Partitioning integers in \( n \) dimensions

A. G. Bell

*Atlas Computer Laboratory, Chilton, Didcot, Berkshire*
*Now at Rutherford Laboratory, Chilton, Didcot, Berkshire*

In one dimension, it is possible to partition an integer \( N \) into \( 2^{N-1} \) ordered sets of non-zero integers. Likewise we can partition an integer \( N \) into exactly \( K \) non-negative integers in \( \binom{N + K - 1}{K - 1} \) ways.

In the present paper, for particular cases in 2 and 3 dimensions, we obtain exact values (many by counting) of partitions into matrices of non-negative integers. Some implications and formulae are obtained or conjectured.

(Received November 1968)

---

**Partitioning integers in square matrices**

**Statement of problem:**

Given a square matrix of \( m \) columns and rows, containing integers \( a_{ij} \) such that:

1. \( \sum_{i=1}^{m} a_{ij} = \sum_{j=1}^{m} a_{ij} = N \) (a constant)
2. \( 0 < a_{ij} < N \)

How many different arrangements or partitions can the integers within the matrix assume?

**Notation for square matrix:**

The number of partitions of the integer \( N \) in a matrix of \( m \) columns and \( m \) rows is represented by \( P_m^m(N) \).

For \( m = 1 \) no partitioning can occur, hence we say

\[
P_1^1(N) = 1
\]

For \( m = 2 \) the partition is completely defined by the contents of one cell and, as this may take values between 0 and \( N \), therefore

\[
P_2^2(N) = N + 1
\]

The case of \( m = 3 \) is less trivial. Fig. 1 represents the 6 partitions or arrangements of the matrix content for \( N = 1 \).

and Fig. 2 represents the 21 partitions for \( N = 2 \).

The number of arrangements for \( N > 2 \) can be calculated by generating and counting them. Appendix 1 is an ALGOL program to do this, it consists mainly of two algorithms which are executed alternately after reading in \( m \) and \( N \) and initialising. Briefly, the two algorithms are:

(a) **BASIC OPERATION**

We have to fill out a matrix with integers such that the sum in each row and each column is \( N \). We store two vectors \( \text{sumrow} \) and \( \text{sumcol} \) which record the, as yet, unused part of the sum of each row and column. Initially every element in \( \text{sumrow} \) and \( \text{sumcol} \) is set to \( N \) (see (C1) in the program). We now move across each row from left to right (C3) and move down the rows, as completed, from top to bottom (C2); the order in which elements are considered when \( m = 3 \) is shown in Fig. 3a. The BASIC OPERATION stores in each cell, as visited, the lesser of the two elements corresponding to the cell position, one in \( \text{sumrow} \), the other in \( \text{sumcol} \) (these indicate the unused part of \( N \) (C4) and subtracts the amount set from each (C5). On completion the vectors \( \text{sumrow} \) and \( \text{sumcol} \) are both zero.

The first time through the BASIC OPERATION will produce in the array \( A \) the 'Starting Position' which is characterised by the integers in each row being inserted as far to the left as possible, an example of this is Matrix 1 of Fig. 2.

The count is incremented by one (C6) and the program generates the next partition via

(b) **BACK TRACK**

Again moving from left to right in rows (C8), but this time upwards from row to row (C7), the BACK TRACK (see Fig. 3b), covers each cell in turn. At each, we add the contents of the cell back into the associated elements of \( \text{sumrow} \) and \( \text{sumcol} \) (C9); we could also zeroise the cell, if it were necessary, at this stage. We then test if the content of the cell could be increased by a unit, i.e. we see whether \( a_{ij} + 1 \) is less than or equal to the lesser of the two corresponding revised elements in \( \text{sumrow} \) and \( \text{sumcol} \) (C10). If it is not, the BACK TRACK continues to the next cell. If, however, we can use \( a_{ij} + 1 \) as a new \( a_{ik} \) (C11, C12) we return to BASIC OPERATION (C14) after suitably resetting the row and column counts (C13), so that only cells already visited during BACK TRACK are reset (this is why it was not
necessary to zeroise them); the unvisited cells are correct for a new partition without alteration. We thus obtain the next partition. In Fig. 2 the cell increased by 1 at each stage is underlined. Eventually BACK TRACK will be unable to increase any element (see Matrix 21 of Fig. 2), the program then prints the count and terminates.

The author recommends anyone who wishes to fully understand the program to regenerate the 21 partitions of $P_3(2)$ or, better still, to try to find a simpler algorithm to produce them.

$$
\begin{array}{cccccccccc}
1 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 1 & 1 & 0 & 2 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 1 & 1 & 0 & 2 & 0 \\
0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 \\
0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 \\
0 & 0 & 2 & 0 & 0 & 1 & 1 & 0 & 2 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
\end{array}
$$

Fig. 2

It is difficult to prove that these algorithms will generate all the arrangements. However, as will be shown later by theory, for all cases which yielded to mathematical analysis, the program gave the predicted value.

There are numerous short cuts which can be implemented to speed up the program. For example, when back tracking, no element in the first column or last row can ever be increased, so these can be ignored.

Another short cut is to consider the patterns occurring in the first row only and the number of ways they can occur. For example, in the case of $P_3(3)$ the first row can contain a 3 in one of the cells in 3 different ways, the numbers 1 and 2 in two of the cells in 6 different ways and the number 1 in each of the cells in only 1 way.

Therefore, if the program calculates and starts from the three positions shown in Fig. 4

$$
\begin{array}{cccccccccc}
(1) & 1 & 2 & (2) & 2 & 1 & 0 & (3) & 1 & 1 & 1 \\
0 & 3 & 0 & 1 & 2 & 0 & 2 & 1 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 3 & 0 & 1 & 2 & 0 \\
\end{array}
$$

Fig. 4

and does not consider increasing any element in the first row then the number of possible partitions is

$$3 \times (a) + 6 \times (b) + (c)$$

By extending this pre-analysis the number of partitions can be calculated very quickly.

Results by computer

In the case of $P_3^k(N)$ the computer obtained the following:

$$
\begin{array}{ccccccc}
P_3^k(N) & \text{Number of_Partitions} & \Delta_1 & \Delta_2 & \Delta_3 & \Delta_4 \\
0 & 1 & 5 & & & \\
1 & 6 & 10 & 9 & & \\
2 & 21 & 34 & 19 & 3 & \\
3 & 55 & 65 & 31 & 12 & 3 \\
4 & 120 & 111 & 46 & & \\
5 & 231 & & & & \\
\end{array}
$$

The differencing indicates a polynomial of degree 4 namely:

$$P_3^k(N) = \frac{1}{2 \cdot 4} (N+1)(N+2)[N(N+3) + 4]$$

Test runs for $6 < N < 12$ gave the predicted answer. Using the same method the following exact formula was obtained for the 4 by 4 matrix. (See Appendix 2)

$$P_4^k(N) = \frac{1}{6 \cdot 1890} (N+1)(N+2)(N+3)[N(N+4) + 4]$$

$$(N(N+4)[11N(N+4) + 155] + 840] + 1890]$$

Two features of the results for $P_3^k(N)$, $P_3^k(N)$ and $P_4^k(N)$ should be noted.

(1) If $m$ is the number of rows/columns then each formula is a polynomial of order $(m - 1)^2$. 
(2) Each formula, on graphing, has a symmetry about a negative integer, thus it would seem that for square matrices of side $m$ we should obtain a curve which passes through zero at $-1, -2, \ldots, -(m - 1)$. Hence we only need $0 \cdot 5(m - 2)(m - 1) + 1$ points on the curve to obtain a solution. But we already know that $P_3^k(N) = m!$ (a 1 can be inserted in the first row in $m$ different ways, the second row in $m - 1$ ways, etc.) so we only require 

$$0 \cdot 5(m - 2)(m - 1) + 1$$ points where $m \geq 3$ and $N \geq 2$. 

Thus, the knowledge that $P_3^2(2) = 21$ is, apparently, sufficient to have calculated $P_4^3(N)$ and to solve $P_4^4(N)$ only requires $P_3^2(2)$, $P_4^3(3)$ and $P_4^4(4)$.

To generalise these observations we predict $P_m^m(N)$ to be a polynomial of degree $(m - 1)^2$ of the following form (let $k = 0.5(m - 2)(m - 1) + 1$)

$$P_m^m(N) = \frac{1}{(m - 1)!k} (N + 1)(N + 2) \ldots$$

$$(N + m - 1)[N(N + m)][N(N + m)[ \ldots$$

$$[a_{11}N(N + m) + a_{12} + a_{21} + \ldots + a_{kk}]$$

Thus, to solve $P_3^2(N)$ we should require seven points on the curve and to verify the result we must, at least, calculate $P_3^2(8)$ but this will be of the order $8^{16}$ and is, therefore, impractical to attempt by the methods so far described.

**Partitioning 2 row matrices**

Slight modification of the program allows matrices of only 2 rows but any length to be calculated. Let $m$ be the number of columns, if $m$ is odd the conditions are

1. $\sum_{i=1}^{2} a_{ij} = 2N$
2. $\sum_{j=1}^{m} a_{ij} = mN$
3. $0 \leq a_{ij} \leq 2N$

and if $m$ is even the conditions are

4. $\sum_{i=1}^{2} a_{ij} = N$
5. $\sum_{j=1}^{m} a_{ij} = \frac{mN}{2}$
6. $0 \leq a_{ij} \leq N$

**Notation for 2 row matrices:**

The number of partitions of the integer $N$ in a matrix of $m$ columns and 2 rows is represented by $P_2^m(N)$.

The following results were obtained.

<table>
<thead>
<tr>
<th>$P_2^m(N)$</th>
<th>Starting Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_2^2(N)$</td>
<td>$N \quad 0$</td>
</tr>
<tr>
<td></td>
<td>$0 \quad N$</td>
</tr>
<tr>
<td>$P_2^3(N)$</td>
<td>$2N \quad N \quad 0$</td>
</tr>
<tr>
<td></td>
<td>$0 \quad N \quad 2N$</td>
</tr>
<tr>
<td>$P_2^4(N)$</td>
<td>$N \quad N \quad 0 \quad 0$</td>
</tr>
<tr>
<td></td>
<td>$0 \quad 0 \quad N \quad 2N$</td>
</tr>
<tr>
<td>$P_2^5(N)$</td>
<td>$2N \quad 2N \quad N \quad 0 \quad 0$</td>
</tr>
<tr>
<td></td>
<td>$0 \quad 0 \quad N \quad 2N \quad 2N$</td>
</tr>
<tr>
<td>$P_2^6(N)$</td>
<td>$N \quad N \quad N \quad 0 \quad 0 \quad 0$</td>
</tr>
<tr>
<td></td>
<td>$0 \quad 0 \quad 0 \quad N \quad N \quad N$</td>
</tr>
<tr>
<td>$P_2^8(N)$</td>
<td>$N \quad N \quad N \quad N \quad 0 \quad 0 \quad 0 \quad 0$</td>
</tr>
<tr>
<td></td>
<td>$0 \quad 0 \quad 0 \quad 0 \quad N \quad N \quad N \quad N$</td>
</tr>
</tbody>
</table>

**Partitioning in 3 dimensions**

Again, slight modification plus extension of the program allows 3 dimension cubic matrices to be calculated. Let $m$ be the number of rows, columns and layers, the conditions are:

1. $\sum_{i=1}^{m} a_{ijk} = \sum_{j=1}^{m} a_{ijk} = \sum_{k=1}^{m} a_{ijk} = N$
2. $0 \leq a_{ijk} \leq N$

**Notation for cubic matrix:**

The number of partitions of the integer $N$ in a matrix of $m$ columns, $m$ rows and $m$ layers is represented by $P_m^m(N)$.

For $m = 2$ we again note that the partition is completely defined by the contents of one cell hence $P_2^2(N)^3 = P_2^2(N) = N + 1$

For $m = 3$ the following exact formula was obtained (see Appendix 3)

$$P_3^3(N) = \frac{1}{4032} N(N + 1)[N(N + 1)][N(N + 1)[31N(N + 1) + 1004] + 6820] + 4272] + 1$$

This formula was the last obtained by the program, any other cases being beyond the scope of the methods so far applied. Like Shen Lin (1965) the simple approach via the computer has been exhausted but some idea of the properties of the solution has been obtained.

**A theoretical approach**

This section is in two parts. The first part deals with a particular case and is close to the way the program works; it is intended to introduce the more general and more efficient approach described in the second part.

**Formula**

$$N + 1$$

$$3 \cdot N(N + 1) + 1$$

$$\frac{1}{3}(N + 1)[2N(N + 2)] + 1$$

$$\frac{1}{12} N(N + 1)[115N(N + 1) + 70] + 1$$

$$\frac{1}{20} N(N + 1)[11N(N + 2) + 27] + 1$$

$$\frac{1}{5040} N(N + 2)[N(N + 2)[2472N(N + 2) + 7752] + 11616]] + 1$$
A particular case:

Consider the $P_2^m(N)$ case, i.e.,

$$
\begin{array}{ccc}
2N & 2N & 2N \\
3N & A & B \\
3N & & \\
\end{array}
$$

and we need only change elements $A$ and $B$ under the following restrictions

1. $0 < A < 2N$
2. $0 < B < 2N$
3. $N < A + B < 3N$

Let $f(x) = \sum_{A=0}^{2N} x^A = \frac{x^{2N+1} - 1}{x - 1}$ for $x \neq 1$

The number of combinations with $A + B = r$ (where $r$ is within the limits of (3)) is the coefficient $C_r$ of $x^r$ in $[f(x)]^2$

$$
[f(x)]^2 = \sum_{A,B} x^{A+B} = \sum_r \sum_{A+B=r} x^r = \sum_r C_r x^r
$$

Now $[f(x)]^2 = \left( \frac{x^{2N+1} - 1}{x - 1} \right)^2 = \frac{x^{4N+2} - 2x^{2N+1} + 1}{(x - 1)^2}$

Breaking this into three parts

1. $\frac{1}{(1 - x)^2} = 1 + 2x + 3x^2 + \ldots + (r + 1)x^r$
2. $\frac{x^{2N-1}}{(1-x)^2} = x^{2N+1} + 2x^{2N+2} + 3x^{2N+3} + \ldots$
3. $\frac{x^{4N-2}}{(1-x)^2} = x^{4N+2} + 2x^{4N+3} + 3x^{4N+4} + \ldots$

Hence if $0 < r < 2N$ the coefficient of $x^r$ is $(r + 1)$. And if $2N + 1 < r < 4N + 1$ the coefficient of $x^r$ is $(r + 1) - 2(r - 2N) = 4N + 1 - r$. Hence number of solutions is

$$
\sum_{r=0}^{3N} C_r = \sum_{r=0}^{2N-1} (r + 1) + (2N + 1) + \sum_{r=2N+1}^{3N} (4N + 1 - r)
$$

$$
= 2 \sum_{r=0}^{2N-1} (r + 1) + 2N + 1
$$

$$
= 3N^2 + 3N + 1
$$

which is in agreement with the solution obtained by the machine.

The general case:

Now let us consider the more general case of $P_2^m(N)$.

<table>
<thead>
<tr>
<th>$mN$</th>
<th>$2N$</th>
<th>$2N$</th>
<th>$\ldots$</th>
<th>$2N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$a_3$</td>
</tr>
</tbody>
</table>

where

1. $0 < a_1 < 2N$
2. $0 < a_2 < 2N$
3. $\ldots$
4. $0 < a_m < 2N$

and

$$
\sum_{i=1}^{m} a_i = mN
$$

Consider, as before,

$$
f(x) = 1 + x + x^2 + \ldots + x^{2N} = \frac{x^{2N+1} - 1}{x - 1}
$$

Number of solutions is the coefficient of $x^{mN}$ in

$$
\left[ \frac{x^{2N+1} - 1}{x - 1} \right] = \sum_{s=0}^{m} \binom{m}{s} (-1)^s x^{(2N+1)s} \frac{1}{(1-x)^s}
$$

now

$$
\frac{1}{(1-x)^m} = \sum_{p=0}^{\left\lfloor \frac{m}{r} \right\rfloor} \binom{-m}{p} x^p (-1)^p
$$

Thus need $2N + 1 + s + p = mN$

or $p = mN - (2N + 1)s$

Hence the coefficient is

$$
\sum \binom{m}{s} \left( \frac{mN - (2N + 1)s - 1}{mN - (2N + 1)s} \right) (-1)^p
$$

The terms for which $(2N + 1)s > mN$ must be counted as zero. Note that if $m$ is even we read $2N = N$ to obtain the machine answers.

$$
m = 1 \quad C_1 = \binom{N}{N} = 1
$$

$$
m = 2 \quad C_2 = \left\{ \begin{align*}
0 & \quad 2N + 1 \\
2N & \quad 2N + 1
\end{align*} \right\} = 2N + 1 = N + 1
$$

$$
m = 3 \quad C_3 = \left\{ \begin{align*}
0 & \quad 3N + 2 \\
2 & \quad 3N + 2
\end{align*} \right\} - \left\{ \begin{align*}
0 & \quad 1 \\
2 & \quad 1
\end{align*} \right\} \left( \begin{align*}
N + 1 & \quad N + 1 \\
1 & \quad 1
\end{align*} \right) = 3N^2 + 3N + 1
$$

$$
m = 4 \quad C_4 = \left\{ \begin{align*}
0 & \quad 4N + 3 \\
3 & \quad 4N + 3
\end{align*} \right\} + \left\{ \begin{align*}
0 & \quad 2N + 2 \\
1 & \quad 2N + 2
\end{align*} \right\} + \left\{ \begin{align*}
0 & \quad 1 \\
2 & \quad 1
\end{align*} \right\} \left( \begin{align*}
1 & \quad 1 \\
3 & \quad 3
\end{align*} \right) \left( \begin{align*}
N + 1 & \quad N + 1 \\
2 & \quad 2
\end{align*} \right) = 1/4(2N + 1)(8N^2 + 8N + 3) = 1/4(N + 1)(2N + 4N + 3)

These formulae were derived as far as $m = 8$ and in all cases complete agreement was found. This verifies, axiomatically, the algorithms given in Appendix I.

To further generalise this procedure an attempt to solve the general two dimensional case was made. The approach is as follows.

We wish to find the number of arrangements of integers $a_{ij}$ in a matrix of $m$ columns and $n$ rows such that
in the hope that someone may attempt to prove or disprove it.

Acknowledgements

The author would like to thank Dr. J. C. P. Miller and Dosent Helge Tverberg of Matematisk Institutet, Universitetet i Bergen, Bergen, Norway, for pointing out that the 2-dimensional matrices are of a well-known type, namely the DOUBLY STOCHASTIC matrices of Combinatorial Analysis and discussing the work of Ryser (1963) in this field.

Appendix 1

ALGOL program to count the number of partitions of the integer \(N\) in a square matrix side \(m\)


\[
\text{begin integer } i, j, m, N, \text{ count, startingrow, upto; ENTRY: } m := \text{ read; } N := \text{ read; }
\]

\[
\text{begin integer array sumcol, sumrow } [1:m], A [1:m, 1:m];
\]

\[
\text{for } i := 1 \text{ step 1 until } m \text{ do}
\]

\[\text{sumcol}[i] := \text{sumrow}[i] := N; \text{ count := 0; startingrow := 1; upto := } m; \]

C1: \[
\text{BASE\textsc{ic} \text{ OPERATION: for } i := \text{startingrow} \text{ step 1 until } m \text{ do}
\]

\[
\text{begin for } j := 1 \text{ step 1 until } \text{upto do begin}
\]

\[
\text{A}[i, j] := \text{if sumcol}[j] < \text{sumrow}[i] \text{ then}
\]

\[
\text{sumcol}[j] := \text{sumrow}[i] - \text{A}[i, j];
\]

\[
\text{count := count + 1;}
\]

C2: \[
\text{BACK \textsc{track: for } i := m \text{ step } -1 \text{ until } 1 \text{ do}
\]

\[
\text{for } j := 1 \text{ step 1 until } m \text{ do begin}
\]

\[
\text{sumrow}[i] := \text{sumrow}[i] + \text{A}[i, j];
\]

\[
\text{sumcol}[j] := \text{sumcol}[j] + \text{A}[i, j];
\]

C3: \[
\text{if } A[i, j] + 1 < (\text{if sumcol}[j] < \text{sumrow}[i] \text{ then}
\]

\[
\text{sumcol}[j] \text{ else sumrow}[i]) \text{ then begin}
\]

\[
\text{A}[i, j] := A[i, j] + 1;
\]

C4: \[
\text{sumrow}[i] := \text{sumrow}[i] - \text{A}[i, j];
\]

\[
\text{sumcol}[j] := \text{sumcol}[j] - \text{A}[i, j];
\]

C5: \[
\text{upto := } j - 1; \text{ startingrow := } i;
\]

C6: \[
\text{go to \textsc{bas\textsc{ic} \text{ operation;}}}
\]

C7: \[
\text{print(count,5,0)}
\]

C8: \[
\text{end}
\]

Conclusion

This paper presents nice results, arrived at in a simple way, and verified by machine for a few simple cases. Its main purpose is to introduce the conjecture given above.
Appendix 2

In the case of \( P_2^0(N) \) the computer obtained the following:

<table>
<thead>
<tr>
<th>( P_2^0(N) )</th>
<th>Number of Partitions</th>
<th>( \Delta_2 )</th>
<th>( \Delta_4 )</th>
<th>( \Delta_6 )</th>
<th>( \Delta_8 )</th>
<th>( \Delta_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_2^0(0) )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_2^0(1) )</td>
<td>24</td>
<td>235</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_2^0(2) )</td>
<td>282</td>
<td>1468</td>
<td>3712</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_2^0(3) )</td>
<td>2008</td>
<td>6413</td>
<td>10532</td>
<td>7800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_2^0(4) )</td>
<td>10147</td>
<td>21890</td>
<td>25152</td>
<td>13232</td>
<td>2112</td>
<td></td>
</tr>
<tr>
<td>( P_2^0(5) )</td>
<td>40176</td>
<td>62519</td>
<td>53004</td>
<td>20776</td>
<td>2464</td>
<td>352</td>
</tr>
<tr>
<td>( P_2^0(6) )</td>
<td>132724</td>
<td>156152</td>
<td>101632</td>
<td>30784</td>
<td>2816</td>
<td></td>
</tr>
<tr>
<td>( P_2^0(7) )</td>
<td>381424</td>
<td>351417</td>
<td>181044</td>
<td>43608</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_2^0(8) )</td>
<td>981541</td>
<td>727726</td>
<td>304064</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_2^0(9) )</td>
<td>2309384</td>
<td>1408099</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_2^0(10) )</td>
<td>5045326</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Appendix 3

In the case of \( P_3^0(N) \), the computer obtained the following:

<table>
<thead>
<tr>
<th>( P_3^0(N) )</th>
<th>Number of Partitions</th>
<th>( \Delta_2 )</th>
<th>( \Delta_4 )</th>
<th>( \Delta_6 )</th>
<th>( \Delta_8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_3^0(0) )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_3^0(1) )</td>
<td>12</td>
<td>109</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_3^0(2) )</td>
<td>132</td>
<td>595</td>
<td>1278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_3^0(3) )</td>
<td>847</td>
<td>2359</td>
<td>3388</td>
<td>2150</td>
<td></td>
</tr>
<tr>
<td>( P_3^0(4) )</td>
<td>3921</td>
<td>7511</td>
<td>7648</td>
<td>3390</td>
<td>310</td>
</tr>
<tr>
<td>( P_3^0(5) )</td>
<td>14506</td>
<td>20311</td>
<td>15298</td>
<td>4940</td>
<td>310</td>
</tr>
<tr>
<td>( P_3^0(6) )</td>
<td>45402</td>
<td>48409</td>
<td>27888</td>
<td>6800</td>
<td></td>
</tr>
<tr>
<td>( P_3^0(7) )</td>
<td>124707</td>
<td>104395</td>
<td>47278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_3^0(8) )</td>
<td>308407</td>
<td>207659</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P_3^0(9) )</td>
<td>699766</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

References
