Permutation decoding: an update

J. D. Key
Department of Mathematical Sciences
Clemson University
Clemson SC 29634
U.S.A.

March 29, 2003

Abstract

We give a brief survey of permutation decoding and some recent results in the search for PD-sets.

1 Introduction

Permutation decoding was first developed by MacWilliams [12]. It involves finding a set of automorphisms of the code, called a PD-set, that acts in a certain way such that the set can be used for decoding and correcting the maximum number of errors of which the code is capable. The method is described fully in MacWilliams and Sloane [13, Chapter 15] and, more recently, in Huffman [9, Section 8], where a survey of results up to the time of writing that chapter is given.

We will give here a brief, but complete, description of permutation decoding, and discuss some recent results. In particular we will look at codes defined by designs or graphs, where the automorphism group is known and large.

2 Codes from designs

Terminology for codes and designs will be as in Assmus and Key [1]. An incidence structure $\mathcal{D} = (\mathcal{P}, \mathcal{B}, \mathcal{I})$, with point set $\mathcal{P}$, block set $\mathcal{B}$ and incidence $\mathcal{I}$ is a t-(v, k, λ) design, if $|\mathcal{P}| = v$, every block $B \in \mathcal{B}$ is incident with precisely k points, and every t distinct points are together incident with precisely λ blocks.

The code of the design $\mathcal{D}$ over the finite field F is the space spanned by the incidence vectors of the blocks over F. If the point set of $\mathcal{D}$ is $\mathcal{P}$ and the block set is $\mathcal{B}$, and if $\mathcal{Q}$ is any subset of $\mathcal{P}$, then we will denote the incidence vector of $\mathcal{Q}$ by $v^{\mathcal{Q}}$. Thus the code of the design is $C = \langle v^B \mid B \in \mathcal{B} \rangle$, and is a subspace of $F^{\mathcal{P}}$, the full vector space of functions from $\mathcal{P}$ to F.

All the codes here will be **linear codes**, i.e. subspaces of the ambient vector space. If a code C over a field of order q is of length n, dimension k, and minimum weight d, then we write $[n, k, d]_q$ to show this information. A **generator matrix** matrix for the code is a $k \times n$ matrix made up of a basis for C. The **dual** or **orthogonal** code $C^\perp$ is the orthogonal under the standard inner product, i.e. $C^\perp = \{v \in F^n | (v, c) = 0 \text{ for all } c \in C\}$. A **check matrix** or **parity-check matrix** for C is a generator matrix for $C^\perp$; the **syndrome** of a vector $y \in F^n$ is Hy^T. A code C is **self-orthogonal** if $C \subseteq C^\perp$ and is **self-dual** if $C = C^\perp$. If c is a codeword then the **support** of c is the set of non-zero coordinate positions of c. A **constant word** in the code is a codeword, all of whose coordinate entries are either 0 or 1. The all-one vector will be denoted by $\boldsymbol{\jmath}$, and is the constant vector of weight the length of the code. Two linear codes of the same length and over the same field are **equivalent** if each can be obtained from the other by permuting the coordinate positions and multiplying each coordinate position by a non-zero field element. They are **isomorphic** if they can be obtained from one another by permuting the coordinate positions. Any code is isomorphic to a code with generator matrix in so-called **standard form**, i.e. the form $[I_k \mid A]$; a check matrix then is given by $[-A^T \mid I_{n-k}]$. The first k coordinates are the **information symbols** and the last $n - k$ coordinates are the **check symbols**. An **automorphism** of a code C is any permutation of the coordinate positions that maps codewords to codewords.

3 Permutation decoding

A **PD-set** for a t-error-correcting code C is a set $\mathcal{S}$ of automorphisms of C which is such that every possible error vector of weight $s \leq t$ can be moved by some member of $\mathcal{S}$ to another vector where the s non-zero entries have been moved out of the information positions. In other words, every t-set of coordinate positions is moved by at least one member of $\mathcal{S}$ to a t-set consisting only of check-position coordinates. That such a set will fully use the error-correction potential of the code follows from a result quoted below. That such a set exists at all is clearly not always true. There is a bound on the minimum size that the set $\mathcal{S}$ may have, and we will quote the relevant result below. Both the two results can be found with proofs in [9, Theorem 8.1].

Result 1 *Let C be an $[n, k, d]_q$ t-error-correcting code. Suppose H is a check matrix for C in standard form, i.e. such that I_{n-k} is in the redundancy positions. Let $y = c + e$ be a vector, where $c \in C$ and e has weight $\leq t$. Then the information symbols in y are correct if and only if the weight of the syndrome of y is $\leq t$.*

Proof: Suppose C has generator matrix G in standard form, i.e.

$$G = [I_k | A]$$

and that the encoding is done using G, i.e. the data set $x = (x_1, \ldots, x_k)$ is encoded as xG. The information symbols are then the first k symbols, and the check matrix H is

$$H = [-A^T | I_{n-k}].$$

Suppose the information symbols of y are correct. Then

$$Hy^T = He^T = e^T,$$

and thus $\mathrm{wt}(Hy^T) \leq t$.

Conversely, suppose that not all the information symbols are correct. Then if $e = e_1 \dots e_n$, and $e' = e_1 \dots e_k$, $e'' = e_{k+1} \dots e_n$, we assume that e' is not the zero vector. Now use the fact that for any vectors

$$\mathrm{wt}(x + y) \geq \mathrm{wt}(x) - \mathrm{wt}(y).$$

Then

$$\begin{aligned}
\mathrm{wt}(Hy^T) &= \mathrm{wt}(He^T) = \mathrm{wt}(-A^T e'^T + e''^T) \\
&\geq \mathrm{wt}(-A^T e'^T) - \mathrm{wt}(e''^T) \\
&= \mathrm{wt}(e'A) - \mathrm{wt}(e'') \\
&= \mathrm{wt}(e'A) + \mathrm{wt}(e') - \mathrm{wt}(e') - \mathrm{wt}(e'') \\
&= \mathrm{wt}(e'G) - \mathrm{wt}(e) \\
&\geq d - t \geq t + 1
\end{aligned}$$

which proves the result. □

The algorithm for permutation decoding then is as follows: we have a t-error-correcting $[n, k, d]_q$ code C with check matrix H in standard form. Thus the generator matrix G for C that is used for encoding has I_k as the first k columns, and hence as the information symbols. Any k-tuple v is encoded as vG. Suppose x is sent and y is received and at most t errors occur. Let $\mathcal{S} = \{g_1, \dots, g_s\}$ be the PD-set. Compute the syndromes $H(yg_i)^T$ for $i = 1, \dots, s$ until an i is found such that the weight of this vector is t or less. Now look at the information symbols in yg_i, and obtain the codeword c that has these information symbols. Now decode y as cg_i^{-1}. Note that this is valid since permutations of the coordinate positions correspond to linear transformations of F^n, so that if $y = x + e$, where $x \in C$, then $yg = xg + eg$ for any $g \in S_n$, and if $g \in Aut(C)$, then $xg \in C$.

The next result is also in [9] and due to Gordon [7] using a formula of Schönheim [14]:

Result 2 *If $\mathcal{S}$ is a PD-set for a t-error-correcting $[n, k, d]_q$ code C, and $r = n - k$, then*

$$|\mathcal{S}| \geq \left\lceil \frac{n}{r} \left\lceil \frac{n-1}{r-1} \left\lceil \dots \left\lceil \frac{n-t+1}{r-t+1} \right\rceil \dots \right\rceil \right\rceil \right\rceil.$$

In Gordon [7] and Wolfman [16] small PD-sets for the binary Golay codes were found. In Chabanne [6] abelian codes, i.e. ideals in the group algebra of an abelian group, are looked at using Groebner bases, and the ideas of permutation decoding are generalized. In general it is rather hard to find these PD-sets, and obviously they need not even exist. Note

that PD-seets need not be sought, in general, for codes with minimum weight 3 or 4, since correcting a single error is, in fact, simply done by using syndrome decoding, because in that case multiples of the columns of the check matrix will give the possible syndromes. Thus the syndrome of the received vector need only be compared with the columns of the check matrix, by looking for a multiple.

We use the following observation which we state as a more general lemma:

Lemma 1 *Suppose C is a $[n,k,d]_q$ t-error-correcting code, and let $r = n-k$. Let $\mathcal{T}$ denote the set of t-tuples of the elements of $\{1,\ldots,n\}$ and $\mathcal{E}$ the set of t-tuples of the elements of the check positions $\{k+1,\ldots,n\}$. Then a set $\mathcal{S} = \{g_1,\ldots,g_s\}$ of automorphisms will be a PD-set for C if*

$$\bigcup_{g\in\mathcal{S}} \mathcal{E}^{g^{-1}} = \mathcal{T}.$$

Furthermore, for any $g \in Aut(C)$, the set $g\mathcal{S} = \{gg_1,\ldots,gg_s\}$ will also be a PD-set.

Proof: The first part is clear. The second statement can be proved as follows: we need to show that any t-tuple $\beta \in \mathcal{T}$ satisfies $\beta = \alpha^{e^{-1}}$ for some $\alpha \in \mathcal{E}$ and $e \in g\mathcal{S}$. If $\beta^g = \gamma = \alpha^{h^{-1}}$ for some $\alpha \in \mathcal{E}$ and $h \in \mathcal{S}$, then $\beta = \alpha^{h^{-1}g^{-1}} = \alpha^{(gh)^{-1}}$, as required. □

4 Cyclic codes

MacWilliams [12] developed a theory for finding PD-sets for cyclic codes.

An $[n,k,d]_q$ C is cyclic if whenever $c = c_1c_2\ldots c_n \in C$ then every cyclic shift of c is in C. Thus the mapping $\tau \in S_n$ defined by

$$\tau : i \mapsto i+1$$

for $i \in \{1,2,\ldots n\}$, is in the automorphism group of C, and $\tau^n = 1$. If a message c is sent and t errors occur, then if e is the error vector and if there is a sequence of k zeros between two of the error positions, then τ^j for some j will move the sequence of zeros into the information positions, and thus all the errors will occur in the check positions. Thus $<\tau>$ will be a PD-set for C if $k < \frac{n}{t}$.

As shown in [12], if q is a number prime to the length n, then the map

$$\rho : i \mapsto qi$$

is also an automorphism of the cyclic code and in the normalizer N of $<\tau>$. MacWilliams examines cases where N contains a PD-set.

5 PD-sets through computation

For small t PD-sets can be found computationally. Using Magma such sets have been found, and a list of codes and corresponding PD-sets is given at the website:

`http://www.ces.clemson.edu/~keyj`

under the list of Magma [3] results on PD-sets.

We examined the hermitian and Ree unitals on 28 points, both of whose codes are only single-error correcting, but nevertheless we did go ahead and find PD-sets. We then also looked at some codes from desarguesian projective planes; these codes are cyclic and we were able to find PD-sets in the normalizer of a regular cyclic subgroup of the automorphism group. These were not necessarily the smallest PD-sets.

In addition we looked at codes arising from some primitive actions of finite simple groups and found PD-sets where possible. We list the action of the alternating group A_9 below in item 7 and item 8 on two different designs on 126 points. See [11] for a fuller description of this type of action.

Some examples:

1. C the $[28, 21, 4]_2$ code of the hermitian unital on 28 points, a 2-(28,4,1) design has bound 4; $\mathrm{Aut}(C)$ is $\mathrm{Sp}_6(2)$ and a PD-set of four elements was found.

2. $C^\perp$, for C as above, is a $[28, 7, 12]_2$; here the bound is 10; a PD-set of 30 elements was found.

3. C the $[28, 19, 4]_2$ code of the Ree unital on 28 points, a 2-(28,4,1) design, has bound 4; $\mathrm{Aut}(C)$ is $P\Gamma L_2(8)$ and a PD-set of four elements was found.

4. C the $[31, 16, 6]_5$ code of the projective plane of order 5, $PG_2(5)$, a 2-(31,6,1) design; C has $PGL_3(5)$ acting and is a cyclic code, the bound is 7, and a PD-set of 14 elements, inside a cyclic group of order 31, was found.

5. The dual of the above is a $[31, 15, 10]_5$ self-orthogonal code, with bound 28, and the normalizer of a Sylow 31-subgroup has order 93. One such group was found to be a PD-set.

6. C the $[57, 29, 8]_7$ code of the plane $PG_2(7)$, with $PGL_3(7)$ acting and is a cyclic code; the bound is 15, and a PD-set of 43 elements was found inside the normalizer (of order 171) of a regular cyclic group of order 57; also one of size 54 was found inside a regular cyclic group of order 57. (Computations for this code are included in the appendix, Section 8, and a practice decoding run is shown.)

7. The binary code C of a 1-$(126, 20, 20)$ design acted on by A_9 is a $[126, 56, 6]_2$ code with dual a $[126, 70, 5]_2$ code; the bound is 4, and we found a PD-set of size 17; for $C^\perp$ the bound is 7 and we found a PD-set of size 32.

8. The binary code C of a 1-$(126, 40, 40)$ design acted on by A_9 is a $[126, 48, 16]_2$ code and its dual is a $[126, 78, 5]_2$ code; the bound for $C^\perp$ is 8, and we found a PD-set of size 43 for $C^\perp$.

(In each case the bound referred to is the lower bound of Result 2.)

6 Codes from triangular graphs

Recently Key, Moori and Rodrigues [10] obtained explicit PD-sets for the binary codes defined by the adjacency matrix of a triangular graph. For any n, the triangular graph $T(n)$ is defined to be the line graph of the complete graph K_n. It is a strongly regular graph on $v = \frac{n(n-1)}{2}$ vertices, i.e. on the pairs of letters $\{i, j\}$ where $i, j \in \{1, \ldots, n\}$. The binary codes formed from the span of the adjacency matrix of triangular graphs have been examined by Tonchev [15, p. 171] and Haemers, Peeters and van Rijckevorsel [8, Theorem 4.1]. See also [4, 5, 1, 2]. In particular the dimension and weight enumerator of these codes are easily determined. The code formed by the span of the adjacency matrix is also the code of the 1-$(\frac{n(n-1)}{2}, 2(n-2), 2(n-2))$ design $\mathcal{D}$ obtained by taking the rows of the adjacency matrix as the incidence vectors of the blocks; the automorphism group of this design will contain the automorphism group of the graph, the latter of which is easily seen to be S_n. Similarly, the automorphism group of the code will contain S_n. The facts about the binary codes are summarized in the following, the proof of which can be found in [10].

Result 3 *Let C be the binary code obtained by the row span of an adjacency matrix for the triangular graph $T(n)$, where $n \geq 5$.*

If $n = 2m$ then C is a self-orthogonal $[\binom{2m}{2}, 2m-2, 4(m-1)]_2$ code with weight distribution the zero vector and

$$< 4(m-1), \binom{2m}{2} >, < 8(m-2), \binom{2m}{4} >, \ldots, < m^2, \frac{1}{2}\binom{2m}{m} >$$

if m is even, and

$$< 4(m-1), \binom{2m}{2} >, < 8(m-2), \binom{2m}{4} >, \ldots, < m^2 - 1, \binom{2m}{m-1} >$$

if m is odd.

If n is odd, then C is a $[\binom{n}{2}, n-1, n-1]_2$ code with weight distribution the zero vector and

$$< n-1, n >, \ldots, < 2i(n-2i), \binom{n}{2i} >, \ldots,$$

where $1 \leq i \leq (n-1)/2$.

The minimum weight of $C^\perp$ is 3. The automorphism group of C is S_n unless $n = 6$, in which case it is $PGL_4(2) \cong A_8$.

(Here $< i, j >$ denotes j vectors of weight i.)

In [10], explicit PD-sets for these codes are obtained. The following ordering of the points is used:

$$P_1 = \{1, n\}, P_2 = \{2, n\}, \ldots, P_{n-1} = \{n-1, n\}, \tag{1}$$

first, followed by the set

$$P_n = \{1,2\}, P_{n+1} = \{1,3\}, \ldots, P_{2n-2} = \{2,3\}, \ldots, P_{\binom{n}{2}} = \{n-2, n-1\}. \qquad (2)$$

The generator matrix for C is shown in [10] to be in standard form, with the first $n-1$ coordinates the information symbols for n odd, and the first $n-2$ for n even. For $n \geq 5$ PD-sets can be found for the code C.

Result 4 *Using the ordering of the point set $\mathcal{P}$ given in Equations* (1) *and* (2)*, the following sets of permutations in S_n in the natural action on the points $\mathcal{P}$, are PD-sets for the binary code C of the triangular graph $T(n)$.*

1. *For $n \geq 5$ odd,*
$$\mathcal{S} = \{1_G\} \cup \{(i,n) \mid 1 \leq i \leq n-1\}$$
is a PD-set of n elements.

2. *For $n \geq 6$ and even,*
$$\mathcal{S} = \{1_G\} \cup \{(i,n) \mid 1 \leq i \leq n-1\} \cup \{[(i,n-1)(j,n)]^{\pm 1} \mid 1 \leq i,j \leq n-2\}$$
is a PD-set of $n^2 - 2n + 2$ elements.

This result is proved by examining the action of the group S_n on the points in the positions given.

7 Conclusions

The success of finding PD-sets for the triangular graphs came about by ordering the points in such a way that the nature of the information symbols was known, and the action of the automorphism group apparent. A similar type of approach using the codes of finite geometries, and the projective linear groups, seems to be a possible route for finding PD-sets for those codes. In addition, the PD-sets could be looked for in the normalizer of a Singer group.

Acknowledgement

The author thanks MURST and the research unit "Gruppi, Grafi e Geometrie" in the Dipartimento di Matematica at the Universitá di Roma 'La Sapienza' for their hospitality and financial support in June 2001.

8 Appendix: sample computation

```
The following is the log of a Magma run using the 7-ary code of the
plane PG_2(7) of order 7, i.e. a [57,29,8]_7 code with PGL_3(7)
acting.  A PD-set of 43 elements was found inside the normalizer of a
Singer cycle acting on the plane, the set having 43 elements. The
normalizer has order 171, i.e. 57x3.  A run showing the decoding
algorithm working is then shown, along with the Magma routine that is
used; the latter is quoted at the end as the file "qudecodeR.m". The
code C is obtained by taking the span over F_7 of the incidence vectors
of the 57 lines given below:

 { 15, 25, 27, 28, 33, 42, 49, 53 },{ 9, 11, 12, 17, 26, 33, 37, 56 },
 { 1, 6, 15, 22, 26, 45, 55, 57 },{ 3, 12, 19, 23, 42, 52, 54, 55 },
 { 2, 3, 8, 17, 24, 28, 47, 57 },{ 6, 13, 17, 36, 46, 48, 49, 54 },
 { 1, 10, 17, 21, 40, 50, 52, 53 },{ 4, 6, 7, 12, 21, 28, 32, 51 },
 { 8, 18, 20, 21, 26, 35, 42, 46 },{ 18, 28, 30, 31, 36, 45, 52, 56 },
 { 7, 14, 18, 37, 47, 49, 50, 55 },{ 4, 14, 16, 17, 22, 31, 38, 42 },
 { 2, 11, 18, 22, 41, 51, 53, 54 },{ 9, 16, 20, 39, 49, 51, 52, 57 },
 { 4, 8, 27, 37, 39, 40, 45, 54 },{ 1, 11, 13, 14, 19, 28, 35, 39 },
 { 1, 8, 12, 31, 41, 43, 44, 49 },{ 3, 7, 26, 36, 38, 39, 44, 53 },
 { 9, 19, 21, 22, 27, 36, 43, 47 },{ 12, 22, 24, 25, 30, 39, 46, 50 },
 { 13, 23, 25, 26, 31, 40, 47, 51 },{ 8, 15, 19, 38, 48, 50, 51, 56 },
 { 2, 21, 31, 33, 34, 39, 48, 55 },{ 5, 7, 8, 13, 22, 29, 33, 52 },
 { 3, 13, 15, 16, 21, 30, 37, 41 },{ 5, 12, 16, 35, 45, 47, 48, 53 },
 { 10, 12, 13, 18, 27, 34, 38, 57 },{ 2, 4, 5, 10, 19, 26, 30, 49 },
 { 4, 23, 33, 35, 36, 41, 50, 57 },{ 1, 3, 4, 9, 18, 25, 29, 48 },
 { 11, 21, 23, 24, 29, 38, 45, 49 },{ 4, 11, 15, 34, 44, 46, 47, 52 },
 { 6, 16, 18, 19, 24, 33, 40, 44 },{ 14, 24, 26, 27, 32, 41, 48, 52 },
 { 5, 15, 17, 18, 23, 32, 39, 43 },{ 7, 17, 19, 20, 25, 34, 41, 45 },
 { 3, 10, 14, 33, 43, 45, 46, 51 },{ 6, 8, 9, 14, 23, 30, 34, 53 },
 { 2, 6, 25, 35, 37, 38, 43, 52 },{ 16, 26, 28, 29, 34, 43, 50, 54 },
 { 7, 11, 30, 40, 42, 43, 48, 57 },{ 7, 9, 10, 15, 24, 31, 35, 54 },
 { 8, 10, 11, 16, 25, 32, 36, 55 },{ 5, 14, 21, 25, 44, 54, 56, 57 },
 { 1, 2, 7, 16, 23, 27, 46, 56 },{ 1, 5, 24, 34, 36, 37, 42, 51 },
 { 3, 22, 32, 34, 35, 40, 49, 56 },{ 17, 27, 29, 30, 35, 44, 51, 55 },
 { 4, 13, 20, 24, 43, 53, 55, 56 },{ 19, 29, 31, 32, 37, 46, 53, 57 },
 { 1, 20, 30, 32, 33, 38, 47, 54 },{ 2, 12, 14, 15, 20, 29, 36, 40 },
 { 5, 9, 28, 38, 40, 41, 46, 55 },{ 3, 5, 6, 11, 20, 27, 31, 50 },
 { 2, 9, 13, 32, 42, 44, 45, 50 },{ 10, 20, 22, 23, 28, 37, 44, 48 },
 { 6, 10, 29, 39, 41, 42, 47, 56 }

2-(57, 8, 1) Design with 57 blocks
> Dim(C);
29

> au;
```

```
Permutation group au acting on a set of cardinality 57
Order = 5630688 = 2^5 * 3^3 * 7^3 * 19
\\the PD-set of 43 elements
> PDset43;
[
 Id(nhx),
 (1, 44, 55, 19, 7, 15, 16, 47, 8, 27, 52, 25, 9, 13, 53, 57, 40, 14, 38)(2, 4,
  32, 43, 29, 42, 20, 31, 30, 12, 17, 54, 41, 35, 3, 45, 24, 37, 48)(5, 28,
  39, 26, 6, 18, 50, 56, 46, 11, 36, 22, 33, 49, 51, 23, 34, 10, 21),
 (1, 38, 14, 40, 57, 53, 13, 9, 25, 52, 27, 8, 47, 16, 15, 7, 19, 55, 44)(2,
  48, 37, 24, 45, 3, 35, 41, 54, 17, 12, 30, 31, 20, 42, 29, 43, 32, 4)(5,
  21, 10, 34, 23, 51, 49, 33, 22, 36, 11, 46, 56, 50, 18, 6, 26, 39, 28),
 (1, 55, 7, 16, 8, 52, 9, 53, 40, 38, 44, 19, 15, 47, 27, 25, 13, 57, 14)(2,
 32, 29, 20, 30, 17, 41, 3, 24, 48, 4, 43, 42, 31, 12, 54, 35, 45, 37)(5,
 39, 6, 50, 46, 36, 33, 51, 34, 21, 28, 26, 18, 56, 11, 22, 49, 23, 10),
 (1, 14, 57, 13, 25, 27, 47, 15, 19, 44, 38, 40, 53, 9, 52, 8, 16, 7, 55)(2,
 37, 45, 35, 54, 12, 31, 42, 43, 4, 48, 24, 3, 41, 17, 30, 20, 29, 32)(5,
 10, 23, 49, 22, 11, 56, 18, 26, 28, 21, 34, 51, 33, 36, 46, 50, 6, 39),
 (1, 19, 16, 27, 9, 57, 38, 55, 15, 8, 25, 53, 14, 44, 7, 47, 52, 13, 40)(2,
 43, 20, 12, 41, 45, 48, 32, 42, 30, 54, 3, 37, 4, 29, 31, 17, 35, 24)(5,
 26, 50, 11, 33, 23, 21, 39, 18, 46, 22, 51, 10, 28, 6, 56, 36, 49, 34),
 (1, 40, 13, 52, 47, 7, 44, 14, 53, 25, 8, 15, 55, 38, 57, 9, 27, 16, 19)(2,
 24, 35, 17, 31, 29, 4, 37, 3, 54, 30, 42, 32, 48, 45, 41, 12, 20, 43)(5,
 34, 49, 36, 56, 6, 28, 10, 51, 22, 46, 18, 39, 21, 23, 33, 11, 50, 26),
 (1, 7, 8, 9, 40, 44, 15, 27, 13, 14, 55, 16, 52, 53, 38, 19, 47, 25, 57)(2,
 29, 30, 41, 24, 4, 42, 12, 35, 37, 32, 20, 17, 3, 48, 43, 31, 54, 45)(5,
  6, 46, 33, 34, 28, 18, 11, 49, 10, 39, 50, 36, 51, 21, 26, 56, 22, 23),
 (1, 57, 25, 47, 19, 38, 53, 52, 16, 55, 14, 13, 27, 15, 44, 40, 9, 8, 7)(2,
 45, 54, 31, 43, 48, 3, 17, 20, 32, 37, 35, 12, 42, 4, 24, 41, 30, 29)(5,
 23, 22, 56, 26, 21, 51, 36, 50, 39, 10, 49, 11, 18, 28, 34, 33, 46, 6),
 (1, 15, 52, 57, 44, 16, 25, 40, 55, 47, 9, 14, 19, 8, 13, 38, 7, 27, 53)(2,
 42, 17, 45, 4, 20, 54, 24, 32, 31, 41, 37, 43, 30, 35, 48, 29, 12, 3)(5,
 18, 36, 23, 28, 50, 22, 34, 39, 56, 33, 10, 26, 46, 49, 21, 6, 11, 51),
 (1, 53, 27, 7, 38, 13, 8, 19, 14, 9, 47, 55, 40, 25, 16, 44, 57, 52, 15)(2, 3,
 12, 29, 48, 35, 30, 43, 37, 41, 31, 32, 24, 54, 20, 4, 45, 17, 42)(5, 51,
 11, 6, 21, 49, 46, 26, 10, 33, 56, 39, 34, 22, 50, 28, 23, 36, 18),
 (1, 16, 9, 38, 15, 25, 14, 7, 52, 40, 19, 27, 57, 55, 8, 53, 44, 47, 13)(2,
 20, 41, 48, 42, 54, 37, 29, 17, 24, 43, 12, 45, 32, 30, 3, 4, 31, 35)(5,
 50, 33, 21, 18, 22, 10, 6, 36, 34, 26, 11, 23, 39, 46, 51, 28, 56, 49),
 (1, 13, 47, 44, 53, 8, 55, 57, 27, 19, 40, 52, 7, 14, 25, 15, 38, 9, 16)(2,
 35, 31, 4, 3, 30, 32, 45, 12, 43, 24, 17, 29, 37, 54, 42, 48, 41, 20)(5,
 49, 56, 28, 51, 46, 39, 23, 11, 26, 34, 36, 6, 10, 22, 18, 21, 33, 50),
 (1, 47, 53, 55, 27, 40, 7, 25, 38, 16, 13, 44, 8, 57, 19, 52, 14, 15, 9)(2,
 31, 3, 32, 12, 24, 29, 54, 48, 20, 35, 4, 30, 45, 43, 17, 37, 42, 41)(5,
 56, 51, 39, 11, 34, 6, 22, 21, 50, 49, 28, 46, 23, 26, 36, 10, 18, 33),
 (1, 9, 15, 14, 52, 19, 57, 8, 44, 13, 16, 38, 25, 7, 40, 27, 55, 53, 47)(2,
 41, 42, 37, 17, 43, 45, 30, 4, 35, 20, 48, 54, 29, 24, 12, 32, 3, 31)(5,
```

```
33, 18, 10, 36, 26, 23, 46, 28, 49, 50, 21, 22, 6, 34, 11, 39, 51, 56),
(1, 8, 40, 15, 13, 55, 52, 38, 47, 57, 7, 9, 44, 27, 14, 16, 53, 19, 25)(2,
30, 24, 42, 35, 32, 17, 48, 31, 45, 29, 41, 4, 12, 37, 20, 3, 43, 54)(5,
46, 34, 18, 49, 39, 36, 21, 56, 23, 6, 33, 28, 11, 10, 50, 51, 26, 22),
(1, 25, 19, 53, 16, 14, 27, 44, 9, 7, 57, 47, 38, 52, 55, 13, 15, 40, 8)(2,
54, 43, 3, 20, 37, 12, 4, 41, 29, 45, 31, 48, 17, 32, 35, 42, 24, 30)(5,
22, 26, 51, 50, 10, 11, 28, 33, 6, 23, 56, 21, 36, 39, 49, 18, 34, 46),
(1, 27, 38, 8, 14, 47, 40, 16, 57, 15, 53, 7, 13, 19, 9, 55, 25, 44, 52)(2,
12, 48, 30, 37, 31, 24, 20, 45, 42, 3, 29, 35, 43, 41, 32, 54, 4, 17)(5,
11, 21, 46, 10, 56, 34, 50, 23, 18, 51, 6, 49, 26, 33, 39, 22, 28, 36),
(1, 52, 44, 25, 55, 9, 19, 13, 7, 53, 15, 57, 16, 40, 47, 14, 8, 38, 27)(2,
17, 4, 54, 32, 41, 43, 35, 29, 3, 42, 45, 20, 24, 31, 37, 30, 48, 12)(5,
36, 28, 22, 39, 33, 26, 49, 6, 51, 18, 23, 50, 34, 56, 10, 46, 21, 11),
(1, 24, 33)(2, 56, 55)(3, 11, 15)(4, 21, 27)(5, 44, 30)(6, 47, 37)(7, 43,
39)(8, 12, 22)(9, 45, 28)(10, 14, 35)(13, 31, 46)(16, 20, 34)(17, 26,
57)(18, 38, 42)(19, 54, 51)(23, 53, 48)(25, 29, 49)(32, 36, 40)(41, 50,
52),
(1, 37, 18)(2, 46, 53)(3, 36, 14)(4, 5, 55)(6, 8, 17)(7, 29, 51)(9, 24,
49)(10, 38, 20)(11, 16, 31)(12, 33, 44)(13, 30, 28)(15, 45, 39)(19, 41,
56)(21, 52, 35)(22, 27, 32)(23, 57, 54)(25, 42, 50)(26, 40, 43)(34, 47,
48),
(1, 45, 5)(2, 50, 27)(3, 46, 9)(4, 10, 40)(6, 16, 42)(7, 32, 11)(8, 30,
 21)(12, 36, 57)(13, 20, 23)(14, 41, 18)(15, 35, 34)(17, 39, 19)(22, 47,
 24)(25, 43, 28)(26, 53, 37)(29, 33, 38)(31, 56, 44)(48, 51, 55)(49, 52,
 54),
(1, 48, 10)(2, 11, 47)(3, 22, 52)(4, 28, 53)(5, 19, 35)(6, 27, 43)(7, 42,
 56)(8, 54, 34)(9, 37, 50)(12, 49, 13)(14, 45, 26)(15, 24, 51)(16, 30,
 39)(17, 18, 44)(20, 21, 25)(23, 40, 29)(31, 36, 38)(32, 33, 55)(41, 46,
 57),
(1, 3, 56)(2, 18, 40)(4, 34, 7)(5, 38, 43)(6, 15, 41)(8, 31, 50)(9, 35,
 23)(10, 57, 30)(11, 19, 12)(13, 42, 26)(14, 54, 33)(16, 29, 22)(17, 28,
 52)(20, 51, 44)(21, 47, 45)(24, 36, 53)(25, 32, 46)(27, 48, 49)(37, 39,
 55),
(1, 2, 36)(3, 33, 19)(4, 39, 47)(5, 7, 20)(6, 52, 45)(8, 41, 11)(9, 48,
 21)(10, 44, 54)(12, 51, 16)(13, 17, 50)(14, 24, 23)(15, 37, 56)(18, 55,
 43)(22, 25, 31)(26, 38, 30)(27, 29, 34)(28, 57, 35)(32, 49, 53)(40, 42,
 46),
(1, 35, 51)(2, 6, 7)(3, 50, 47)(4, 23, 25)(5, 14, 17)(8, 20, 49)(9, 41,
 26)(10, 53, 45)(11, 55, 24)(12, 46, 52)(13, 29, 36)(15, 54, 22)(16, 43,
 21)(18, 57, 31)(19, 30, 34)(27, 37, 28)(32, 56, 38)(33, 40, 48)(39, 44,
 42),
(1, 4, 26)(2, 22, 9)(3, 49, 57)(5, 15, 48)(6, 25, 30)(7, 31, 33)(8, 35,
 39)(10, 55, 29)(11, 27, 42)(12, 23, 38)(13, 54, 21)(14, 37, 46)(16, 17,
 56)(18, 19, 45)(20, 28, 40)(24, 34, 52)(32, 51, 47)(36, 44, 41)(43, 50,
 53),
(1, 41, 39)(2, 26, 25)(3, 18, 53)(4, 51, 38)(5, 40, 37)(6, 19, 31)(7, 48,
 22)(8, 42, 28)(9, 54, 36)(10, 13, 43)(11, 44, 29)(12, 56, 14)(15, 17,
```

```
 21)(16, 32, 50)(20, 33, 57)(23, 52, 30)(24, 46, 27)(34, 55, 45)(35, 49,
 47),
(1, 32, 23)(2, 33, 15)(3, 51, 8)(4, 6, 9)(5, 16, 54)(7, 30, 18)(10, 19,
 24)(11, 52, 37)(12, 34, 25)(13, 41, 22)(14, 48, 28)(17, 46, 38)(20, 39,
 27)(21, 53, 29)(26, 44, 35)(31, 49, 40)(36, 55, 42)(43, 56, 47)(45, 50,
 57),
(1, 54, 11)(2, 39, 38)(3, 6, 55)(4, 49, 16)(5, 57, 42)(7, 37, 21)(8, 29,
 46)(9, 17, 10)(12, 50, 15)(13, 32, 18)(14, 30, 51)(19, 20, 22)(23, 27,
 45)(24, 56, 40)(25, 48, 36)(26, 52, 31)(28, 47, 41)(33, 53, 35)(34, 44,
 43),
 (1, 43, 46)(2, 49, 14)(3, 23, 44)(4, 18, 15)(5, 47, 29)(6, 13, 35)(7, 12,
 10)(8, 45, 56)(9, 32, 34)(11, 25, 17)(16, 41, 33)(19, 37, 36)(20, 26,
 55)(21, 57, 24)(22, 53, 42)(27, 31, 51)(28, 38, 54)(30, 50, 40)(39, 52,
 48),
(1, 17, 34)(2, 28, 16)(3, 26, 27)(4, 33, 13)(5, 53, 41)(6, 44, 32)(7, 24,
 50)(8, 43, 23)(9, 12, 18)(10, 25, 37)(11, 38, 48)(14, 31, 39)(15, 30,
 49)(19, 42, 21)(20, 36, 52)(22, 55, 35)(29, 56, 57)(40, 45, 51)(46, 47,
 54),
(1, 29, 28)(2, 51, 52)(3, 34, 13)(4, 50, 14)(5, 8, 24)(6, 53, 20)(7, 17,
 36)(9, 43, 11)(10, 15, 32)(12, 21, 40)(16, 35, 18)(19, 48, 26)(22, 57,
 37)(23, 55, 31)(25, 54, 39)(27, 30, 56)(33, 47, 42)(38, 41, 49)(44, 45,
 46),
(1, 12, 6)(2, 5, 13)(3, 39, 40)(4, 22, 44)(7, 45, 49)(8, 32, 26)(9, 30,
 33)(10, 52, 42)(11, 14, 20)(15, 31, 28)(16, 48, 46)(17, 23, 47)(18, 25,
 24)(19, 29, 50)(21, 55, 41)(27, 35, 36)(34, 38, 37)(43, 51, 57)(53, 54,
 56),
(1, 42, 49)(2, 23, 19)(3, 10, 16)(4, 56, 52)(5, 27, 12)(6, 57, 48)(7, 54,
 26)(8, 37, 33)(9, 29, 39)(11, 13, 45)(14, 32, 21)(15, 43, 36)(17, 22,
 40)(18, 47, 20)(24, 28, 44)(25, 41, 51)(30, 46, 55)(31, 34, 53)(35, 50,
 38),
(1, 30, 22)(2, 21, 44)(3, 28, 7)(4, 36, 8)(5, 9, 31)(6, 38, 24)(10, 27,
 41)(11, 40, 35)(12, 26, 47)(13, 48, 56)(14, 42, 34)(15, 20, 46)(16, 37,
 23)(17, 51, 53)(18, 52, 29)(19, 43, 49)(25, 45, 33)(32, 39, 57)(50, 55,
 54),
(1, 20, 50)(2, 34, 57)(3, 21, 38)(4, 46, 19)(5, 52, 32)(6, 40, 54)(7, 41,
 23)(8, 48, 18)(9, 42, 51)(10, 47, 31)(11, 53, 30)(12, 28, 55)(13, 24,
 39)(14, 43, 22)(15, 29, 26)(16, 45, 36)(17, 33, 27)(25, 35, 56)(37, 49,
 44),
(1, 31, 21)(2, 10, 8)(3, 5, 25)(4, 11, 57)(6, 14, 29)(7, 35, 46)(9, 20,
 56)(12, 39, 53)(13, 37, 51)(15, 42, 23)(16, 24, 26)(17, 49, 55)(18, 27,
 54)(19, 32, 28)(22, 38, 45)(30, 36, 47)(33, 52, 43)(34, 40, 41)(44, 48,
 50),
 (1, 33, 24)(2, 55, 56)(3, 15, 11)(4, 27, 21)(5, 30, 44)(6, 37, 47)(7, 39,
 43)(8, 22, 12)(9, 28, 45)(10, 35, 14)(13, 46, 31)(16, 34, 20)(17, 57,
 26)(18, 42, 38)(19, 51, 54)(23, 48, 53)(25, 49, 29)(32, 40, 36)(41, 52,
 50),
 (1, 49, 42)(2, 19, 23)(3, 16, 10)(4, 52, 56)(5, 12, 27)(6, 48, 57)(7, 26,
```

```
   54)(8, 33, 37)(9, 39, 29)(11, 45, 13)(14, 21, 32)(15, 36, 43)(17, 40,
   22)(18, 20, 47)(24, 44, 28)(25, 51, 41)(30, 55, 46)(31, 53, 34)(35, 38,
   50),
  (1, 22, 30)(2, 44, 21)(3, 7, 28)(4, 8, 36)(5, 31, 9)(6, 24, 38)(10, 41,
   27)(11, 35, 40)(12, 47, 26)(13, 56, 48)(14, 34, 42)(15, 46, 20)(16, 23,
   37)(17, 53, 51)(18, 29, 52)(19, 49, 43)(25, 33, 45)(32, 57, 39)(50, 54,
   55),
  (1, 51, 35)(2, 7, 6)(3, 47, 50)(4, 25, 23)(5, 17, 14)(8, 49, 20)(9, 26,
   41)(10, 45, 53)(11, 24, 55)(12, 52, 46)(13, 36, 29)(15, 22, 54)(16, 21,
   43)(18, 31, 57)(19, 34, 30)(27, 28, 37)(32, 38, 56)(33, 48, 40)(39, 42,
   44),
  (1, 36, 2)(3, 19, 33)(4, 47, 39)(5, 20, 7)(6, 45, 52)(8, 11, 41)(9, 21,
   48)(10, 54, 44)(12, 16, 51)(13, 50, 17)(14, 23, 24)(15, 56, 37)(18, 43,
   55)(22, 31, 25)(26, 30, 38)(27, 34, 29)(28, 35, 57)(32, 53, 49)(40, 46,
   42)
]
> PDset:=PDset43;
> load "qudecodeR.m";
Loading "qudecodeR.m"
check matrix
sent.....
(2 5 5 4 5 0 6 0 3 6 3 6 2 3 6 5 3 0 6 4 0 2 3 1 6 3 6 3 3 3 5 3 0 5 6 4
 0 3 5 6 2 5 5 5 5 5 4 0 5 2 1 5 6 4 3 5 2)
received.
(2 5 5 4 6 0 6 0 3 6 2 6 2 3 6 5 3 0 6 4 0 2 3 1 6 3 6 3 3 3 4 3 0 5 6 4
 0 3 5 6 2 5 5 5 5 5 4 0 5 2 1 5 6 4 3 5 2)
32 th pdset elt
(1, 17, 34)(2, 28, 16)(3, 26, 27)(4, 33, 13)(5, 53, 41)(6, 44, 32)(7, 24, 50)
(8, 43, 23)(9, 12, 18)(10, 25, 37)(11, 38, 48)(14, 31, 39)(15, 30, 49)(19, 42,
 21)(20, 36, 52)(22, 55, 35)(29, 56, 57)(40, 45, 51)(46, 47, 54)
3 errors
corrected (2 5 5 4 5 0 6 0 3 6 3 6 2 3 6 5 3 0 6 4 0 2 3 1 6 3 6 3 3 3 5 3 0 5 6 4
    0 3 5 6 2 5 5 5 5 5 4 0 5 2 1 5 6 4 3 5 2)
It is true that the corrected vector is the sent word
------------
sent..... (2 0 4 4 3 1 3 0 3 3 1 0 6 5 5 1 6 5 1 2 2 2 5 5 5 6 3 1 4 3 6 1 4 1 2 4
    4 3 2 0 2 0 0 3 0 3 5 0 0 0 1 5 2 0 5 3 6)
received. (0 0 4 4 3 1 3 0 3 3 1 0 0 5 6 1 6 5 1 2 2 2 5 5 5 6 3 1 4 3 6 1 4 1 2 4
    4 3 2 0 2 0 0 3 0 3 5 0 0 0 1 5 2 0 5 3 6)
4 th pdset elt
(1, 55, 7, 16, 8, 52, 9, 53, 40, 38, 44, 19, 15, 47, 27, 25, 13, 57, 14)(2, 32,
    29, 20, 30, 17, 41, 3, 24, 48, 4, 43, 42, 31, 12, 54, 35, 45, 37)(5, 39, 6,
    50, 46, 36, 33, 51, 34, 21, 28, 26, 18, 56, 11, 22, 49, 23, 10)
3 errors
corrected (2 0 4 4 3 1 3 0 3 3 1 0 6 5 5 1 6 5 1 2 2 2 5 5 5 6 3 1 4 3 6 1 4 1 2 4
    4 3 2 0 2 0 0 3 0 3 5 0 0 0 1 5 2 0 5 3 6)
It is true that the corrected vector is the sent word
```

```
------------
sent..... (5 3 6 6 6 2 0 3 0 0 4 5 2 3 4 6 3 2 3 3 1 2 5 2 5 4 3 0 4 4 6 5 2 0 1 1
    5 3 4 6 1 0 0 6 0 1 5 1 3 6 3 3 5 2 5 6 0)
received. (5 3 6 6 6 1 0 3 0 0 4 5 2 3 4 6 3 2 3 3 1 2 5 2 5 4 3 0 4 4 6 5 2 0 1 1
    5 3 4 6 1 0 0 6 0 1 5 1 3 6 0 3 5 2 5 6 0)
4 th pdset elt
(1, 55, 7, 16, 8, 52, 9, 53, 40, 38, 44, 19, 15, 47, 27, 25, 13, 57, 14)(2, 32,
    29, 20, 30, 17, 41, 3, 24, 48, 4, 43, 42, 31, 12, 54, 35, 45, 37)(5, 39, 6,
    50, 46, 36, 33, 51, 34, 21, 28, 26, 18, 56, 11, 22, 49, 23, 10)
2 errors
corrected (5 3 6 6 6 2 0 3 0 0 4 5 2 3 4 6 3 2 3 3 1 2 5 2 5 4 3 0 4 4 6 5 2 0 1 1
    5 3 4 6 1 0 0 6 0 1 5 1 3 6 3 3 5 2 5 6 0)
It is true that the corrected vector is the sent word
------------
sent..... (0 1 1 2 2 4 0 5 4 5 0 3 5 4 3 3 2 4 6 1 6 0 0 5 3 2 4 5 0 3 1 2 1 6 6 1
    6 1 3 2 2 2 6 4 2 3 2 1 1 2 0 2 3 5 3 4 1)
received. (0 1 1 2 2 4 0 5 4 5 0 3 5 4 3 3 2 4 6 1 6 0 0 5 3 2 4 5 0 3 1 2 1 6 6 1
    6 1 3 2 2 6 6 4 2 3 2 1 1 2 0 2 3 5 3 0 1)
1 th pdset elt
Id(nhx)
2 errors
corrected (0 1 1 2 2 4 0 5 4 5 0 3 5 4 3 3 2 4 6 1 6 0 0 5 3 2 4 5 0 3 1 2 1 6 6 1
    6 1 3 2 2 2 6 4 2 3 2 1 1 2 0 2 3 5 3 4 1)
It is true that the corrected vector is the sent word
------------
sent..... (5 3 6 0 0 5 2 0 0 6 4 3 4 2 4 3 4 2 4 0 0 1 3 1 5 5 3 5 2 1 1 4 6 4 3 1
    4 6 1 6 6 4 3 1 3 0 2 1 2 2 2 6 6 4 5 2 0)
received. (5 3 6 4 0 5 2 0 0 6 4 3 4 2 4 1 4 2 4 0 0 1 3 1 5 5 3 5 2 1 1 4 6 4 3 1
    4 6 3 6 6 4 3 1 3 0 2 1 2 2 2 6 6 4 5 2 0)
8 th pdset elt
(1, 7, 8, 9, 40, 44, 15, 27, 13, 14, 55, 16, 52, 53, 38, 19, 47, 25, 57)(2, 29,
    30, 41, 24, 4, 42, 12, 35, 37, 32, 20, 17, 3, 48, 43, 31, 54, 45)(5, 6, 46,
    33, 34, 28, 18, 11, 49, 10, 39, 50, 36, 51, 21, 26, 56, 22, 23)
3 errors
corrected (5 3 6 0 0 5 2 0 0 6 4 3 4 2 4 3 4 2 4 0 0 1 3 1 5 5 3 5 2 1 1 4 6 4 3 1
    4 6 1 6 6 4 3 1 3 0 2 1 2 2 2 6 6 4 5 2 0)
It is true that the corrected vector is the sent word
------------
> load "qudecodeR.m";
Loading "qudecodeR.m"
check matrix
sent..... (6 5 2 0 2 4 0 3 4 2 5 3 3 1 6 3 5 0 0 6 2 3 2 6 5 1 3 4 6 0 3 1 0 1 5 1
    5 6 4 0 0 0 1 1 4 4 0 5 4 3 0 4 6 2 1 5 3)
received. (6 5 2 0 2 4 0 3 4 2 5 3 3 1 6 6 5 0 0 6 2 3 2 6 5 1 3 4 6 0 3 1 0 1 5 1
    5 6 4 0 0 0 1 1 4 4 0 5 4 3 0 4 6 2 2 5 3)
11 th pdset elt
(1, 53, 27, 7, 38, 13, 8, 19, 14, 9, 47, 55, 40, 25, 16, 44, 57, 52, 15)(2, 3, 12,
```

```
    29, 48, 35, 30, 43, 37, 41, 31, 32, 24, 54, 20, 4, 45, 17, 42)(5, 51, 11, 6,
    21, 49, 46, 26, 10, 33, 56, 39, 34, 22, 50, 28, 23, 36, 18)
2 errors
corrected (6 5 2 0 2 4 0 3 4 2 5 3 3 1 6 3 5 0 0 6 2 3 2 6 5 1 3 4 6 0 3 1 0 1 5 1
    5 6 4 0 0 0 1 1 4 4 0 5 4 3 0 4 6 2 1 5 3)
It is true that the corrected vector is the sent word
------------
sent..... (4 1 5 4 5 6 2 3 0 3 0 3 4 1 2 2 0 5 3 5 0 2 1 5 1 0 2 1 5 3 4 0 6 5 2 0
    1 0 1 6 6 3 2 4 1 5 0 6 4 3 0 3 4 5 0 6 1)
received. (4 1 5 4 5 6 2 3 0 3 0 3 4 1 2 2 0 5 3 5 6 2 1 5 4 0 2 1 5 3 4 0 6 5 2 6
    1 0 1 6 6 3 2 4 1 5 0 6 4 3 0 3 4 5 0 6 1)
6 th pdset elt
(1, 19, 16, 27, 9, 57, 38, 55, 15, 8, 25, 53, 14, 44, 7, 47, 52, 13, 40)(2, 43,
    20, 12, 41, 45, 48, 32, 42, 30, 54, 3, 37, 4, 29, 31, 17, 35, 24)(5, 26, 50,
    11, 33, 23, 21, 39, 18, 46, 22, 51, 10, 28, 6, 56, 36, 49, 34)
3 errors
corrected (4 1 5 4 5 6 2 3 0 3 0 3 4 1 2 2 0 5 3 5 0 2 1 5 1 0 2 1 5 3 4 0 6 5 2 0
    1 0 1 6 6 3 2 4 1 5 0 6 4 3 0 3 4 5 0 6 1)
It is true that the corrected vector is the sent word
------------
sent..... (4 5 4 5 0 6 6 6 5 3 6 2 0 2 1 5 5 3 5 4 4 0 4 1 1 0 5 5 6 4 0 2 6 5 5 3
    1 1 0 4 1 2 5 4 0 3 0 6 5 2 4 3 2 6 1 2 1)
received. (4 5 4 5 0 6 6 6 5 3 6 2 0 2 5 5 5 3 5 4 4 0 4 1 1 0 5 5 6 4 0 2 6 5 5 3
    1 1 0 4 1 2 5 4 0 3 0 6 5 2 6 3 2 6 1 2 1)
4 th pdset elt
(1, 55, 7, 16, 8, 52, 9, 53, 40, 38, 44, 19, 15, 47, 27, 25, 13, 57, 14)(2, 32,
    29, 20, 30, 17, 41, 3, 24, 48, 4, 43, 42, 31, 12, 54, 35, 45, 37)(5, 39, 6,
    50, 46, 36, 33, 51, 34, 21, 28, 26, 18, 56, 11, 22, 49, 23, 10)
2 errors
corrected (4 5 4 5 0 6 6 6 5 3 6 2 0 2 1 5 5 3 5 4 4 0 4 1 1 0 5 5 6 4 0 2 6 5 5 3
    1 1 0 4 1 2 5 4 0 3 0 6 5 2 4 3 2 6 1 2 1)
It is true that the corrected vector is the sent word
------------
sent..... (1 0 1 2 1 3 0 1 5 1 4 3 1 0 3 4 4 3 5 0 5 4 0 6 3 2 3 1 6 2 4 2 2 0 4 5
    0 2 0 6 0 3 3 1 4 5 0 2 3 0 0 1 5 0 3 3 3)
received. (1 0 1 2 1 3 0 1 5 1 4 2 1 0 3 4 4 3 5 0 5 4 0 6 3 2 3 1 5 2 4 4 2 0 4 5
    0 2 0 6 0 3 3 1 4 5 0 2 3 0 0 1 5 0 3 3 3)
6 th pdset elt
(1, 19, 16, 27, 9, 57, 38, 55, 15, 8, 25, 53, 14, 44, 7, 47, 52, 13, 40)(2, 43,
    20, 12, 41, 45, 48, 32, 42, 30, 54, 3, 37, 4, 29, 31, 17, 35, 24)(5, 26, 50,
    11, 33, 23, 21, 39, 18, 46, 22, 51, 10, 28, 6, 56, 36, 49, 34)
3 errors
corrected (1 0 1 2 1 3 0 1 5 1 4 3 1 0 3 4 4 3 5 0 5 4 0 6 3 2 3 1 6 2 4 2 2 0 4 5
    0 2 0 6 0 3 3 1 4 5 0 2 3 0 0 1 5 0 3 3 3)
It is true that the corrected vector is the sent word
------------
sent..... (2 0 3 3 5 3 3 4 0 2 4 2 6 2 6 2 5 6 6 3 2 1 2 1 6 3 4 4 6 5 5 5 4 5 6 2
```

```
    1 5 6 1 4 6 1 6 4 6 6 3 5 2 0 0 1 6 4 3 6)
received. (2 1 3 3 5 3 3 4 0 2 4 2 6 2 6 2 5 6 6 3 2 1 2 1 6 3 4 4 6 1 5 5 4 5 6 2
    1 5 6 1 1 6 1 6 4 6 6 3 5 2 0 0 1 6 4 3 6)
3 th pdset elt
(1, 38, 14, 40, 57, 53, 13, 9, 25, 52, 27, 8, 47, 16, 15, 7, 19, 55, 44)(2, 48,
    37, 24, 45, 3, 35, 41, 54, 17, 12, 30, 31, 20, 42, 29, 43, 32, 4)(5, 21, 10,
    34, 23, 51, 49, 33, 22, 36, 11, 46, 56, 50, 18, 6, 26, 39, 28)
3 errors
corrected (2 0 3 3 5 3 3 4 0 2 4 2 6 2 6 2 5 6 6 3 2 1 2 1 6 3 4 4 6 5 5 5 4 5 6 2
    1 5 6 1 4 6 1 6 4 6 6 3 5 2 0 0 1 6 4 3 6)
It is true that the corrected vector is the sent word
------------
>
> nhx;
Permutation group nhx acting on a set of cardinality 57
Order = 171 = 3^2 * 19
    (1, 38, 14, 40, 57, 53, 13, 9, 25, 52, 27, 8, 47, 16, 15, 7, 19, 55, 44)(2,
        48, 37, 24, 45, 3, 35, 41, 54, 17, 12, 30, 31, 20, 42, 29, 43, 32, 4)(5,
        21, 10, 34, 23, 51, 49, 33, 22, 36, 11, 46, 56, 50, 18, 6, 26, 39, 28)
    (3, 43, 32)(4, 31, 54)(5, 34, 51)(6, 50, 28)(7, 27, 16)(8, 38, 9)(10, 39,
        22)(11, 26, 21)(12, 20, 29)(13, 57, 52)(14, 15, 40)(17, 35, 45)(18, 49,
        33)(19, 55, 53)(23, 56, 46)(24, 37, 42)(25, 44, 47)(30, 48, 41)
    (1, 33, 24)(2, 55, 56)(3, 15, 11)(4, 27, 21)(5, 30, 44)(6, 37, 47)(7, 39,
        43)(8, 22, 12)(9, 28, 45)(10, 35, 14)(13, 46, 31)(16, 34, 20)(17, 57,
        26)(18, 42, 38)(19, 51, 54)(23, 48, 53)(25, 49, 29)(32, 40, 36)(41, 52,
        50)
> Seqset(PDset43) subset {x:x in nhx};
true
> quit;

Total time: 21.300 seconds
```

qudecodeR.m

```
//values for  blox= blocks of design,p=p-ary code
//C= code of design over GF(p),pdset=PD set for code

IV:=func< v, block,p |
 CharacteristicVector(VectorSpace(GF(p),v), block) > ;

ba1:=Basis(C);
seq:=[];
d:=Dimension(C);
v:=Length(C);
for j:=1 to d do
 b:=ba1[j];
  seq:=seq cat [b[k]: k in [d+1..v]];
```

```
 end for;

ma:=KMatrixSpace(GF(p),d,v-d);
sm:=ma!seq;
smt:=-Transpose(sm);
z:=[0:j in [1..v-d]];
seqn:=[];
 for j:=1 to v-d do
  r:=[smt[j][i]:i in [1..d]];
  zj:=z;
  zj[j]:=1;
   rc:=r cat zj;
 seqn:=seqn cat rc;
end for;
cseq:=&cat[Eltseq(ba1[i]):i in [1..d]];
man:=KMatrixSpace(GF(p),v-d,v);
ma1:=KMatrixSpace(GF(p),1,d);
ma2:=KMatrixSpace(GF(p),d,v);
ma3:=KMatrixSpace(GF(p),1,v-d);
hsmt:=man!seqn;
"check matrix";
H:=hsmt;

cs:=ma2!cseq;
kset:={};
for i:=1 to 5 do
bb:=Random(C);
ers:=[];
for i:=1 to t do
ni:={Random({1..v})};
ai:=Random(f);
cc:=IV(v,ni,p);
ers:=Append(ers,ai*cc);
end for;

b:=bb + &+[ers[i]:i in [1..t]];
"sent.....",bb;
"received.",b;

for k:=1 to #pdset do
e:=PDset[k];
seq1:=[];
for i:=1 to v-d do
seq1:=Append(seq1, InnerProduct(b^e,hsmt[i]));
end for;
s:={i:i in [1..v-d]|seq1[i] ne 0};
if #s le t then
```

```
k,"th pdset elt";
e;
#s,"errors";
kset:=kset join {k};
bee:=b^e;
aseq:=[bee[i]:i in [1..d]];
vv:=ma1!aseq;
r:=C!(vv*cs);
"corrected",r^(e^-1);
"It is",r^(e^-1) eq bb,"that the corrected vector is the sent word";
break k;
end if;
end for;
"------------";
end for;
```

References

[1] E. F. Assmus, Jr. and J. D. Key. *Designs and their Codes*. Cambridge: Cambridge University Press, 1992. Cambridge Tracts in Mathematics, Vol. 103 (Second printing with corrections, 1993).

[2] E. F. Assmus, Jr. and J. D. Key. Designs and codes: an update. *Des. Codes Cryptogr.*, 9:7–27, 1996.

[3] Wieb Bosma and John Cannon. *Handbook of Magma Functions*. Department of Mathematics, University of Sydney, November 1994. http://www.maths.usyd.edu.au:8000/u/magma/.

[4] A. E. Brouwer and C. J. van Eijl. On the p-rank of the adjacency matrices of strongly regular graphs. *J. Algebraic Combin.*, 1:329–346, 1992.

[5] A. E. Brouwer and J.H. van Lint. Strongly regular graphs and partial geometries. In D.M. Jackson and S.A. Vanstone, editors, *Enumeration and Design*, pages 85–122. Toronto: Academic Press, 1984. Proc. Silver Jubilee Conf. on Combinatorics, Waterloo, 1982.

[6] Hervé Chabanne. Permutation decoding of abelian codes. *IEEE Trans. Inform. Theory*, 38:1826–1829, 1992.

[7] D. M. Gordon. Minimal permutation sets for decoding the binary Golay codes. *IEEE Trans. Inform. Theory*, 28:541–543, 1982.

[8] Willem H. Haemers, René Peeters, and Jeroen M. van Rijckevorsel. Binary codes of strongly regular graphs. *Des. Codes Cryptogr.*, 17:187–209, 1999.

[9] W. Cary Huffman. Codes and groups. In V. S. Pless and W. C. Huffman, editors, *Handbook of Coding Theory*, pages 1345–1440. Amsterdam: Elsevier, 1998. Volume 2, Part 2, Chapter 17.

[10] J. D. Key, J. Moori, and B. G. Rodrigues. Binary codes of triangular graphs and permutation decoding. Submitted.

[11] J. D. Key, J. Moori, and B. G. Rodrigues. On some designs and codes from primitive representations of some finite simple groups. J. Combin. Math and Combin. Comput. To appear.

[12] F. J. MacWilliams. Permutation decoding of systematic codes. *Bell System Tech. J.*, 43:485–505, 1964.

[13] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1983.

[14] J. Schönheim. On coverings. *Pacific J. Math.*, 14:1405–1411, 1964.

[15] Vladimir D. Tonchev. *Combinatorial Configurations, Designs, Codes, Graphs*. Pitman Monographs and Surveys in Pure and Applied Mathematics, No. 40. New York: Longman, 1988. Translated from the Bulgarian by Robert A. Melter.

[16] J. Wolfmann. A permutation decoding of the (24,12,8) Golay code. *IEEE Trans. Inform. Theory*, 29:748–750, 1983.