This Notebook uses recurrence relations to compute the table in Figure 6 of “Invariant Synchrony and Anti-Synchrony Subspaces of Weighted Networks” by Nijholt, Sieben and Swift.

The first part of a sequence, for example $(e_n) = (e_0, e_1, e_2, \ldots)$, is represented as a list with the name of the sequence. Thus, $(e_0, e_1, e_2, \ldots, e_n) \longleftrightarrow$ e = {e $_0, e_1, e_2, \ldots, e_n$}. Note that in Mathematica the first element of the list is e[[1]] = e_0, and the last element of the list is e[[n+1]] = e_n.

These computations use the recurrence relations of Proposition 7.20, implemented using Remark 7.21. This uses very low-level functions compared the the EGF method. The recurrence method could be programmed in a language like C++ without a computer algebra system.

In[∘]:=
```
nMax = 20; (* A global variable used in
 all the cells.  So this cell must be run first *)
```

In[∘]:=
```
p = {1}; (* initialize with p = {p0} = {1} *)
(* pn = p〚-1〛 is the last element of
  the list p. Compute pn+1 and append it to the list p. *)
For[n = 0, n < nMax, n++, AppendTo[p,
```

$$\mathtt{p[\![-1]\!]} + \sum_{k=0}^{n-1} \sum_{l=0}^{n-k-1} \frac{n!}{k!\, l!\, (n-k-l)!}\, \mathtt{p[\![k+1]\!]} + \sum_{k=0}^{n} \frac{n!}{k!\, (n-k)!}\, \mathtt{p[\![k+1]\!]}\Big]\Big];$$

```
Print["The sequence (pn) of the number of polydiagonal subspaces of Rn is ", p];
```

The sequence (p_n) of the number of polydiagonal subspaces of R^n is
{1, 2, 6, 24, 116, 648, 4088, 28 640, 219 920, 1 832 224, 16 430 176, 157 554 048,
1 606 879 040, 17 350 255 744, 197 553 645 440, 2 363 935 624 704, 29 638 547 505 408,
388 328 781 668 864, 5 304 452 565 517 824, 75 381 218 537 805 824, 1 112 348 880 749 130 752}

In[∘]:=
```
s = {1};
(* initialize with s = {s0} = {1} *)
(* Compute sn+1 and append it to the list s. *)
```

$$\mathtt{For}\Big[n = 0,\ n < \mathtt{nMax},\ n{+}{+},\ \mathtt{AppendTo}\Big[s,\ \sum_{k=0}^{n} \frac{n!}{k!\, (n-k)!}\, \mathtt{s[\![k+1]\!]}\Big]\Big];$$

```
Print["The sequence (sn) of the number of synchrony subspaces of Rn is ", s];
```

The sequence (s_n) of the number of synchrony subspaces of R^n is
{1, 1, 2, 5, 15, 52, 203, 877, 4140, 21 147, 115 975, 678 570, 4 213 597, 27 644 437, 190 899 322,
1 382 958 545, 10 480 142 147, 82 864 869 804, 682 076 806 159, 5 832 742 205 057, 51 724 158 235 372}

To get the number of anti-synchrony subspaces use the fact that $a_n = p_n - s_n$

In[∘]:=
```
Print["The sequence (an) of the number of anti-synchrony subspaces of Rn is ", a = p - s]
```

The sequence (a_n) of the number of anti-synchrony subspaces of R^n is
{0, 1, 4, 19, 101, 596, 3885, 27 763, 215 780, 1 811 077, 16 314 201, 156 875 478,
1 602 665 443, 17 322 611 307, 197 362 746 118, 2 362 552 666 159, 29 628 067 363 261,
388 245 916 799 060, 5 303 770 488 711 665, 75 375 385 795 600 767, 1 112 297 156 590 895 380}

To get the minimally tagged subspaces use the fact that $m_n = s_{n+1} - s_n$. Make a list of s_{n+1} starting with n = 0 by first making a list of (s_n) with nMax+1 terms, and then dropping the first term.

In[∘]:=
```
snp1 = {1};
(* initialize with snp1 = {s0} = {1} *)
(* Compute sn+1 and append it to the list snp1. *)
For[n = 0, n < nMax + 1, n++, AppendTo[snp1, Sum[n! / (k! (n - k)!) snp1[[k + 1]], {k, 0, n}]]];
Print["The sequence (mn) of the number of minimally tagged subspaces of R^n is ",
  m = Drop[snp1, 1] - s];
```

$$\text{For}\left[n = 0,\ n < \text{nMax} + 1,\ n{+}{+},\ \text{AppendTo}\left[\text{snp1},\ \sum_{k=0}^{n} \frac{n!}{k!\,(n-k)!}\,\text{snp1}[[k+1]]\right]\right];$$

The sequence (m_n) of the number of minimally tagged subspaces of R^n is
{0, 1, 3, 10, 37, 151, 674, 3263, 17 007, 94 828, 562 595, 3 535 027, 23 430 840, 163 254 885, 1 192 059 223, 9 097 183 602, 72 384 727 657, 599 211 936 355, 5 150 665 398 898, 45 891 416 030 315, 423 145 657 921 379}

In[∘]:=
```
f = {1};
(* initialize with f = {f0} = {1} *)
(* Compute fn+1 and append it to the list f. *)
For[n = 0, n < nMax, n++, AppendTo[f, f[[-1]] + Sum[n! / (k! l! (n - k - l)!) f[[k + 1]], {k, 0, n - 1}, {l, 0, n - k - 1}]]];
Print["The sequence (fn) of the number of synchrony subspaces of R^n is \n", f];
```

$$\text{For}\left[n = 0,\ n < \text{nMax},\ n{+}{+},\ \text{AppendTo}\left[f,\ f[[-1]] + \sum_{k=0}^{n-1}\sum_{l=0}^{n-k-1} \frac{n!}{k!\,l!\,(n-k-l)!}\,f[[k+1]]\right]\right];$$

The sequence (f_n) of the number of synchrony subspaces of R^n is
{1, 1, 2, 7, 29, 136, 737, 4537, 30 914, 229 831, 1 850 717, 16 036 912, 148 573 889, 1 463 520 241, 15 259 826 402, 167 789 512 807, 1 939 125 333 629, 23 484 982 837 816, 297 289 975 208 417, 3 924 325 664 733 097, 53 906 145 745 657 634}

In[∘]:=
```
e = {1}; (*initialize with e = {e0} = {1} *)
(* en = e[[-1]] is the last element of the list e. Compute
   en+1 and append it to the list e. Note that en-2l-1 = e[[n-2l]]*)
For[n = 0, n < nMax, n++, AppendTo[e,
   e[[-1]] + Sum[n! / ((n - 2 l - 1)! l! (l + 1)!) e[[n - 2 l]], {l, 0, Floor[(n-1)/2]}]]];
Print["The sequence (en) of the number of evenly tagged
    polydiagonal subspaces of R^n is ", e];
```

$$e[[-1]] + \sum_{l=0}^{\text{Floor}[(n-1)/2]} \frac{n!}{(n-2l-1)!\,l!\,(l+1)!}\,e[[n-2l]]$$

The sequence (e_n) of the number of evenly tagged polydiagonal subspaces of R^n is
{1, 1, 2, 4, 13, 41, 176, 722, 3774, 18 958, 116 302, 687 182, 4 812 226, 32 541 874, 255 274 826, 1 938 568 634, 16 798 483 589, 141 220 228 149, 1 337 121 257 864, 12 305 678 519 102, 126 208 299 343 263}

In[∘]:=
```
tableRecurrence = {Range[0, nMax], p, s, a, m, f, e};
TableForm[tableRecurrence,
 TableHeadings → {{"n", "pn", "sn", "an", "mn", "fn", "en"}, None}]
```

Out[∘]//TableForm=

n	0	1	2	3	4	5	6	7	8	9	10
p_n	1	2	6	24	116	648	4088	28 640	219 920	1 832 224	16 430 176
s_n	1	1	2	5	15	52	203	877	4140	21 147	115 975
a_n	0	1	4	19	101	596	3885	27 763	215 780	1 811 077	16 314 201
m_n	0	1	3	10	37	151	674	3263	17 007	94 828	562 595
f_n	1	1	2	7	29	136	737	4537	30 914	229 831	1 850 717
e_n	1	1	2	4	13	41	176	722	3774	18 958	116 302

Run the Sequences-EGF.nb notebook first (select all then run), and then run this Sequences-recurrence.nb notebook and the next line should return “True” indicating that both methods give the same results.

```
tableRecurrence == tableEGF
```

Out[ ]= `True`

Note that $s_n = e_n$ for n = 0, 1, 2, and $s_n > e_n$ for $3 \le n \le 9$, and $s_n < e_n$ for $n \ge 10$. (We have not proved this last observation.)

That was easy to program, but it hides a computation done by the computer algebra system. See the notebook sequences-recurrence.nb for a more elementary method of computing these sequences.