The structure of some good codes

Jürgen Bierbrauer
Department of Mathematical Sciences
Michigan Technological University
Houghton, Michigan 49931 (USA)
Yves Edel
Mathematisches Institut der Universität
Im Neuenheimer Feld 288
69120 Heidelberg (Germany)

September 25, 1996

Abstract

We present more detailed information on the structure of some good linear codes as constructed by us in other papers.

1 The binary case

1.1 Binary code $[155, 132, 8]$

This was constructed in [1].

1.1.1 Check matrix of binary code $[154, 132, 7]$

```
10000000000000000000001100101111110010100110111110100110011111110011011010111100
0100000000000000000000011100110001011111111101010110001100001110011100100011100
00100000000000000000000111001100010111111111010101100011000011100111001000111000
000100000000000000000001110011000101111111110101011000110000111001110010001110001
00001000000000000000000100100000100100001011001010110011010111000100100110010011 0
000001000000000000000000111110010000111000111101000100111111100000111110110001001
0000001000000000000000001111100100001110001111010001001111111000001111101100010011
000000010000000000000001010111000001011110101100001110101010000000101011011110001 0
0000000010000000000000000111001100010111111111010101100011000011100111001000111
000000000100000000000000001110011000101111111110101011000110000111001110010001
00000000001000000000000001110011000101111111110101011000110000111001110010 0011
0000000000010000000000001110011000101111111110101011000110000111001110010001110
00000000000010000000000000111001100010111111111010101100011000011100111001000
00000000000001000000001111100101111101001010101110111010100000111111010101100001
00000000000000100000001010111011101101111110011110011111100001101011000100000110
00000000000000010000000011111001011111010010101011101110101000001111110101011000
00000000000000001000000111110010111110100101010111011101010000011111101010110000
000000000000000001000001011000100100100011100110100000001000010000110100110100111
000000000000000000100010110001001001000111001101000000010000100001101001101001110
000000000000000000010000101100010010010001110011010000000100001000011010011010011
0000000000000000000001010111000001011110101100001110101010000000101011011110001011
000000000000000000000010101110000010111101011000011101010100000001010110111100010 1
```

```
100001000011100100011011000000001100000001011110110001110000010000011101101
0111010001100100011011011001111001110111001000001000100000101100000001101001
11101000110010001101101100111100111011100100000001101001010010111010101001010
1101000110010001101101100111100111011100100000000000110011110101011100000111
00101111011010000100000111110001001110001100011111001101010101100010011000
11010010100110111010111011100000111100000100100110110100111100101010110100
10100101001101110101110111000001111000001001001001110101011100110000111111
11000110001001011001011010000000101000000011100011011000010000011111111110000
000111010001100100011011011001111001110111001000011110000001111110011010111
11000111010001100100011011011001111001110111001000101000011101000000001111
10001110100011001000110110110011110011101110010011000110100110110010111101
00111010001100100011011011001111001110111001000011001101111000111000110100
11100011101000110010001101101100111100111011100111000001010011111010001010
10011101110111111001010100011010110111000111110010110101110000111010101011
1011011111110100000001110011011100111001001111101111001011101011001101010
0110011101110111111001010100011010110111000111111101110010110100110011011
110011101110111111001010100011010110111000111101011110011010011101110100
01111010110011110011001000010001000011011011011011001101111000110110110001
11110101100111100110010000100010000110110110110010001000010001110001011101
10111101011001111001100100001000100001101101101101111000100101100101011111
0001100010010110010110100000001010000000111000111010100111010011100000011011
1000110001001011001011010000000101000000011100011100111100011110000111110111
```

1.2 Binary code $[72, 46, 10]$

This was constructed in [1].

1.2.1 Check matrix of binary code $[71, 46, 9]$

```
10000000000000000000000001001100110101110101010101000110110110011000000
01000000000000000000000001111011111011111001000000011110001101010010110
00100000000000000000000001000101010011011011111111101110000000010101110
00010000000000000000000001000101010011011011111111101110000000100111100
00001000000000000000000001011111111000101000000000001011101101111011001
00000100000000000000000001101010101111001111111111100101101101010110101
00000010000000000000000001010111010010110011011111101010101011111101101
00000001000000000000000001000001000110010110010000010000111000101001001
00000000100000000000000001111010111110101010011011100001100011010110010
00000000010000000000000001000001000110010110010000010000111000110000011
00000000001000000000000001010111110000011010110010010101011100111000100
00000000000100000000000001000001010111000010100110101111000011000101011
00000000000010000000000001000001010111000010100110101111000011000001101
00000000000001000000000001000001010111000010100110101111000011111010000
00000000000000100000000001100010100101110111101011010000010101001010011
00000000000000010000000001100010100101110111101011010000010101001100000
00000000000000001000000001100010100101110111101011010000010101100000001
00000000000000000100000001100110111011100001000010111111101111111010010
00000000000000000010000001011001110010111111011110111010011010101001000
00000000000000000001000001011001110010111111011110111010011010111011100
00000000000000000000100001111111100010100000000000101110110111101100010
00000000000000000000010001010100110110111111111011100000000100010110110
00000000000000000000001000101010011011011111111101110000000100011011101
00000000000000000000000101111100110011100000000100000101100101111101001
00000000000000000000000011010101011110011111111111001011011010110110000
```

1.3 Binary code $[39, 12, 14]$

This was constructed in [1].

1.3.1 Generator matrix of binary $[38, 12, 13]$

```
11110010100000111110110100110000000000
00011011100110110011011101010000000000
11010111110110001100000111001000000000
01011001111101001011101010000100000000
00101100111100011101110101000010000000
10110100001111101011010011000001000000
10001001010110000111001110000000100000
00101111011011001011110110000000010000
01111101000100001000110011000000001000
10010001111111011101011001000000000100
01101000100011100110001101000000000010
10101010010110111001111011000000000001
```

2 The ternary case

2.1 Ternary $[86, 77, 5]$

This was constructed in [2]. The weight distribution that the dual is a $[86, 9, 50]$.

2.1.1 Check matrix of ternary code $[86, 77, 5]$

11111111111111111111111111111111111111111111111111111111111111111111111111111111
01220112020112001212001220120112001212001220101220112012001220101220112020112001
01201201201201201201201201201201201201201201201201201201201201201201201201201201
02102102102102102102102102121021021021021021021021010210210210210210210210210210
01201201220120120112012012001201201220120120112012012001201201220120120112012012
01121222121222101122101121221222101100212202020010111022101121220010111002000212
01101101120020020020020020001112220012220001101112220001120012201120012212201120
02202202221102210021110002200111222000122011211211211201020212112112112101012120
01120020002021221200222122101120020021210110111000200201120020010102002022111011

100000
200000
200000
200000
000000
201001
011000
200101
000011

2.1.2 Weight distribution of ternary $[86, 77, 5]$ and its dual

i	$A_i(\mathcal{C})$	$A_i(\mathcal{C}^\perp)$
0	1	1
1	0	0
2	0	0
3	0	0
4	0	0
5	59510	0
6	1609578	0
7	34656012	0
8	689688702	0
9	11960376480	0
10	184259398914	0
11	2545287155388	0
12	31819230576450	0
13	362242121685480	0
14	3777692766008058	0
15	36265754744822376	0
16	321858811617661080	0
17	2650601709385493940	0
18	20321279506224703440	0
19	145457581829047955064	0
20	974565789816813629868	0
21	6125842132925716048770	0
22	36198158002919611059690	0
23	201450618524347733305620	0
24	1057615747246660911603642	0
25	5245774106025267140407704	0
26	24614786191062740165753526	0
27	109399049734983558258823620	0
28	461038852459476376941280950	0
29	1844155409834101222446588396	0
30	7007790557365615792557533178	0
31	25318469110504584536147880288	0
32	87032237567298589874411890950	0
33	284832777493109388926731857570	0
34	888008071007720217888245047608	0

i	$A_i(\mathcal{C})$	$A_i(\mathcal{C}^\perp)$
35	2638652553851681224694343081168	0
36	7476182235913243680875095379712	0
37	20205897934899900619228049526690	0
38	52109947305795986735243007224430	0
39	128270639521957001493791909880324	0
40	301436002876602372191506263906918	0
41	676392982064566977062788276696740	0
42	1449413532995503486940878900290390	0
43	2966241648921033117870030391695540	0
44	5797654131982007115036660365595702	0
45	10822287713033102865136618756242744	0
46	19291904184102454291708355350325310	0
47	32837283717621243500319893732244600	0
48	53360586041134463600894994067189290	0
49	82763357941351480130950554951008886	0
50	122489769753200119674606173678718468	486
51	172926733769223766194998269980294696	324
52	232785987766262702256131689429655100	1944
53	298668814492563515101389948383681790	2754
54	365039662157577595734911405388930558	2832
55	424773425056090316212035297512314380	972
56	470284863454957115392494192302608782	648
57	495036698373639090724675755557868624	0
58	495036698373639073334117308107135114	0
59	469865340829216751969386199647305564	972
60	422878806746295084295219758272706378	648
61	360486851652579407675217643587766284	3888
62	290715202945628554678984926413872110	2106
63	221497297482383667986223886484829840	1296
64	159201182565463254207681861772133868	486
65	107766954352005897062544787545311826	324
66	68578970951276482522082597944504020	0
67	40942669224642672624233111781131520	0
68	22879726919653260313943347602678120	0
69	11937248827645178881571154359637384	0
70	5798092287713372057283563414910528	0

i	$A_i(\mathcal{C})$	$A_i(\mathcal{C}^\perp)$
71	2613224693053915018327819436792400	0
72	1088843622105797114369348245445460	0
73	417638649574826894876425483736340	0
74	146737903904668553596296401772072	0
75	46956129249494091599235471496176	0
76	13592563730116670410657530960456	0
77	3530536033796535367967206161960	0
78	814739084722286418541253055840	0
79	165010447538685475569339082416	0
80	28876828319272046023932189078	0
81	4278048639891575584409516294	2
82	521713248767384596047493524	0
83	50285614339007066619226224	0
84	3591829595645243535218688	0
85	169027275089064619839456	0
86	3930866862540204920128	0

2.2 Ternary $[39, 29, 6]$

This was constructed in [1].

2.2.1 Check matrix of ternary $[39, 29, 6]$

111111111111111111111111111000000000000
000222111111000222222111000000000000000
021210102102021210210102021000000000000
000222111000222111000222111000000000000
000120102021222012012021222100012100101
022112121121100001112010100010001100221
000012021222201210222201210001001120102
000012021210222201201210222000101010212
022100100010121121001112112000011001122
000210201102120222120222102000000111111

2.2.2 Weight distribution of ternary $[39, 29, 6]$ and its dual

i	$A_i(\mathcal{C})$	$A_i(\mathcal{C}^\perp)$
0	1	1
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	4890	0
7	31914	0
8	267138	0
9	1840352	0
10	10981332	0
11	58280796	0
12	271158990	0
13	1126437678	0
14	4186409346	0
15	13951007586	6

i	$A_i(\mathcal{C})$	$A_i(\mathcal{C}^\perp)$
16	41852441610	0
17	113255943120	0
18	276840452490	564
19	611933642406	0
20	1223975987946	0
21	2214718693866	6498
22	3623877612828	0
23	5357628686268	0
24	7143083583948	17394
25	8571313298862	0
26	9231529921038	0
27	8889230340014	23186
28	7618909422960	0
29	5780477075472	0
30	3853430272944	10512
31	2237379475392	0
32	1118797997112	0
33	474619134552	744
34	167502631968	0
35	47862526752	0
36	10636657552	144
37	1724216544	0
38	181621440	0
39	9307776	0

2.3 Ternary code $[85, 74, 6]$

Construction: the dual of the primitive A-code $[81, 11, 45]$ has parameters $[81, 70, 6]$ and possesses a 4-step extension, found by computer.

2.3.1 Check matrix of ternary $[85, 74, 6]$

10000000000110002001002112122012111101112022112020102221102101002111202111020122
01000000000121102100120102110210200112110100012210001021221221000212002102201210
00100000000120221010002211011210200112100201022210001022212211000202102102201 21
00010000000200011122012222002211101012000200210212211000120201101001021100112220
00001000000210211211100122111220021211100202011221120202122211011101002020101001
00000100000000211121002001212012101001122022000222200012011221120002101021022012
00000010000101122102201110022202112121112010110110212000021010021000022002020011
00000001000112221022012121121020111200111022122102110200210220201202011100120111
00000000100011211210212101112210012120012122021210111022001210010220011110012011
00000000010101101110101222122200201200002020211212100100121200200022110011222011
00000000001012002201122021220221020120002201002121110011001021010001111001122201

12110
01210
02211
10210
12102
12112
10012
00121
11020
02121
10220

2.4 Ternary code $[116, 102, 6]$

2.4.1 Check matrix of ternary code $[116, 102, 6]$

Omitting the last row and the last 13 columns produces a check matrix of $[103, 90, 6]$. A similar procedure leads to a check matrix of $[95, 83, 6]$. The construction is in [4]

```
10000000000110002001002112122012111101112022112020102221102101002111202111020122
01000000000121102100120102110210200112111010001221000102122122100021200210220121 0
00100000000120221010002211011210200112100201022210001022212211000202102102201 21
00010000000200011122012222002211101012000200210212211000120201101001021100112220
00001000000210211211100122111220021211100202011221120202122211011101002020101001
00000100000000211121002001212012101001122022000222200012011221120002101021022012
00000010000101122102201110022202112121112010110110212000021010021000022002020011
00000001000112221022012121121020111200111022122102110200210220201202011100120111
00000000100011211210212101112210012120012122021210111022001210010220011110012011
00000000010101101110101222122200201200002020211212100100121200200022110011222011
00000000001012002201122021220221020120002201002121110011001021010001111001122201
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
121100222121220021121200212201011121
012100012201210021011020100100212012
022110001111022002121200121112110020
102100100110002012200120011110001212
121020120221201011111210020112202020
121120100022100002001010100121020122
100120010212110002102000000002010202
001210001002010000000010010102000012
110200000210202001100010010001000001
021210000001101001100020010000000000
102200000011122000100000000000000000
000001111111111000011220001122001122
000000000000000111111110000000111111
000000000000000000000001111111111111
```

2.5 Ternary code $[25, 14, 7]$

This was found purely experimentally by Yves Edel, as an *extension of nothing.*

2.5.1 Check matrix of ternary code $[25, 14, 7]$

```
1000000000012012201210010
0100000000011222212111112
0010000000011110220021101
0001000000011002121022001
0000100000010210111212201
0000010000010101012200211
0000001000001210120010121
0000000100001101000101012
0000000010000011001100121
0000000001000000111100001
0000000000100000000011111
```

2.6 Ternary code $[33, 21, 7]$

This was constructed in [2] as a 7-step extension of the primitive BCH-code $(n = 3, w = 26)$ with parameters $[26, 14, 7]$.

2.6.1 Check matrix of ternary code $[33, 21, 7]$

```
010000000000212121102002211201221
100000000000112121011201011111121
001000000000102121002121220000001
000100000000222000202211000010021
000010000000002220002022111011120
000001000000022200020221102100220
000000100000001211012012000201210
000000010000012110120120002020201
000000001000121101201200000122101
000000000100000121101201202200012
000000000010000012110120120021000
000000000001212122010011220011110
```

2.7 Ternary $[52, 38, 7]$ and $[42, 29, 7]$

2.7.1 Check matrix of ternary code $[52, 38, 7]$

If the last row and the 10 last columns are deleted, we obtain a check matrix of $[42, 29, 7]$. This is in [4].

0202122102220010121120111000000020202122000100010212
2021221022200101211201110000010010111012210221121102
0212210222001012112011100200020020011021210002202110
2221220021010222122002101002002100120100110100210121
0120201112110012020111211021202100001112010011222010
2212200210102221220021010201101200000021010012211012
2120112220200212011222020010200000000002210000020012
1001210221110100121022111001001200000000120000112000
2202002120112220200212011200100000000002020000121010
1211120011010212221002202000012210000022210000002001
2202012111200110102122210000001120000000200000000012
2210022020121112001101021200001110000001220000000002
0000000000000000000000000000000001111111110000000112
0000000000000000000000000000000000000000001111111111

2.8 Ternary code $[57, 42, 7]$

This was constructed in [2] as 16-step extension of the A-code $[41, 26, 7]$ corresponding to the BCH-code $[40, 26, 6]$.

2.8.1 Check matrix of ternary $[57, 42, 7]$

```
100000000000000120021022222212120221210021102221110021220
010000000000000122121112000020002022001220121121112220200
001000000000000120201001200020211101022111220021222220221
000100000000000111022220000021211202122021010002111010011
000010000000000201001210120011110211010220202122221000112
000001000000000101122011012022022220220011001112112020121
000000100000000212220111110100221200201012011020011101100
000000010000000202120011101211021020200010101201000201200
000000001000000210200121111022111221201111102011221122111
000000000100000210001122111111000220001122200211110002000
000000000010000102022002211102011221122100011110120120210
000000000001000220001000022120112210122201200200210211200
000000000000100010202200221110201122112210111201211221222
000000000000010022000100002212011221012221202022021210002
000000000000001110222200000212112021220211011120200010002
```

2.9 Ternary code $[88, 72, 7]$

This was constructed in [2]

2.9.1 Check matrix of ternary $[88, 72, 7]$

10000000000000002001201220101020110121021111110222122000220210210000102020012122
01000000000000001222212121010012102100022101220222102121010102021100100001002012
00100000000000001012221020100010020110110022111220112121020020020202220212112001
00010000000000002212010011000022200211022201021200212121122200011220122020212221
00001000000000002112100100010121211220121012102222121021111221002220102222210100
00000100000000002112100100010121211220121012102222121021111221002220102222221010
00000010000000001120102122100021221000201220021202010000202211210112101112010022
00000001000000000211210010001012121122012101210222212102111122100222010222222101
00000000100000000202102022021002102201101202002002011012111212110220100210111022
00000000010000002021020220210021022011012020020020110121112121102201002101110220
00000000001000002202220210202122110101102021221100110222000021222212102011220020
00000000000100000220222021020212211010110202122110011022200002122221210201122002
00000000000010001222102010001002011011002211122011212102002002020222021211200101
00000000000001001011120210010212112220211121102111212201212020011121102011120221
00000000000000102201002210102010121012000221200111210220012200220112210010021121
000000000000000112102002200001110012201110201210012121222111000221102110101211122

02101021
11111121
12202001
12200122
22002020
21200010
11010222
22021110
01001000
02021221
11202112
00010022
02111201
12121211
12000111
21000110

2.10 Ternary $[108, 91, 7]$

2.10.1 Check matrix of ternary code $[108, 91, 7]$

We start from the ternary code $[85, 70, 7]$ and obtain the code given in the title. If the last row and the 16 last columns are deleted, we obtain a check matrix of $[92, 76, 7]$. It should be clear that row i of the check matrix consists of row i of the first table, followed by row i of the second table.

000000000000001212020111202111100122001122120010112201
000000000000010110221220022222101012012011121111201212
000000000000100112222121000101210001122002021102200221
000000000001000012201102201012000222222101020012220010
000000000010000221011021020201002222221120000122200100
000000000100000011100021101211122011212121220202110200
000000001000000111000211210111220112121122202021102000
000000010000000222012002222010101002210110202201211101
000000100000000111110100221122110111102120012022220210
000001000000000202111220220101001022020220200212120202
000010000000000122102011021211110012200121022101011220
000100000000000011120212212021121011222212202120221201
001000000000000220002122011221220212122111202200112122
010000000000000212001110200110001000221001101012011201
100000000000000222101211012221120221111102211101121020
000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000

111110112202212011120002201100202002210001101020020010

122221100120100112022022201210101002210201200122222100

010122102111120000112220102021201201200102111212122220

102222002010221122000210112102001100020122022102222211

221202100222012012020011120210001121100111210022200222

000210000200101012021010101121200110100112202011002012

020001020010012001200111010112200112000010010101202012

022210021102112222221002222211000010110011120022001211

101001122010101200022122111021200012020002220120101120

020010022011101012122210212002000001120002200000212011

111101212112021112010220021000200000200002010000001221

201210001122022110110111022120100002010000220000100120

212020012102101012102222222011200001220000000000110202

002100200120022100011121122221200000120000000000010011

011202120201110222222002100121200001110000000000020012

000000000000000000000000000000011111110000001112112222

000000000000000000000000000000000000001111111111111111

2.11 Ternary $[34, 20, 8]$

We start from the extended primitive BCH-code $[27, 14, 8]$ and obtain a code $[34, 20, 8]$.

2.11.1 Check matrix of ternary code $[34, 20, 8]$

10000000000000200222202110000120112
01000000000000221212122001102011 01
00100000000000101120001022102120 22
00010000000001011112210120101122 22
00001000000000211010212111101120 10
00000100000001211211112222100210 10
00000010000000202100010210110100 221
00000001000000000001111111100112 22
00000000100001100002102121000010 12
00000000010002020011202200200011 20
00000000001000001102110210200001 21
00000000000101122112100101100000 20
00000000000010122001011020100000 02
00000000000000000000000000001111111

2.12 Ternary code $[44, 29, 8]$

This was constructed in [2] as 3-point extension of the A-code $[41, 26, 8]$ corresponding to the BCH-code $[40, 26, 7]$.

2.12.1 Check matrix of ternary $[44, 29, 8]$

10000000000000012002102222221212022121002110
01000000000000012212111200002000202200122112
00100000000000012020100120002021110102211222
00010000000000011102222000002121120212202212
00001000000000020100121012001111021101022202
00000100000000010112201101202202222022001212
00000010000000021222011111010022120020101120
00000001000000020212001110121102102020001112
00000000100000021020012111102211122120111011
00000000010000021000112211111100022000112100
00000000001000010202200221110201122112210200
00000000000100022000100002212011221012220011
00000000000010001020220022111020112211221201
00000000000001002200010000221201122101222201
00000000000000111022220000021211202122021001

2.13 Ternary code $[83, 66, 8]$

This was constructed in [1].

2.13.1 Check matrix of ternary code $[83, 66, 8]$

```
1000000000000000012122111022222012220211201100101112121210120000212002011222022012
0100000000000000020012012201010201101210211111102221220002202102100001020200121 2
00100000000000000122221212101001210210002210122022210212101010202110010000100201
00010000000000000101222102010001002011011002211122011212102002002020222021211200
00001000000000000221201001100002220021102220102120021212112220001122012202021222
00000100000000000211210010001012121122012101210222212102111122100222010222221010
0000001000000000021121001000101212112201210121022221210211112210022201022222101
00000001000000000112010212210002122100020122002120201000020221121011210111201002
0000000010000000021121001000101212112201210121022221210211112210022201022222210
000000000100000002021020220210021022011012020020020110121112121102201002101110 2
00000000001000000202102022021002102201101202002002011012111212110220100210111022
00000000000100000220222021020212211010110202122110011022200002122221210201122002
0000000000001000022022202102021221101011020212211001102220000212222121020112200
00000000000001000122210201000100201101100221112201121210200200202022202121120010
00000000000000100101112021001021211222021112110211121220121202001112110201112022
00000000000000010220100221010201012101200022120011121022001220022011221001002112
00000000000000001121020022000011100122011102012100121212221110002211021101012111 2
```

112
201
211
111
112
022
020
212
120
200
002
011
201
100
112
112
220

2.14 Ternary code $[25, 10, 10]$

This was constructed in [2] as a 4-step extension of the A-code (parameters $[21, 6, 10]$) corresponding to the BCH-code $[20, 6, 9]$, case $n = 4, w = 20$.

2.14.1 Generator matrix of ternary code $[25, 10, 10]$

1121110001020221000000000
2102011011021110100000000
1200220110111020010000000
2120122022100020001000000
0122011220002220000100000
1112000110220210000010000
1021220212121010000001000
1011212111022210000000100
1201221002000010000000010
1101200201022120000000001

2.15 Ternary code $[30, 7, 16]$

This optimal code was constructed in [2] and at about the same time independently by Boukliev.

2.15.1 Weight distribution of ternary code $[30, 7, 16]$ and its dual

0	1	1
1	0	0
2	0	0
3	0	0
4	66	0
5	2440	0
6	18434	0
7	116352	0
8	682144	0
9	3358400	0
10	14072406	0
11	51148624	0
12	161964932	0
13	448578048	0
14	1089571012	0
15	2324118560	0
16	4357481454	180
17	7177654012	316
18	10367793752	212
19	13095479904	288
20	14405040154	252
21	13719820184	180
22	11225024922	306
23	7808183776	240
24	4555371476	174
25	2186381016	36
26	840883894	2
27	249185392	0
28	53394000	0
29	7362112	0
30	491360	0

2.16 Ternary code $[33, 8, 17]$

This code was constructed in [2]. It is not optimal, the upper bound on d being $d \leq 18$.

2.16.1 Weight distribution of ternary code $[33, 8, 17]$ and its dual

0	1	1
1	0	0
2	2	0
3	0	0
4	24	0
5	604	0
6	14342	0
7	78984	0
8	539478	0
9	3011244	0
10	14443956	0
11	60462354	0
12	221451804	0
13	715638312	0
14	2044413312	0
15	5180989800	0
16	11654002638	0
17	23309081442	444
18	41440940058	474
19	65430294300	522
20	91603419504	682
21	113413163652	528
22	123722559270	786
23	118347276144	896
24	98619270018	684
25	71004845520	1098
26	43698901104	390
27	22656469922	2
28	9710131644	0
29	3348600712	8
30	892919124	12
31	172758864	24
32	21624794	10
33	1306516	0

2.17 Ternary code $[85, 7, 54]$

This length-optimal code was constructed in [2].

2.17.1 Weight distribution of ternary code $[85, 7, 54]$ and its dual

0	1	1
1	0	0
2	0	0
3	8	0
4	15120	0
5	486000	0
6	12945960	0
7	287017992	0
8	5644202796	0
9	96329352996	0
10	1465539483240	0
11	19976918023656	0
12	246420354121920	0
13	2767333779980784	0
14	28464561855628488	0
15	269462705750809704	0
16	2357803898471008998	0
17	19139807384644147290	0
18	144611900934435501912	0
19	1019894435219250400104	0
20	6731303236774594616952	0
21	41669972723493521824824	0
22	242443476580605013697400	0
23	1328168613538451948850936	0
24	6862204498187071836266472	0
25	33487557957944444442752136	0
26	154557959803220647028032056	0
27	675475528008440127003748152	0
28	2798398616120869439290897944	0
29	11000601456262159783707099864	0
30	41068912103758534584125440056	0
31	145728397787045202491165961912	0
32	491833342531603324879566349428	0
33	1579828312374696333034198074108	0
34	4832416014320155547733318639720	0
35	14083040956024999846401924738888	0
36	39119558211168814057836300240888	0
37	103613964991761395006883960779160	0
38	261761595768642116731976590505832	0
39	630912564160327914000074426125224	0
40	1451098897568764218244717977788352	0

41	3185339043443577375464075645948304	0
42	6674043710072375947124538055070760	0
43	13348087420144541931801181925666856	0
44	25482712347548985381208116785515512	0
45	46435164722199955171702777384117656	0
46	80756808212522172531809219003252712	0
47	134021937033546852979663250846288040	0
48	212201400303116461340213343838390548	0
49	320467420865930399886813857796753756	0
50	461473086046940289465587037262285416	0
51	633394431829133287482087779310075528	0
52	828285026238097764713004365404153416	0
53	1031449277956876137590665750109804808	0
54	1222458403504446013104224584186378152	1536
55	1378044018495920859393636416683611432	0
56	1476475734102772376259036359429596056	0
57	1502378817157206952516613205370942008	0
58	1450572651048337807857987342100980456	0
59	1327642765366275214280409657333474792	0
60	1150623729984105217825699699752523848	0
61	943134204905004287781454727331163656	0
62	730168416700648454470137411354630120	0
63	533138843940156028937552533925831784	648
64	366532955208857271242118660088589838	0
65	236836678750338538819676636769664674	0
66	143537381060811236875625072487108048	0
67	81409260900161599376334534135892800	0
68	43099020476556139715976916073113032	0
69	21237198495694327598164397171758584	0
70	9708433598031696453007676863228752	0
71	4102155041421840305140157229436464	0
72	1595282516108495156517405583891860	0
73	568182813956449884419789526923916	0
74	184275507229118761435736610069072	0
75	54054148787208381734792799649488	0
76	14224775996633647144435046280936	0
77	3325272310901433281956080636312	0
78	682107140697709169433785425200	0
79	120879746452764093437932380624	0
80	18131961967913463418082992806	0
81	2238513823199379649225718762	2
82	218391592507233766051892832	0
83	15787344036669478021254960	0
84	751778287460342398417312	0
85	17688900881422692916128	0

3 The quaternary case

3.1 Quaternary code $[82, 73, 5]$

This was constructed in [2] as a 19-point extension of primitive BCH-code $[63, 54, 5]$ (case $n = 3, w = 63$).

3.1.1 Check matrix of quaternary $[82, 73, 5]$

10000000013323030033231033230220312210333111212322222302323022132203322320330323
01000000012011333030112130113202323031300200333110000132111320322321232302201220
00100000033013113321132231131230003233312313102100111212003123322212213132122133
00010000031113331310030201033033231213113102221001100320012303011132113303200331
0000100003111333131003020103303323121311310222100110032001230303000133223003122
00000100032103113331023320130220103222313202113001100202012132323332001220302112
00000010031002331311021310133132221212013013320111001221013202131300010031031320
00000001010223203102333102223133130331132210120333322220222302331200100113012310
00000000133230300332310332302203122103331112123222223023230221101311011231200321

13
32
20
10
30
12
11
00
22

3.2 Quaternary $[126, 116, 5]$

This was constructed in [1].

3.2.1 Check matrix of quaternary $[126, 116, 5]$

11111111111111111111111111111111111111111111111111111111111111111110110222320031
01231032230132101032012332102301230132100123103232102301103201231000100122311002
01231032230132102301321001231032321023011032012310320123321023010100120100230320
01232301321010321032321023010123230101231032321032101032012323010010010101120010
01321023320123101023013223103201320123100132102323103201102301320000000010001110
01322310102332011023320101322310320110232310013223100132320110230000000000000000
01321023320123103201231001321023231032011023013210230132231032010000000000000000
01112322223222233233100000010010332300011000010033320010010010000001110001112223
01111303133010330221321132221303012213303121322202123222103331120000001111111111
01111330103313031220333221201123102213123233201212022220112313323000000000000000

1213223023112300020210032003001203200001021030
3000223230013330133002101300033121101220001112
2110211022303003113013111301303202223222320210
0101202001001001323032112111002003301200101310
0000000111201322230130223110120133000020132320
0000000000000000011011111212230011331201313103
0000000000000000000000000000001100001111111311
3300011122233300011122233300011122233300011122
1100000000000011111111111122222222222333333333
0011111111111111111111111111111111111111111111

3.3 The quaternary code $[81, 70, 6]$

Construction: start from the primitive A-code $[64, 8, 33]$. By theory this possesses a 3-step extension $[67, 11, 33]$. Its dual has parameters $[67, 56, 6]$. We constructed a 14-step extension of this code by computer.

3.3.1 Check matrix of quaternary $[81, 70, 6]$

```
100000000002121033213211330111310330011303221102303302221210120223011120132120112
010000000001222302321320102301132112031132302033203021002223321110022010301201320
001000000001300121313212112130113023230221132231230323012001013032113311000312311
000100000000130011231321211211330012322302113222331032130200101322311011030000301
000010000003102211330322220320221200210101010301220110320132203211010322303011121
000001000002023311001102021130001203033313002023100023200322312112221200201302211
000000100001020011221230300211220112332203002232220023030211110333212212233330101
000000010003013211331333331221220210211001101200330011331133302330322230202323033
000000001003210113320323032321123210003231311101202012312001123113033021312201103
000000000101103333213312201130300033033101233121202220121023233202320203202023320
000000000012223230213201023011113220313123020303320210202233211132223112220111203
```

3.3.2 Weight distribution of quaternary $[81, 70, 6]$ and its dual

0	1	1
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	89670	0
7	1612902	0
8	50909001	0
9	1221360867	0
10	26466210360	0
11	511959694968	0
12	8961557660442	0
13	142687646373258	0
14	2079189717112074	0
15	27861059218823562	0
16	344780849117583456	0
17	3954838272328932339	0
18	42184944722500925568	0
19	419629180982931690624	0
20	3902551377164748971760	0
21	34007947782241901111280	0
22	278246845307003277650520	0
23	2141290940271262901320152	0
24	15524359316335983394155492	0
25	106186617726696080073681180	0
26	686128914529826237353327968	0
27	4193010033267296371017100896	0
28	24259558049580509338247847768	0
29	133009301030459987146649153304	0
30	691648365358425586413950793864	0
31	3413619351607870413972020892552	0
32	16001340710661196888174173899070	0
33	71278699529309844656551550189292	0
34	301886256830018966868875594797824	0
35	1216170348943786216574197406589696	0

36	4661986337617853663572048236371088	0
37	17009950150767842412339070188251280	0
38	59087195260561972884459723014431476	0
39	195442261246474221177845264334785460	0
40	615643122926393817649695785597500398	0
41	1846929368779181403853689895289062218	0
42	5276941053654804058306775183527687056	54
43	14358188448316559864540134588243113744	36
44	37200760979729268738476992212315790620	168
45	91761877083332196196390859957452877628	234
46	215440928804345156353600432268960815116	678
47	481304202648005136434896738627492710924	1698
48	1022771430627010915040955108598450799244	3729
49	2066415747593348583425269709804692520634	6636
50	3967518235379229279923420435449243456128	10098
51	7234886193926829863840509526997663583872	17808
52	12521918412565667071981431946280352035472	34779
53	20554847205532321418996594717077698491280	61128
54	31974206764161388876334955988955451049080	94950
55	47089286325401318159824065625933299731448	146124
56	65588648810380407441697730718269670587412	220437
57	86300853697868957153974221798565636420428	281280
58	107132094245630429577072933242826916404192	302994
59	125290076321161010855748294464398061301984	342381
60	137819083953277111945199087392726657008984	426891
61	142337742443548492661827646472548720947224	456645
62	137746202364724347739904183612598230953128	368604
63	124627516425226790810211591351125506276776	330702
64	105154466983785104746856761940360647921641	336570
65	82505812556508312955741472889776255450940	311904
66	60004227313824227603412245634261900596736	192618
67	40301346703314779733784479864808679738880	105966
68	24892008257929716894873244349370886008048	64293
69	14069395971873318244347213079388111740656	40422
70	7235689356963420811667337548610085852662	19554

71	33630668842224350251645409236999365202 14	8100
72	1401277868426014593662423085846708240081	2667
73	518280855445238274508743191852376178539	1164
74	168091088252509710572648280717295922424	1116
75	47065504710702718992325814369848973112	969
76	11147093220955907120014160175334531098	477
77	2171511666419981908731566464303692042	267
78	334078717910766447120104606802307002	54
79	38059600774644278830227842803643706	72
80	2854470058098320908996184149285356	36
81	105721113262900774521126582094635	0

3.4 The quaternary code $[93, 81, 6]$

This was constructed in [2] as a 30-step extension of the primitive BCH-code $[63, 51, 6]$ (case $n = 3, w = 63, t = 5$).

3.4.1 Check matrix of quaternary $[93, 81, 6]$

100000000000233131020021132320122203230321011032212112002203303320223321333310021
010000000000101030112011232022303322203221132122310120201322132211020012300101 21
001000000000301311021233330332003033130210131201112231023233312033013201331021 30
000100000000321323132111120103033002003113031133232000101022230321021303022312 03
000010000000321323132111120103033002003113031133232000101022231010110202332220 01
000001000000121130221332230011321232100222103330203011000112020203323312302002 21
000000100000012113022133223001132123210022210333020301100011202121333112300000 03
000000010000310003332221131230320113031130203020021213113300021210201330201112 01
000000001000202131313203021203110212133232031330210033313133301323133311101132 10
000000000100213322111301230200233222023002212101233111333110033021321010312131 11
000000000010103011201123202230332220322113212231012020132213201023022032120012 11
000000000001223230100133212103111021202130330211313310011022023313313012121332 02

3302301330030
1122231121331
3121313231212
0201230301231
1303313321300
1033003013332
3122320120200
1100300210213
2013112203023
2003030133013
2110330033330
0321312322100

3.5 Quaternary $[70, 57, 7]$

This was constructed in [1].

3.5.1 Check matrix of quaternary code $[70, 57, 7]$

```
1000000000000323102211120230122030332100120020012130203330332332302020
0100000000000233131020021132320122203230321011032212112002203303200231
0010000000000101030112011232022303322203221132122310120201322132112021
0001000000000301311021233330332003033130210131201112231023233312332120
0000100000000321323132111120103033002003113031133232000101022230111212
0000010000000321323132111120103033002003113031133232000101022223211133
0000001000000121130221332230011321232100222103330203011000112020233102
0000000100000121130221332230011321232100222103330203011000112020222031
0000000010000310003332221131230320113031130203020021213113300021011302
0000000001000202131313203021203110212133232031330210033313133301121312
0000000000100213322111301230200233222023002212101233111333110033202130
0000000000010103011201123202230332220322113212231012020132213201032201
0000000000001223230100133212103111021202130330211313310011022023133223
```

3.6 Quaternary code $[53, 6, 36]$

3.6.1 Generator matrix of quaternary $[53, 6, 36]$

10000032303321033010333122002333103213001320322001132
01000021113031032331333011232132022212231101203210 10
00100010220203122232331032132223102332211303302221302
00010001032303321030313033122002333123003103123120232
00001010323033210331030331220023331002331012330022133
00000103230332103303103312200233310330210133202112031

This was constructed in [2] as an extension of a BCH-code in case $q = 4, n = 4, w = 51$. It is d-optimal. After concatenation a binary code $[159, 12, 72]$ is obtained. These parameters are new. All weights of this binary code are multiples of 8.

3.6.2 Weight distribution of quaternary $[53, 6, 36]$ and its dual

0	1	1
1	0	0
2	0	0
3	0	0
4	5763	0
5	171360	0
6	4115904	0
7	82129584	0
8	1420067358	0
9	21294291120	0
10	281112522240	0
11	3296571329520	0
12	34614111980754	0
13	327502864383024	0
14	2807169396132672	0
15	21895911891500496	0
16	156008364900327291	0
17	1018642992438132432	0
18	6111857589419421696	0
19	33776055281439212304	0
20	172257882888364272153	0
21	812072875019876035632	0
22	3543590724235719143808	0
23	14328432072118807713504	0
24	53731620254526233503956	0
25	186986038480262858403552	0
26	604108739742387798435840	0
27	1812326219201604112465248	0
28	5048623039151989423248684	0

29	13056783722063619010237152	0
30	31336280932889487973392000	0
31	69748496269914036319329696	0
32	143856273556722097160296551	0
33	274634704063184917394469792	0
34	484649477757693057304734720	0
35	789286292349366583009873440	0
36	1183929438523061390785849605	1284
37	1631902739586656650584570816	0
38	2061350828950505460065148096	0
39	2378481725713885200385293168	0
40	2497405811997095964811470894	1836
41	2375581138243871689352037744	0
42	2036212404206653445321353728	0
43	1562674635788175552515707824	0
44	1065459978945519863958633186	918
45	639275987367768008082108912	0
46	333535297756918907179088448	0
47	149026409636126500146601872	0
48	55884903613533029177197821	51
49	17107623555166072325898384	0
50	4105829653239414197265408	0
51	724558174101121474908624	0
52	83602866242433728014719	6
53	4732237711835983559088	0

3.7 Quaternary code $[70, 7, 48]$

This is a length-optimal code, which was constructed in [2] by applying construction X to a pair $[64, 7, 44] \supset [64, 4, 48]$ of extended primitive BCH-codes.

3.7.1 Weight distribution of quaternary $[70, 7, 48]$ and its dual

0	1	1
1	0	0
2	0	0
3	0	0
4	2205	0
5	190080	0
6	5919858	0
7	159570432	0
8	3779168616	0
9	78137525760	0
10	1429754157408	0
11	23395794485760	0
12	345089397097512	0
13	4618889159606784	0
14	56416427530052976	0
15	631863972426742272	0
16	6516097202314642284	0
17	62094573742572320256	0
18	548502067806462040608	0
19	4503490657594281239040	0
20	34451703539286920071452	0
21	246083596733835804847104	0
22	1644285850804167365098872	0
23	10294659239787856895869440	0
24	60481123034407990829412408	0
25	333855799149207474570640896	0
26	1733482034041452721941612384	0
27	8474801055320913328882371072	0
28	39044619147730203181046067000	0

29	169642138365965757379196625408	0
30	695532767300504730424388946960	0
31	2692384905679436952649778692608	0
32	9844032311390176339699330808598	0
33	34006657075711741148117578403328	0
34	111021733394235806149600490742048	0
35	342581348759354951562807526476288	0
36	999195600548119644346409696846814	0
37	2754539223132654881484028392338688	0
38	7176299555003493344017258847070252	0
39	17664737366162446720080312140729856	0
40	41070514376327687622619303313656728	0
41	90154787655353467573713376828740096	0
42	186749203000375020608243892189937440	0
43	364812396558872155159429329041032704	0
44	671586457301560113602713625745344088	0
45	1164083192656037449830309757678401024	0
46	1897961727156582956127990356052001296	0
47	2907515837346254550463019715585289728	0
48	4179554016185241074840527180639563660	4572
49	5629603368739304211602199635632660992	0
50	7093300244611523364774459971621320032	1728
51	8345059111307674453608036455497606656	0
52	9147468641241104872656357982143724668	2880
53	9320062389189049974203335183572817920	0
54	8802281145345214178426616811212924792	1152
55	7681990817755822992739068796316947968	0
56	6173028335696643699222363342555843144	4320
57	4548547194723842580895528496265914880	0
58	3058505872314308022806085999017058336	1728
59	1866206972937543839883925010354253312	0
60	1026413835115649127739609832841322440	0
61	504793689401138909699128341689427456	0
62	219829509900495978490565547566183280	0
63	83744575200188943755070839590639104	0
64	27478688737561997262256187484029553	3
65	7609483035017168456276518565140992	0
66	1729427962503901924195154975887200	0
67	309748291791743627961318838070784	0
68	40996097443024891956060724819053	0
69	3564878038523903647331509756032	0
70	152780487365310156339493304178	0

References

[1] J.Bierbrauer, Y.Edel: *New code parameters from Reed-Solomon subfield codes,* manuscript.

[2] J.Bierbrauer, Y.Edel: *Extending and lengthening BCH-codes,* manuscript.

[3] Y.Edel, J.Bierbrauer: *Twisted BCH-codes,* manuscript.

[4] Y.Edel, J.Bierbrauer: *Inverting construction* Y_1, manuscript.