm[10]+nm[14]+nm[18]+nm[22];
    sm2=nm[2]+nm[8]+nm[14]+nm[20]+nm[21];
    sm3=nm[2]+nm[6]+nm[15]+nm[19]+nm[23];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){stp15();}
}
/* Checks n1+n10+n14+n18+n22=65 & Others *
void stp15(void){
    short sm1, sm2, sm3;
    sm1=nm[3]+nm[9]+nm[15]+nm[16]+nm[22];
    sm2=nm[3]+nm[7]+nm[11]+nm[20]+nm[24];
    sm3=nm[4]+nm[10]+nm[11]+nm[17]+nm[23];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)){stp16();}
}
/* The Last Checks *
void stp16(void){
    short sm1, sm2;
    sm1=nm[4]+nm[8]+nm[12]+nm[16]+nm[25];
    sm2=nm[5]+nm[6]+nm[12]+nm[18]+nm[24];
    if((sm1==LSM)&&(sm2==LSM)){solprint();}
}
//
// Print Answers with Decomposed Units and Check Sums
void solprint(){
    short n;
    cnt++;
    anm[0][0]=cnt;
    for(n=1;n<26;n++){anm[0][n]=nm[n];}
    for(n=1;n<26;n++){tn[n]=nm[n]-1; anm[1][n]=tn[n];}; chksms(1);
    for(n=1;n<26;n++){tn[n]=(nm[n]-1)/5; anm[2][n]=tn[n];}; chksms(2);
    for(n=1;n<26;n++){tn[n]=(nm[n]-1)%5; anm[3][n]=tn[n];}; chksms(3);
    pr1sol();
}
//
// Print the Answer one by one
void pr1sol(){
    short l, l5, n;
    printf("%15d# [Math] IRw|C|\\P1/P2| ", anm[0][0]);
    printf("/D5i HRw|C|\\P1/P2| LIRw|C|\\P1/P2|\\n");
    for(l=0;l<5;l++){l5=l*5;
        printf(" ");
        for(n=1;n<6;n++){printf("%3d", anm[0][l5+n]);}
        printf(" ");
        for(n=1;n<6;n++){printf("%3d", anm[1][l5+n]);}
        printf("|%2d|%2d|%2d|%2d| ", cs[1][l+1], cs[1][l+6], cs[1][l+11], cs[1][l+16]);
        for(n=1;n<6;n++){printf("%2d", anm[2][l5+n]);}
        printf("|%2d|%2d|%2d|%2d| ", cs[2][l+1], cs[2][l+6], cs[2][l+11], cs[2][l+16]);
        for(n=1;n<6;n++){printf("%2d", anm[3][l5+n]);}
        printf("|%2d|%2d|%2d|%2d|\\n", cs[3][l+1], cs[3][l+6], cs[3][l+11], cs[3][l+16]);
    }
}

```



```

    }
}
//
/** Check Sums *
void chksms(short x){
    cs[x][1]=tn[1]+tn[2]+tn[3]+tn[4]+tn[5];
    cs[x][2]=tn[6]+tn[7]+tn[8]+tn[9]+tn[10];
    cs[x][3]=tn[11]+tn[12]+tn[13]+tn[14]+tn[15];
    cs[x][4]=tn[16]+tn[17]+tn[18]+tn[19]+tn[20];
    cs[x][5]=tn[21]+tn[22]+tn[23]+tn[24]+tn[25];
    cs[x][6]=tn[1]+tn[6]+tn[11]+tn[16]+tn[21];
    cs[x][7]=tn[2]+tn[7]+tn[12]+tn[17]+tn[22];
    cs[x][8]=tn[3]+tn[8]+tn[13]+tn[18]+tn[23];
    cs[x][9]=tn[4]+tn[9]+tn[14]+tn[19]+tn[24];
    cs[x][10]=tn[5]+tn[10]+tn[15]+tn[20]+tn[25];
//
    cs[x][11]=tn[1]+tn[7]+tn[13]+tn[19]+tn[25];
    cs[x][12]=tn[2]+tn[8]+tn[14]+tn[20]+tn[21];
    cs[x][13]=tn[3]+tn[9]+tn[15]+tn[16]+tn[22];
    cs[x][14]=tn[4]+tn[10]+tn[11]+tn[17]+tn[23];
    cs[x][15]=tn[5]+tn[6]+tn[12]+tn[18]+tn[24];
    cs[x][16]=tn[1]+tn[10]+tn[14]+tn[18]+tn[22];
    cs[x][17]=tn[2]+tn[6]+tn[15]+tn[19]+tn[23];
    cs[x][18]=tn[3]+tn[7]+tn[11]+tn[20]+tn[24];
    cs[x][19]=tn[4]+tn[8]+tn[12]+tn[16]+tn[25];
    cs[x][20]=tn[5]+tn[9]+tn[13]+tn[17]+tn[21];
}
//

```