Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(4): 8398–8421.
DOI: 10.3934/math.2025387
Received: 02 December 2024
Revised: 20 March 2025
Accepted: 24 March 2025
Published: 11 April 2025

Research article

Ultimate linear block and convolutional codes

Ted Hurley*

University of Galway (previously National University of Ireland Galway), Ireland

* **Correspondence:** Email: Ted.Hurley@universityofgalway.ie.

Abstract: Linear block and convolutional codes are designed using unit schemes and families of these to required length, rate, distance and type are mined. Properties, such as type and distance, of the codes follow from the types of units used and thus required codes are built from specific units. Orthogonal units, units in group rings, Fourier/Vandermonde units and related units are used to construct and analyse linear block and convolutional codes and to construct these to predefined length, rate, distance and type. Series of self-dual, dual containing, quantum error-correcting and linear complementary dual codes are constructed for both linear block and convolutional codes. Low density parity check linear block and linear convolutional codes are constructed from unit schemes.

Keywords: code; linear; convolutional; self-dual; dual-containing; quantum code; complementary dual; LDPC
Mathematics Subject Classification: 15B99, 94B05, 94B10

1. Introduction

Unit-derived methods for coding theory were initiated in [21, 22] and further developed in [13, 14, 25]; an introduction in chapter form is available in [23, 24]. All linear block codes over fields are equivalent to unit-derived codes (see, for example Proposition 2.1 below) and using units of different types enables the construction and analysis of linear block codes to required length, rate, distance and type.

The method extends to the design of convolutional codes and to build these to type, length and rate as required. The lack of any algebraic methods for constructing convolutional codes has long been a problem – for example McEliece [33] remarks: "A most striking fact is the lack of algebraic constructions of families of convolutional codes", Blahut ([3], p.312), writes: "No general method is yet known for constructing a family of high performance multiple-error-correcting (convolutional) codes. . . The codes now in common use have been found by computer search". The computer search is not practical except for small length.

Infinite families of codes with rates approaching a given rational R, $(0 < R < 1)$, and relative distances approaching $(1 - R)$ can be designed by the methods. Some codes derived have applications in solving underdetermined systems of equations (see [19]).

The (free) distance of a convolutional code designed from a unit is much better than that of the distance of the linear block code of the same rate designed. An illuminating example of this is given in Example 3.9 below in which a unit-derived form for the Hamming $[7, 4, 3]$ binary code is obtained and from this a memory 1 convolutional (binary) code of the same length and rate but with twice the distance is derived.* The process is general and a unit-derived form for a linear block code can be used to derive convolutional codes of the same length and rate as the block code but with much better distance.

New families of linear block and convolutional codes are now available and such codes to required length, rate and type required are mined by the unit-derived method. Codes are formed from within a unit structure and properties of the codes flow from the properties of the substructures of the unit used. In mathematics we often think of breaking a structure into more manageable parts or building a structure from already known structures but here we think of looking at the bigger (unit) structure already formed with which embedded substructures are mined to create codes and codes to specific requirements.

Using special types of units such as orthogonal units, Vandermonde/Fourier units or units in group rings allows the construction of codes such as self-dual codes, dual containing codes, complementary dual codes and quantum code as well as these to specified lengths, rates, and distances. Codes over particular required fields, such as over fields of characteristic 2 or codes over prime fields (for which modular arithmetic is available) and families of such are constructible by the methods. Special linear block and convolutional codes may also be induced from Hadamard matrices; an Example 4.4 is given, but the general Hadamard case is dealt with separately in [16]. These can have large rates, as opposed to Walsh-Hadamard matrices, but note that now convolutional codes related to Hadamard matrices are constructed.

Using units in group rings allows the construction of low density parity check (LDPC) linear block and LDPC convolutional codes and families of such. These are constructed with no short cycles in the control matrix, a huge advantage in decoding LDPC codes.

Dual-containing (DC) codes have their own intrinsic interest but are also used for designing quantum error-correcting codes by the Calderbank-Shor-Steane (CSS) method, [5, 6, 40]. Here then families of convolutional quantum error correcting codes of different lengths and rates are explicitly constructed. Linear complementary dual (LCD) codes have been studied extensively in the literature. For background, history and general theory on LCD codes, consult the articles [7–9, 34] by Carlet, Mesnager, Tang, Qi and Pelikaan. LCD codes were originally introduced by Massey in [30, 31]. These codes have been studied amongst other things for improving the security of information on sensitive devices against side-channel attacks (SCA) and fault non-invasive attacks (see [10]) and have found use in data storage and communications' systems.

The relationships between DC linear block codes and LCD convolutional codes and between LCD linear block codes and convolutional DC codes when formed from the same unit scheme are quite remarkable.

Hermitian codes over fields of the form $GF(q^2)$ may be formed by looking at unitary matrices over

*The convolutional code obtained has parameters $(7, 4, 4; 1, 6)$; see Section 1.1 for explanation of convolutional code parameters.

these fields; this was initiated in [14], and further development is left to later work.

Requiring one of $\{U, V\}$ in $UV = I$ to be of low density enables the construction of LDPC linear block and convolutional LDPC codes by the unit-derived method. These are constructed so that there are no short cycles in the control matrix using units in group rings. The linear block case for such LDPC codes has been dealt with in [26] and the convolutional case follows from the unit-derived method. Iterative decoders for low density parity check codes are impacted by short cycles. Here for a given unit scheme, described in a precise way, multiple such codes, all with no short cycles, are constructed and with prescribed rate and dimension. These LDPC codes have many applications and can further be stored using an algebraic 'short' formula which, for example, is important in applications requiring low storage and low power.

Some of this work is additional to and complementary to that in [15, 20]. As pointed out in [15], convolutional codes which appeared previously in the literature are very special cases of constructions using unit-derived methods and now whole families and required types are available.

The unit-derived method allows the construction of multiple linear block codes and multiple convolutional codes from the same unit. Methods are introduced for constructing families of self-dual codes from units by adding if necessary a square root of (-1) to the field over which the unit is defined.

A number of examples are given here which of necessity are of relatively small length; these can be looked upon as prototype examples for large length constructions which are attainable. The codes are easily implementable once the units from which they are derived are formed. The brilliant computer algebra system GAP [42] proves extremely useful in constructing examples and verifying properties.

Coding theory background is contained in Subsection 1.1 of this introduction. The contents of other sections are given in Subsection 1.4.

1.1. Background on coding theory

Basics on linear block coding theory may be found in any of [3, 27, 29, 33] and many others. The notation $[n, r, d]$ is used here for a linear block code of length n, dimension r, and (minimum) distance d. The rate is then $\mathcal{F}rn$. A maximum distance separable (MDS) linear block code is one of the form $[n, r, n - r + 1]$, where the maximum distance possible for the length and rate is achieved.

Different equivalent definitions for convolutional codes are given in the literature. The notation and definitions used here follow that given in [38, 39, 41]. A rate $\frac{k}{n}$ convolutional code with parameters (n, k, δ) over a field $\mathcal{F}$ is a submodule of $\mathcal{F}[z]^n$ generated by a reduced basic matrix $G[z] = (g_{ij}) \in \mathcal{F}[z]^{r\times n}$ of rank r where n is the length, $\delta = \sum_{i=1}^{r} \delta_i$ is the degree with $\delta_i = \max_{1\leq j\leq r} \deg g_{ij}$. Also $\mu = \max_{1\leq i\leq r} \delta_i$ is known as the memory of the code and then the code may then be given with parameters $(n, k, \delta; \mu)$. The parameters $(n, r, \delta; \mu, d_f)$ are used for such a code with free (minimum) distance d_f.

Suppose C is a convolutional code in $\mathcal{F}[z]^n$ of rank k. A generating matrix $G[z] \in \mathcal{F}[z]_{k\times n}$ of C having rank k is called a generator or encoder matrix of C. A matrix $H \in \mathcal{F}[z]_{n\times(n-k)}$ satisfying $C = \ker H = \{v \in \mathcal{F}[z]^n : vH = 0\}$ is said to be a control matrix or check matrix of the code C.

Convolutional codes can be catastrophic or non-catastrophic; see, for example [33], for the basic definitions. A catastrophic convolutional code is prone to catastrophic error propagation and is not much use. A convolutional code described by a generator matrix with right polynomial inverse is a non-catastrophic code; this is sufficient for our purposes. The designs given here for the generator matrices allow for specifying directly the control matrices and the right polynomial inverses where appropriate. There exist very few algebraic constructions for designing convolutional codes and search methods limit their size and availability, see McEliece [33] for discussion and also [1, 2, 11, 37].

By Rosenthal and Smarandache [38] the maximum free distance attainable by an (n, r, δ) convolutional code is $(n-r)(\lfloor \frac{\delta}{r} \rfloor + 1) + \delta + 1$. The case $\delta = 0$, which is the case of zero memory, corresponds to the linear Singleton bound $(n - r + 1)$. The bound $(n-r)(\lfloor \frac{\delta}{r} \rfloor + 1) + \delta + 1$ is then called the generalised Singleton bound [38] (GSB), and a convolutional code attaining this bound is known as an MDS convolutional code. The papers [38] and [41] are major beautiful contributions to this area.

The criteria for a convolutional code to be an MDS code are given in terms of the parameters for a convolutional code and the criteria for a linear block code to be an MDS code are given in terms of the parameters for a linear block code.

Let $G(z)$ be the generator matrix for a convolutional code C with memory m. Suppose $G(z)H^{\mathrm{T}}(z) = 0$, so that $H^{\mathrm{T}}(z)$ is a control matrix, and then $H(z^{-1})z^m$ generates the convolutional dual code of C, see [4] and [12]. This is also known as the module-theoretic dual code.† The code is then dual-containing provided the code generated by $H(z^{-1})z^m$ is contained in the code generated by $G(z)$. The code is self-dual provided the code generated by $H(z^{-1})z^m$ is equal to the code generated by $G(z)$.

Let $G(z)$ be the generator matrix for a convolutional (n, r) code C. Code words will consist of $P(z)G(z)$ where $P(z)$ is a polynomial in z with coefficients which are $1 \times r$ vectors. The polynomial $P(z)$ is said to be an information vector for the code C. The support of $P(z)$ is the number of non-zero coefficient vectors appearing in its expression as a polynomial.

1.2. Self-dual, dual-containing and linear complementary dual codes

The dual of a code C is denoted by $C^{\perp}$. Note the definition of the dual code of a convolutional code is as given in Subsection 1.1 above. A self-dual code is a code equal to its dual. A code C is said to be dual containing, written DC, if it contains its dual $C^{\perp}$. Say a code is a linear complementary dual, written LCD, code provided it has trivial intersection with its dual.

A self-dual code is a particular type of dual-containing code. Thus

C is a dual containing (DC) code $\Longleftrightarrow C \cap C^{\perp} = C^{\perp}$;

C is a linear complementary dual (LCD) code $\Longleftrightarrow C \cap C^{\perp} = 0$;

C is a self-dual code provided $C = C^{\perp}$.

Constructions of DC, including self-dual, convolutional codes and LCD convolutional codes were initiated in [14]. DC codes are theoretically interesting in themselves but in addition a DC code enables the construction of a quantum error-correcting code by the CSS method.

LCD codes and DC codes are 'supplemental' to one another: C is DC if and only if $C \cap C^{\perp} = C^{\perp}$ and C is LCD if and only if $C \cap C^{\perp} = 0$. As noted in [14], MDS DC block linear codes lead to the construction of MDS LCD convolutional codes and LCD MDS block linear codes leads to the construction of MDS DC convolutional codes.

Main abbreviations used:

DC: dual-containing;

MDS: maximum distance separable; ‡

LCD: linear complementary dual;

†In convolutional coding theory, the idea of dual code has two meanings. The other dual convolutional code defined is called the sequence space dual; the generator matrices for these two types are related by a specific formula.

‡To be MDS has different parameter requirements for block codes, convolutional codes and quantum codes.

QECC: quantum error-correcting code;
LDPC: low density parity check.

1.3. Decoding

Many decoding schemes exist for linear block codes such as syndrome decoding [3]. Efficient decoding techniques for unit-derived linear block codes from Fourier/Vandermonde type matrices are established in [25], Algorithms 6.1–6.3. The constructions quickly lead to the establishment of error-correcting pairs for the codes; error-correcting pairs are due to Pellikan [36]. The algorithms are especially useful for solving underdetermined systems using error-correcting codes, see [19]. Several efficient algorithms exist for decoding convolutional codes, the most common ones being the Viterbi algorithm and the sequential decoding algorithm. Algebraic decoding schemes for convolutional codes are being developed.

1.4. Technical sections' contents

- Section 2: The unit-derived method in general is described.
- Section 2.1: Orthogonal unit schemes are used to design codes and families of codes, including general methods for designing LCD and self-dual codes.
- Section 2.2: Fourier type units are used to construct families of MDS, LCD, DC and QECC codes.
- Section 3: Methods for designing families of convolutional codes are devised. In particular families of DC, self-dual, and QECC convolutional codes are designed.
- Section 4: Series of higher memory convolutional codes are constructed.
- Section 5: General algorithms using units are described for constructing LDPC linear block and convolutional codes and to design these with no short cycles in the control matrix.

The designs are general in nature and series with large lengths and distances are achievable. Each technical section contains explicit examples and prototype examples of the general constructions.

2. Unit-derived

The unit-derived method for constructing and analysing linear block codes was initiated in [21–24] and continued in [13, 25] and elsewhere. Any linear block code over a field (see, for example, Proposition 2.1) can be constructed by the unit-derived method although this may not have been the original line of thought in the construction of the code. The unit-derived method gives further information on the code in addition to describing the generator and control matrices. Codes over a field formed by other methods may be given in a unit derived form from which further development can be performed.

A linear block code with generator matrix G and check matrix H is described by $GH^{\mathrm{T}} = 0$. The matrix H generates the dual of the code and the term 'control matrix' is also used for the 'check matrix' H^{T}. The basic unit-derived method is as follows: U is an invertible $n \times n$ matrix and is broken up as $U = \left(\begin{smallmatrix} A \\ B \end{smallmatrix}\right)$. The inverse of U has a compatible form $\left(\begin{smallmatrix} C & D \end{smallmatrix}\right)$ so that $\left(\begin{smallmatrix} A \\ B \end{smallmatrix}\right)\left(\begin{smallmatrix} C & D \end{smallmatrix}\right) = I_n$.

Now A has size $r \times n$ for some r and then B has size $(n-r) \times n$, C has size $n \times r$ and D has size $n \times (n-r)$. Then precisely $AC = I_r, AD = 0_{r\times(n-r)}, BC = 0_{(n-r)\times r}, BD = I_{(n-r)}$. So $AD = 0$ defines an

$[n, r]$ code C where A is the generator matrix, D is a control matrix and D^{T} generates the dual code of C.

A more general form of the unit-derived method (see [21–24]) is as follows: Given a unit matrix system $UV = I_n$, taking any r rows of U gives an $[n, r]$ code and a control matrix is obtained by eliminating the corresponding columns of V. Thus many codes may be derived, and codes of a particular type, from a single unit scheme.

The properties of the units are used to obtain properties of the codes, and units are formed with a particular type, length, rate, distance or field type in mind. Infinite families of required codes are also constructed and analysed; see for example the paper [17]. The number of choices of r rows from n is $\binom{n}{r}$, thus deriving many codes from a single unit scheme. Having a big choice is also useful in producing cryptographic schemes from large unit schemes.

In the basic unit scheme above A is the generator matrix of an $[n, r]$ code C and then D, which is a $(n - r) \times r$ matrix, is a check matrix for C. But note that B, C have been ignored! They can also be used to describe a 'complementary code' but even better can be used to form a convolutional code with A, D. Distances of the convolutional codes formed from a unit can often be determined in terms of a sum of the distances of linear codes formed from that unit. Convolutional codes have in addition their own efficient decoding algorithms, such as Viterbi algorithm and sequential decoding algorithm.

The matrices $\{A, B, C, D\}$ have full ranks as they are parts of invertible matrices.

That every linear block code over a field arises as a unit-derived code is shown in the following propositions.

Proposition 2.1. *Let C be a linear code over a field. Then C is equivalent to a unit-derived code.*

Proof. Assume C is an $[n, r]$ code with generator matrix A and check matrix H. Then $AH^{\mathrm{T}} = 0$ for an $r \times n$ matrix A, and an $(n - r) \times n$ matrix H; here $0 = 0_{r\times(n-r)}$. Let $\{e_1, e_2, \ldots, e_r\}$ be the rows of A which are linearly independent. Extend these to a basis $\{e_1, e_2, \ldots, e_r, e_{r+1}, \ldots, e_n\}$ for the whole space n-dimensional space. Let $B = \begin{pmatrix} e_{r+1} \\ \vdots \\ e_n \end{pmatrix}$ and $G = \begin{pmatrix} A \\ B \end{pmatrix}$. Then G is invertible with inverse given by $K = G^{-1} = \begin{pmatrix} C & D \end{pmatrix}$ where C is an $n \times r$ matrix and D is an $n \times (n - r)$ matrix. Thus $\begin{pmatrix} A \\ B \end{pmatrix}(CD) = I_n$. Then $AD = 0$. Now D^{T} has rank $(n - r)$ and H^{T} has rank $(n - r)$ and hence the code generated by A with check matrix H is equivalent to the code generated by A with check matrix D^{T}. (D and H^{T} generate the null space of $Ax = 0$ and have the same rank.) □

A code may not be originally constructed as a unit-derived code but it is useful to look at a code in this manner which leads to further and better constructions and in particular to constructions of convolutional codes. A code is a structures which is part of a bigger structure on which more is already known. Using the bigger structure to construct and analyse the embedded structures has many advantages. Multiple codes of a particular type may be deduced from just one unit.

If we require particular types of codes, as for example DC (including self-dual) codes or LCD codes, then look for particular types of units which give such codes in the unit-derived way.

The convolutional codes derived in [14] use unit-derived methods from Vandermonde/Fourier and other related matrices.

Unit-derived codes may also be obtained from a scheme where $UV = \alpha I_n, \alpha \neq 0$. The process is similar: Choose any r rows of U for a generator matrix and a check matrix is obtained by eliminating the corresponding columns of V. This is useful when considering Vandermonde/Fourier matrices, Propositions 2.11 and 2.12.

2.1. Using orthogonal units

Proposition 2.2. *Let U be an orthogonal matrix. Then any unit-derived block linear code from U is an LCD (linear complementary dual) code.*

Proof. Now $UU^{\mathrm{T}} = I_n$. Thus the unit scheme is $UU^{\mathrm{T}} = \binom{A}{B}(C\ D) = I_n$ for matrices A, B, C, D where A is of size $r \times n$, B is os size $(n-r) \times n$, C is of size $n \times r$ and D is of size $n \times (n-r)$. Denote the code generated by A by C. This code has control matrix D, which means $AD = 0$. Now $U^{\mathrm{T}} = (C\ D)$ and so $U = \binom{C^{\mathrm{T}}}{B^{\mathrm{T}}}$ giving that $C^{\mathrm{T}} = A, D^{\mathrm{T}} = B$. Thus $AD = 0$ is the same as $AB^{\mathrm{T}} = 0$.

Hence B generates the dual code of C. Now no non-trivial sum of rows of A can be a sum of rows of B as $U = \binom{A}{B}$ is non-singular. Hence $C \cap C^{\perp} = 0$ as required. □

The following proposition is shown in a similar manner to Proposition 2.2.

Proposition 2.3. *Let X be an $n \times n$ matrix such that $XX^T = \alpha I_n$, for $\alpha \neq 0$. Suppose X is broken as follow: $X = \binom{A}{B}$. From this it follows that $\binom{A}{B}(A^T\ B^T) = \alpha I_n$ where A has size $r \times n$. Then A generates an $[n, r]$ LCD code and B generates the dual of this code.*

Orthogonal matrices are thus a rich source for LCD codes. Given an orthogonal $n \times n$ matrix U any r rows of U may be chosen as the generator matrix for a $[n, r]$ code and this code is then an LCD linear block code.

An orthogonal matrix may also be used to form a self-dual code by combining it with an identity as described in the following Propositions 2.4 and 2.10. The extended Hamming $[8, 4, 4]$ and Golay $[24, 12, 8]$ codes are constructed in this way, see Examples 2.5 and 2.6 below. Other self-dual codes may be constructed in a similar manner from orthogonal matrices.

The following proposition is known but is given here in a form suitable for the constructions.

Proposition 2.4. *Let X be an orthogonal $n \times n$ matrix in a field of characteristic 2. Then the matrix $A = (I_n, X)$ generates a self-dual $[2n, n]$ matrix. Conversely if $A = (I_n, X)$ is a self-dual code where X is an $n \times n$ matrix in a field of characteristic 2, then X is orthogonal.*

Proof. Suppose X is orthogonal. Then $(I_n, X)\begin{pmatrix} I_n \\ X^{\mathrm{T}} \end{pmatrix} = I_n + XX^{\mathrm{T}} = I_n + I_n = O_{n\times n}$. Thus $\begin{pmatrix} I_n \\ X^{\mathrm{T}} \end{pmatrix}$, of rank n, is a control matrix for the code C generated by (I_n, X). Thus $(\begin{pmatrix} I_n \\ X^{\mathrm{T}} \end{pmatrix})^{\mathrm{T}} = (I_n, X)$ generates the dual code of C. Hence C is self-dual.

On the other hand if the code generated by (I_n, X) is self-dual then $(I, X)^{\mathrm{T}} = \begin{pmatrix} I_n \\ X^{\mathrm{T}} \end{pmatrix}$ is a control matrix for this code and so $(I_n, X)\begin{pmatrix} I_n \\ X^{\mathrm{T}} \end{pmatrix} = 0_{n\times n}$. Hence $I_n + XX^{\mathrm{T}} = 0_{n\times n}$ and so $XX^{\mathrm{T}} = I_n$. □

Example 2.5. *The Hamming $[8, 4, 4]$ self-dual binary code $\mathcal{H}$ is formed this way. Let $U = \begin{pmatrix} 0&1&1&1 \\ 1&1&1&0 \\ 1&1&0&1 \\ 1&0&1&1 \end{pmatrix}$, then $U^2 = I_4, U = U^T$ and $A = (I_4, U)$ is a generator matrix for the Hamming $[8, 4, 4]$ self-dual code $\mathcal{H}$. In addition the control matrix for the code has the form $\binom{I_4}{U}$ in which each row is unique and can be used to correct any one error in the one-error correcting code $\mathcal{H}$.*

Example 2.6. *The Golay* $[24, 12, 8]$ *is formed in this way, [32]. Let U be the reverse circulant matrix formed using* $(0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0)$ *as the first row. Then* $U^2 = I_{12}, U = U^T$ *and* (I_{12}, U) *is a generator matrix for the self-dual Golay* $[24, 12, 8]$ *code* $\mathcal{G}$*. See [32] for details. The control matrix for the code has the form* $\binom{I_{12}}{U}$*. The sum of any* 1, 2 *or* 3 *rows is unique and thus a lookup table can be formed to correct up to three errors in this 3-error correcting Golay* $[24, 12, 8]$ *self-dual code.*

A generator matrix for a linear block code may be given in systematic form, $G = (I_n, P)$ where P is an $n \times t$ matrix, see [3]. The distance of the code generated by (I_n, P) is a function of the 'unit-derived' type codes from P.

Proposition 2.7. *Consider the code* C *generated by* $G = (I_n, P)$*. Suppose the code generated by any* s *rows of* P *has distance* $\geq (d - s)$ *and for some choice of* r *rows the code generated by these* r *rows has distance exactly* $(d - r)$*; then the distance of* C *is* d*.*

The proposition makes sense even if the number of columns of P is less than n. The following Lemma is easy from results on fields.

Lemma 2.8. *Let* F *be a field. Then* F *has an square root of* (-1) *or else a quadratic extension of* F *has a square root of* (-1)*.*

Proof. If F does not have a square root of (-1) then $x^2 + 1$ is irreducible over F. □

Lemma 2.9. *Let* F *be a field with contains a square root of* (-1)*, denoted by* i*, and* X *an* $n \times n$ *matrix over* F*. Then* $XX^T = I_n$ *if and only if* $(iX)(iX)^T = -I_n$*.*

Proposition 2.4 is implicit in the following more general proposition which enables the construction of self-dual codes over fields.

Proposition 2.10. *Let* X *be an* $n \times n$ *matrix over a field* $\mathcal{F}$*.*

(i) If X *is an orthogonal matrix then* (I_n, iX) *generates a self-dual code where* i *is a square root of* (-1) *in* $\mathcal{F}$ *or in a quadratic extension of* $\mathcal{F}$*.*

(ii) If (I_n, X) *is self-dual then* iX *is orthogonal where* i *is a square root of* (-1) *in* $\mathcal{F}$ *or in a quadratic extension of* $\mathcal{F}$*.*

Proof. (i) $(I_n, iX)\begin{pmatrix} I_n \\ (iX)^{\mathrm{T}} \end{pmatrix} = I_n - XX^{\mathrm{T}} = 0$ and so (I_n, iX) generates a self-dual code as $(\begin{pmatrix} I_n \\ (iX)^{\mathrm{T}} \end{pmatrix})^{\mathrm{T}} = (I_n, iX)$.

(ii) Suppose (I_n, X) is self-dual. Then a control matrix of the code is $(I_n, X)^{\mathrm{T}} = \begin{pmatrix} I_n \\ X^{\mathrm{T}} \end{pmatrix}$ and so $(I_n, X)\begin{pmatrix} I_n \\ X^{\mathrm{T}} \end{pmatrix} = 0$. Hence $I_n + XX^{\mathrm{T}} = 0$ and so $XX^{\mathrm{T}} = -I_n$. Hence $iX(iX)^{\mathrm{T}} = I_n$. □

In a field of characteristic 2, $(-1) = 1$ and so Proposition 2.4 follows from Proposition 2.10.

This gives a general method for constructing and analysing self-dual codes from unit orthogonal and orthogonal like matrices.

2.2. Codes from Fourier type units

Using Fourier (or more generally Vandermonde) unit matrices to construct various types of MDS codes was initiated in [13, 25] and further developed in [17] and others. Here we present propositions in a very general form from which these constructions may be derived. Families of required types, lengths and rates are achievable.

Let F_n be a Fourier matrix over a finite field $\mathcal{F}$. Over which finite fields this F_n can be constructed is discussed in [17, 25] and elsewhere. Let F_n^* be the inverse of F_n giving the unit scheme $F_n F_n^* = I_n$. Let F_n have rows $\{e_0, e_1, \ldots, e_{n-1}\}$ in order and F_n^* have columns $\{f_0, f_1, \ldots, f_{n-1}\}$ in order. Now $e_1 = (1, \omega, \omega^2, \ldots, \omega^{n-1})$ and $e_i = (1, \omega^i, \omega^{2i}, \ldots, \omega^{(n-1)i})$ where ω is a primitive n^{th} root of unity in the field $\mathcal{F}$.

Then (see [13, 17, 25]) it is noted that $f_i = \frac{1}{n} e_{n-i}{}^{\mathrm{T}}$ and $e_i = n f_{n-1}{}^{\mathrm{T}}$.

Proposition 2.11. *Let F_n be a Fourier matrix over a finite field $\mathcal{F}$. Let F_n^* be the inverse of F_n giving the unit scheme $F_n F_n^* = I_n$. Let F_n have rows $\{e_0, e_1, \ldots, e_{n-1}\}$ in order where $e_i = (1, \omega^i, \omega^{2i}, \ldots, \omega^{(n-1)i})$ and ω is a primitive n^{th} root of unity in the field $\mathcal{F}$.*

Then the basic unit scheme from the Fourier matrix, $F_n F_n^* = I_n$, is given as follows:

$$\begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{pmatrix} \left(e_0{}^T, \quad e_{n-1}{}^T, \quad e_{n-2}{}^T, \quad \ldots, \quad e_1{}^T \right) = nI_n.$$

Having nI_n rather than I_n is no problem in describing codes from the scheme as $n \neq 0$ in a field in which the Fourier $n \times n$ matrix exists; H is a control, respectively generator, matrix if and only if αH is a control, respectively generator, matrix for $\alpha \neq 0$.

This gives the following, see for example [25]:

Proposition 2.12. *Let F_n be Fourier $n \times n$ matrix over a finite field and has rows $\{e_0, e_1, \ldots, e_{n-1}\}$. Suppose $F_n = \binom{A}{B}$ where $A = \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{r-1} \end{pmatrix}$.*

(i) The code generated by A is an $[n, r, n-r+1]$ MDS block code.

(ii) When $r > \frac{n}{2}$ the code generated by A is an $[n, r, n-r+1]$ DC MDS code.

(iii) In the case when $r > \frac{n}{2}$ the CSS construction from the DC $[n, r, n-r+1]$ code gives a quantum error-correcting $[[n, 2r-n, n-r+1]]$ which is an MDS quantum code[§].

If r rows of F_n are chosen in arithmetic order, starting at any row, with arithmetic difference k where $\gcd(k, n) = 1$ then an MDS $[n, r, n-r+1]$ is still obtained; the differences are taken mod n. This may be used to construct LCD codes. It will be shown later that DC convolutional codes may in many circumstances be produced from the unit scheme that produced the LCD codes. Examples are given as follows before the general result is described.

Example 2.13. *Let F_7 denote a Fourier 7×7 matrix over a finite field. Such a matrix exists over a field whose characteristic does not divide 7 and which has an element of order 7. Thus such a matrix exists for example over $GF(2^3)$ or over $GF(13^2)$. Look at a unit scheme formed by rearranging the rows of F_7 as follows:*

$$\begin{pmatrix} e_6 \\ e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{pmatrix} \left(e_1{}^T \; e_0{}^T \; e_6{}^T \; e_5{}^T \; e_4{}^T \; e_3{}^T \; e_2{}^T \right) = 7I_7.$$

[§] MDS in the quantum code sense

Then the first three rows $\begin{pmatrix} e_6 \\ e_0 \\ e_1 \end{pmatrix}$ *generates an* $[8, 3, 6]$ *MDS code. A control code is* $\begin{pmatrix} e_5{}^T & e_4{}^T & e_3{}^T & e_2{}^T \end{pmatrix}$ *and thus the dual code is generated by the transpose of this,* $\begin{pmatrix} e_5 \\ e_4 \\ e_3 \\ e2 \end{pmatrix}$*. Thus a* $[8, 3, 6]$ *LCD code is obtained; the* e_i *are independent as rows of an invertible matrix.*

Looking at the unit scheme as given, it will be shown later how a convolutional code which is DC can be constructed.

Example 2.14. *Let* F_8 *be a Fourier* 8×8 *matrix over a finite field. Such a matrix exists in a field of characteristic not dividing* 8 *and having an element of order* 8*, for example over* $GF(3^2)$ *or over the prime field* $GF(17)$*.*

The rows of F_8 *in order are denoted by* $\{e_0, e_1, \ldots, e_7\}$*. Look at the unit-scheme in the form*

$$\begin{pmatrix} e_6 \\ e_7 \\ e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{pmatrix} \begin{pmatrix} e_2{}^T & e_1{}^T & e_0{}^T & e_7{}^T & e_6{}^T & e_5{}^T & e_4{}^T & e_3{}^T \end{pmatrix} = 8I_8.$$

Then $\begin{pmatrix} e_6 \\ e_7 \\ e_0 \\ e_1 \\ e_2 \end{pmatrix}$ *generates an* $[8, 5, 4]$ *MDS code. A control matrix is* $\begin{pmatrix} e_5{}^T & e_4{}^T & e_3{}^T \end{pmatrix}$ *and thus the transpose of this,* $\begin{pmatrix} e_5 \\ e_4 \\ e_3 \end{pmatrix}$*, generates the dual of the code. Hence the code is an LCD code - the* e_i *are independent as rows of an invertible matrix.*

The idea is to keep a row e_i and its 'conjugate' e_{n-i} together in the generating matrix and thus get an LCD code. When using the same unit scheme to obtain convolutional codes, then DC convolutional codes are obtainable. The following is the general result which is best understood by looking at Examples 2.13 and 2.14.

Proposition 2.15. *Let* F_n *be a Fourier matrix over a finite field. Let the rows in order of* F_n *be denoted by* $\{e_0, e_1, \ldots, e_{n-1}\}$*. Rearrange the Fourier matrix unit scheme as follows:*

$$\begin{pmatrix} e_r \\ \vdots \\ e_{n-1} \\ e_0 \\ e_1 \\ \vdots \\ e_{n-r} \\ e_{n-r+1} \\ \vdots \\ e_{r-1} \end{pmatrix} (e_{n-r}{}^T, \ldots, e_1{}^T, e_0{}^T, e_{n-1}{}^T, \ldots, e_r{}^T, e_{r-1}{}^T, \ldots, e_{n-r+1}{}^T) = nI_n.$$

Then the code generated by $\begin{pmatrix} e_r \\ \vdots \\ e_{n-1} \\ e_0 \\ e_1 \\ \vdots \\ e_{n-r} \end{pmatrix}$ *is an LCD MDS code.*

An advantage of this is that looking at the full unit scheme allows the construction of convolutional DC codes and from the convolutional DC code quantum error-correcting codes are defined by the CSS construction.

From full unit schemes:

DC linear block codes from unit $\xrightarrow{unit-derived}$ convolutional LCD codes from the unit.

LCD linear block codes from unit $\xrightarrow{unit-derived}$ convolutional DC codes from the unit $\xrightarrow{unit-derived}$ convolutional quantum codes from the unit scheme.

3. Convolutional unit-derived codes

The basic unit-derived scheme is given by: $\begin{pmatrix} A \\ B \end{pmatrix}\begin{pmatrix} C & D \end{pmatrix} = I_n$.

As noted, using $AD = 0$ defines a block linear code where A generates the code and D^{T} generates the dual of this code. The total power of the unit is not used as $\{B, C\}$ are ignored. This can be rectified by going on to describe convolutional codes from the unit scheme. The general idea is to use A, B to describe convolutional codes $G(z) = A + Rz$ where R is formed from B. DC and LCD convolutional codes are obtained. Constructing DC convolutional codes lead to the construction of quantum error correcting convolutional (QECC) codes, by CSS construction.

A free distance can be prescribed as a linear functional of the distances of the (block linear) codes generated by A and B.

Convolutional MDS codes are constructed in [14, 17] by the method.

3.1. Same block sizes

Suppose that A, B both have the same size, $r \times n$, in the unit-derived formula; now $r = \frac{n}{2}$ and n is even. Let the code generated by A have distance d_1 and the code generated by B have distance d_2. Consider $G[z] = A + Bz$. This generates a convolutional code of memory 1. As $G(z) * C = AC = I_r$ the generator matrix has a right inverse and so the code is non-catastrophic.

Now also $(A + Bz)(D - Cz) = AD - ACz + BDz - BCz = -I_r z + I_r z = 0_r$. Thus $D - Cz$ is a control matrix for the $(n, r, r; 1)$ convolutional code.

The free distance for the code is $(d_1 + d_2)$. This minimum distance is obtained when the information vector has support 1. If the information vector $P(z)$ has support k then $P(z)G(z)$ has distance $\geq (d_1 + d_2 + k - 1)$.

The dual code generator matrix is obtained from the control matrix $H^{\mathrm{T}}(z)$; as noted the dual generator matrix is $H(z^{-1})z^m$ where m is the memory and $H^{\mathrm{T}}(z)$ is the control matrix. In this case the control matrix is $H^{\mathrm{T}}(z) = D - Cz$ and so a generator matrix for the dual code is $(D^{\mathrm{T}} - C^{\mathrm{T}}z^{-1})z = -C^{\mathrm{T}} + D^{\mathrm{T}}z$.

If $C^{\mathrm{T}} = -A$ and $D^{\mathrm{T}} = B$ then a self-dual convolutional code is obtained. Such a situation arises when $\begin{pmatrix} A \\ B \end{pmatrix}$ is orthogonal and the characteristic is 2.

As noted in Section 2.1, Lemma 2.8, a finite field F has a square root of (-1) or else a quadratic extension of F has a square root. Thus define $G(z) = A + iB$ where i denotes a square root of (-1). Then $(A + iB)(iD + Cz) = 0$ and so $H^{\mathrm{T}}(z) = iD + Cz$ is a control matrix giving that $H(z^{-1})z = C^{\mathrm{T}} + iD^{\mathrm{T}}$ is a dual matrix. In case U is an orthogonal matrix, $C^{\mathrm{T}} = A, D^{\mathrm{T}} = B$ and a dual matrix is $A + iB$ giving that $G(z)$ is a self-dual convolutional code of distance equal to the sum of the distances of the codes generated by A and by B.

Proposition 3.1. *Let U be a $2n \times 2n$ invertible matrix. Suppose $U = \begin{pmatrix} A \\ B \end{pmatrix}$ and $\begin{pmatrix} A \\ B \end{pmatrix}\begin{pmatrix} C & D \end{pmatrix} = I_{2n}$ where A and B have size $n \times 2n$ and C, D have size $2n \times n$.*

1) *A generates a* $[2n, n]$ *code* $\mathcal{C}$ *and* D^T *generates the dual code of* $\mathcal{C}$. *B generates a* $[2n, n]$ *code* $\mathcal{D}$ *and* C^T *generates the dual code of* $\mathcal{D}$.
2) $G(z) = A + Bz$ *generates a (non-catastrophic) convolutional* $(2n, n, n; 1)$ *code* C. *Then* $G(z)(D - Cz) = 0$, $D - Cz$ *is a control matrix of* C *and* $-C^T + D^T z$ *generates the dual code of* C.
3) *If the code generated by A has distance* d_1 *and the code generated by B has distance* d_2, *then the (free) distance of C is* $d = d_1 + d_2$ *and C is a* $(2n, n, n; 1, d)$ *convolutional code. Further if the information vector* $P(z)$ *has support k then* $P(z)G(z)$ *has distance* $\geq (d + k - 1)$.

Proposition 3.2. *Let U be a* $2n \times 2n$ *orthogonal matrix. Suppose* $U = \binom{A}{B}$ *and* $\binom{A}{B}(C\ D) = I_{2n}$ *where A and B have size* $n \times 2n$ *and* C, D *have size* $2n \times n$.

(1) *A generates a* $[2n, n]$ *code* $\mathcal{C}$ *and B generates the dual of* $\mathcal{C}$; *hence* $\mathcal{C}$ *is an LCD code. B generates a* $[2n, n]$ *code* $\mathcal{D}$ *and A generates the dual of* $\mathcal{D}$; *hence* $\mathcal{D}$ *is an LCD code.*
(2) $G(z) = A+Bz$ *generates a (non-catastrophic) convolutional* $(2n, n, n; 1)$ *code* C. *Then* $-C^T+D^Tz = -A + Bz$ *generates the dual code of* C.
(3) *If U is a matrix over a field of characteristic* 2 *then* $G(z) = A + Bz$ *generates a self-dual convolutional code.*
(4) *If the code generated by A has distance* d_1 *and the code generated by B has distance* d_2, *then the (free) distance of the code generated by* $G(z)$ *is* $d = d_1 + d_2$ *and is a* $(2n, n, n; 1, d)$ *convolutional code. Further if the information vector* $P(z)$ *has support k then* $P(z)G(z)$ *has distance* $\geq (d+k-1)$.
(5) *In case of characteristic* 2, *the code generated by* $G(z)$ *is used to generate a quantum convolutional code of memory* 1 *which has type* $[[2n, 0, d]]$ *where* $d = d_1 + d_2$ *is given in item 3.2.*
(6) *Define* $G(z) = A + iB$ *where i denotes a square root of* (-1). *Then* $(A + iB)(iD + Cz) = 0$ *and so* $H^T(z) = iD + Cz$ *is a control matrix giving that* $H(z^{-1})z = C^T + iD^Tz$ *is a dual matrix. The free distance of* $G(z)$ *is equal to the sum, d, of the distances of the codes generated by A and by B.*
(7) *Suppose now* $U = \binom{A}{B}$ *is an orthogonal matrix, then in item (6),* $C^T = A, D^T = B$. *A dual matrix is then* $A + iBz$ *giving that* $G(z)$ *is a self-dual convolutional* $(2n, n, n; 1, d)$ *code. This can be used to define a* $[[2n, 0, d]]$ *quantum convolutional code.*

These propositions are very general and can be used to construct infinite series of such codes with increasing distances.

3.2. Different block sizes

Consider cases where A has size greater than B in the unit-derived formula $\binom{A}{B}(C\ D) = I_n$ with $U = \binom{A}{B}$.

Let A have size $r \times n$, then B has size $(n - r) \times n$. Now $r > (n - r)$ is equivalent to $2r > n$. Let $t = r - (n - r) = 2r - n$ and $0_t = 0_{t \times n}$. Thus $B_1 = \begin{pmatrix} 0_t \\ B \end{pmatrix}$ is an $r \times n$ matrix. Now C is an $n \times r$ matrix and thus has the form $C = (X, C_1)$ where C_1 has size $n \times (n - r)$ and X has size $n \times (2r - n)$. As $AC = I_r$ then $AC_1 = \begin{pmatrix} 0_{(2r-n)\times(n-r)} \\ I_{(n-r)\times(n-r)} \end{pmatrix}$.

Define $G(z) = A + B_1 z$. This defines a generator matrix for a convolutional $(n, r, n - r; 1)$ code. $BC = 0$ implies $BC_1 = 0_{(n-r)\times(n-r)}$.

Then $(A + B_1z)(D - C_1z) = AD - AC_1z + B_1Dz - B_1C_1z^2 = 0_{r\times(n-r)} - \begin{pmatrix} 0_{2r-n\times n-r} \\ I_{n-r\times n-r} \end{pmatrix} z + \begin{pmatrix} 0_{2r-n\times n-r} \\ I_{(n-r)\times(n-r)} \end{pmatrix} z =$

$0_{r\times(n-r)}$. Thus the control matrix is $(D - C_1z)$ and a dual matrix is $-C^{\mathrm{T}} + Dz$ As $(A + B_1z)C = I_r$, the code is non-catastrophic.

Define $G(z) = A + iB_1z$ where i is a square root of (-1) in the field or in a quadratic extension of the field. Then $(A + iB_1)(iD + C_1z) = 0$ and ${C_1}^{\mathrm{T}} + iD^{\mathrm{T}}z$ is a dual matrix. In case U is orthogonal $C = A^{\mathrm{T}}, D = B^{\mathrm{T}}$ and $A = \begin{pmatrix} X^{\mathrm{T}} \\ {C_1}^{\mathrm{T}} \end{pmatrix}, B_1 = \begin{pmatrix} 0 \\ D^{\mathrm{T}} \end{pmatrix}$. Hence the code generated by $G(z)$ contains its dual.

Proposition 3.3. *Let U be a matrix unit over a field with $U = \begin{pmatrix} A \\ B \end{pmatrix}$ where A has size $r \times n$ and B has size $(n - r) \times n$ with $r > n - r$. Let $t = 2r - n$ and $B_1 = \begin{pmatrix} 0_{t\times n} \\ B \end{pmatrix}$. Then*

(1) $G(z) = A + B_1z$ generates a convolutional $(n, r, n - r; 1)$ code $\mathcal{C}$.

(2) Let A_1 be the matrix of the first $(2r-n)$ rows of A. The distance d of $\mathcal{C}$ is $\min\{d(A_1), d(A)+d(\begin{pmatrix} A_1 \\ B \end{pmatrix})\}$ *where $d(X)$ denotes the distance of the code generated by X.*

Proposition 3.4. *Let U be an orthogonal matrix in a field $\mathcal{F}$ and $U = \begin{pmatrix} A \\ B_1 \end{pmatrix}$ where A has size $r \times n$ and B_1 has size $(n - r) \times n$ with $r > n - r$. Let $t = 2r - n$ and $B_1 = \begin{pmatrix} 0_{t\times n} \\ B \end{pmatrix}$. Let i be a square root of -1 in F or in a quadratic extension of $\mathcal{F}$ Then*

(1) $G(z) = A + iB_1z$ generates a convolutional dual-containing $(n, r, n - r; 1)$ code $\mathcal{C}$.

(2) Let A_1 be the matrix of the first $(2r-n)$ rows of A. The distance d of $\mathcal{C}$ is $\min\{d(A_1), d(A)+d(\begin{pmatrix} A_1 \\ B \end{pmatrix})\}$ *where $d(X)$ denotes the distance of the code generated by X.*

(3) A quantum convolutional code of the form $[[n, 2r - n, d]]$ is constructed from $\mathcal{C}$ where d is the distance of $\mathcal{C}$.

The process is developed similarly by considering $\begin{pmatrix} A \\ B \end{pmatrix}\begin{pmatrix} C & D \end{pmatrix} = \alpha I_n$ with $U = \begin{pmatrix} A \\ B \end{pmatrix}$ where $\alpha \neq 0$.

It is best illustrated by looking at a block decomposition of the form $\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{pmatrix}\begin{pmatrix} B_0 & B_1 & B_2 & B_3 \end{pmatrix} = I_{4n}$, where each A_i has size $n \times 4n$. Take $G(z) = \begin{pmatrix} A_0 \\ A_1 \\ A_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ A_3 \end{pmatrix} z$.

In case U is orthogonal, let $G(z) = \begin{pmatrix} A_0 \\ A_1 \\ A_2 \end{pmatrix} + i\begin{pmatrix} 0 \\ 0 \\ A_3 \end{pmatrix} z$ and then a $(4n, 3n, n; 1, d)$ DC convolutional code is obtained from which a quantum convolutional code of form $[[4n, 2n, d]]$ is obtained. The d may be calculated algebraically and depends on the distances of codes formed from the blocks.

3.2.1. Prototype examples

The following prototype examples exemplify some of the general constructions. The examples given tend to be linear block and convolutional of types DC and LCD, but many types and infinite families of such codes may be built up by the techniques. From DC codes, quantum error correcting codes (QECC) may be formed and families of DC codes lead to the formation of families of quantum codes [¶].

Example 3.5. Let F_7 be a Fourier 7×7 matrix over some field $\mathcal{F}$. The field $\mathcal{F}$ is any field over which the Fourier 7×7 matrix exists. Thus $\mathcal{F}$ could be $GF(2^3)$, a characteristic 2 field, but also over fields with characteristic not dividing 7.

Denote the rows of F_7 in order by $\{e_0, e_1, e_2, e_3, e_4, e_5, e_6\}$ and the columns of the inverse of F_7 in order by $\{f_0, f_1, f_2, f_3, f_4, f_5, f_6\}$. Note that $e_if_j = 0, i \neq j, e_if_i = 1$ but also as F_7 is a Fourier matrix that $f_i^{\mathrm{T}} = \frac{1}{7}e_{7-i}$. The fraction part is no problem for check or control matrices: H is a check matrix if and only if αH is a check matrix for any $\alpha \neq 0$.

[¶]If the DC code has the form $[n, r, d]$ then the QECC formed has length n, distance d and dimension $(2r - n)$ – see [5, 6, 40].

Let $A = \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix}, B_1 = \begin{pmatrix} e_4 \\ e_5 \\ e_6 \end{pmatrix}$ where $F_7 = \begin{pmatrix} A \\ B_1 \end{pmatrix}$ is the Fourier 7×7 matrix. Now from [14] A generates a $[7, 4, 4]$ MDS (block) DC code.

Let $B = \begin{pmatrix} 0 \\ B_1 \end{pmatrix}$. Define $G(z) = A + Bz = A + \begin{pmatrix} 0 \\ e_4 \\ e_5 \\ e_6 \end{pmatrix} z$. Now $G(z) * (f_1, f_2, f_3, f_4) = I_4$ and so $G(z)$ has a right inverse and thus the code generated by $G(z)$ is non-catastrophic. $G(z)$ defines a $[7, 4, 3; 1]$ convolutional code C. The GSB for such a code is $(7 - 4)(\lfloor \frac{3}{4} \rfloor + 1) + 3 + 1 = 3 + 3 + 1 = 7$. It is easy to check that this 7 is the free distance of the code and so the code is an MDS convolutional code.

Note that $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) * B$ has distance ≥ 5 as an element in a $[7, 3, 5]$ code where $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ is a non-zero 1×4 vector, except when $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (\alpha_1, 0, 0, 0)$; but then $(\alpha_1, 0, 0, 0) * A = \alpha_1 e_0$ which has distance 7.

$G(z) * ((f_4, f_5, f_6) - (f_2, f_3, f_4)z) = 0$ and so $H^{\mathrm{T}}(z) = (f_4, f_5, f_6) - (f_2, f_3, f_4)z$ is a control matrix. Then $H(z^{-1})z = \begin{pmatrix} f_4^{\mathrm{T}} \\ f_5^{\mathrm{T}} \\ f_6^{\mathrm{T}} \end{pmatrix} z - \begin{pmatrix} f_1^{\mathrm{T}} \\ f_2^{\mathrm{T}} \\ f_3^{\mathrm{T}} \end{pmatrix}$ generates the dual matrix. Now since $f_i^{\mathrm{T}} = \frac{1}{7} e_{7-i}$ this means $7 * H(z^{-1})z = -\begin{pmatrix} e_6 \\ e_5 \\ e_4 \end{pmatrix} + \begin{pmatrix} e_3 \\ e_2 \\ e_1 \end{pmatrix} z$ generates the dual matrix. Thus C is a convolutional $(7, 4, 3; 1, 7)$ code which is an LCD code.

Example 3.6. *Use the same setup as in Example 3.5 where F_7 is a Fourier 7×7 matrix. Take $A = \begin{pmatrix} e_0 \\ e_1 \\ e_6 \\ e_2 \\ e_5 \end{pmatrix}$. Then it follows from [14] that the code generated by A is an LCD $[7, 5, 3]$ code. Let $B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ e_4 \\ e_3 \end{pmatrix}$.*

Define $G(z) = A + Bz = \begin{pmatrix} e_0 \\ e_1 \\ e_6 \\ e_2 \\ e_5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ e_4 \\ e_3 \end{pmatrix} z$. Then $(A + Bz) * (f_0, f_1, f_6, f_2, f_5) = I_5$ and so $G(z)$ has a right inverse and hence code, C, generated by $G(z)$ is non-catastrophic. Now $G(z) * \{(f_4, f_3) - (f_2, f_5)z\} = 0$ and so $(f_4, f_3) - (f_2, f_5)z = H^T(z)$ is a control matrix.

Now $H(z^{-1})z = \begin{pmatrix} f_4^T \\ f_3^T \end{pmatrix} z - \begin{pmatrix} f_2^T \\ f_5^T \end{pmatrix} = \frac{1}{7}\{-\begin{pmatrix} e_5 \\ e_2 \end{pmatrix} + \begin{pmatrix} e_3 \\ e_4 \end{pmatrix} z\}$. Thus $-\begin{pmatrix} e_5 \\ e_2 \end{pmatrix} + \begin{pmatrix} e_3 \\ e_4 \end{pmatrix} z$ generates the dual code of C.

If the field $\mathcal{F}$ has characteristic 2 the code C is dual containing. Thus when F_7 is a Fourier matrix over $GF(2^3)$, the code C is dual containing. It is easy to check directly that the free distance of C is 5 and is thus an MDS convolutional $(7, 5, 2; 1, 5)$ code; this is dual-containing when F_7 is a Fourier matrix over $GF(2^3)$.

If the field does not have characteristic 2 then define $G(z) = A + iBz$ where i is a square root of (-1) in the field or in a quadratic extension of the field. Then again a MDS convolutional dual-containing code is obtained. A quantum convolutional code is constructed from a dual-containing code.

The next example is a prototype example which although small demonstrates the power of the methods. In larger examples each row is replaced by a block of rows and lengths, distances are increased substantially.

Example 3.7. *Let $X = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ over $GF(2)$. Then $X^2 = I_4, X = X^T$. Thus DC codes are obtained by taking rows of X as a generating matrix and deleting corresponding columns of $X^T = X$ to obtain a control matrix.*

Let $A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ and $\begin{pmatrix} A \\ B \end{pmatrix} \begin{pmatrix} C & D \end{pmatrix} = I_4$ is our unit scheme. Now here $D = B^T, C = A^T$. Thus A generates a $[4, 2, 2]$ code C with control matrix $D = B^T$ and so B generates the dual code of C.

The code $\mathcal{C}$ is an $[4, 2, 2]$ *LCD code.*

Extend this to a convolutional code C using $G(z) = A + Bz$. Now $(A + Bz)(D + Cz) = 0$ so that $H^T(z) = D + Cz$ is a control matrix. Also $G(z) * C = I_2$ and so the code is non-catastrophic. The dual code of C is generated by $H(z^{-1})z = C^T + D^Tz = A + Bz$ and hence the code is self-dual. Thus a convolutional self-dual $(4, 2, 2; 1, 4)$ *is obtained. From this a quantum error-correcting code of form* $[[4, 0, 4]]$ *is obtained.*

$P(z)G(z)$ has distance $\geq 4 + (s - 1)$ for an information vector $P(z)$ of support s.

Any rows of U may be chosen to generate a code and the resulting code is automatically an LCD code. For example choose the first and third row of U and get $A = \begin{pmatrix} 0&1&1&1 \\ 1&1&0&1 \end{pmatrix}, B = \begin{pmatrix} 1&1&1&0 \\ 1&0&1&1 \end{pmatrix}$. Then $D = B^T, C = A^T$ similar to above. A then generates an LCD $[4, 2, 2]$ *and $G(z) = A + Bz$ generates a non-catastrophic self-dual* $(4, 2, 2; 1, 4)$ *convolutional code.*

This idea of choosing arbitrary rows, when used on large size matrices, lends itself to forming McEliece type of cryptographic systems.

Larger rates are obtained as follows. Choose three rows of U and get $A = \begin{pmatrix} 0&1&1&1 \\ 1&1&1&0 \\ 1&1&0&1 \end{pmatrix}, B = \begin{pmatrix} 1&0&1&1 \end{pmatrix}$. Then in general form $C = A^T, D = B^T$. A generates a $[4, 3, 1]$ *LCD code. Give U in its row form: $U = \begin{pmatrix} E_0 \\ E_1 \\ E_2 \\ E_3 \end{pmatrix}$. (In larger length constructions the E_i are blocks of matrices of size $n \times 4n$.) Then $A = \begin{pmatrix} E_0 \\ E_1 \\ E_2 \end{pmatrix}, B = \begin{pmatrix} E_3 \end{pmatrix}$.*

Define $G(z) = \begin{pmatrix} E_0 \\ E_1 \\ E_2 \end{pmatrix} + \begin{pmatrix} 0_4 \\ 0_4 \\ E_3 \end{pmatrix} z = A + B_1 z$, say; 0_4 indicates here a row of zeros of length 4. This gives a $(4, 3, 1; 1)$ convolutional code which is dual-containing. The distance is 2. Note that $(\alpha_1, \alpha_2, \alpha_3) * B_1 = \alpha_3 e_3$ has distance 3 except when $\alpha_3 = 0$ in which case $(\alpha_1, \alpha_2, \alpha_3)A = \alpha_1 e_0 + \alpha_2 e_1$ has distance 2 or 3. If $P(z)$ is an information vector of support s then $P(z)G(z)$ has distance $\geq 2 + (s - 1)$.

Example 3.8. *In another way construct rate $\frac{3}{4}$ and $\frac{1}{4}$ convolutional codes as follows:*

Define $G(z) = \begin{pmatrix} E_0 \\ E_1 \\ E_2 \end{pmatrix} + \begin{pmatrix} E_1 \\ E_0 \\ E_3 \end{pmatrix} z + \begin{pmatrix} E_2 \\ E_3 \\ E_0 \end{pmatrix} z^2 + \begin{pmatrix} E_3 \\ E_2 \\ E_2 \end{pmatrix} z^3$ and $H^T(z) = {E_3}^T + {E_2}^T z + {E_1}^T z^2 + {E_0}^T z^3$.

Then $G(z)H^T(z) = 0$. Let the code generated by $G(z)$ be denoted by C. The dual of C is generated by $H(z^{-1})z^3 = E_0 + E_1 z + E_2 z^2 + E_3 z^3$. Thus C is a dual containing $(4, 3, 9; 3)$ *convolutional code. Its free distance is* 4 *giving a* $(4, 3, 9; 3, 4)$ *convolutional dual-containing code. From this a quantum* $[[4, 2, 4]]$ *convolutional code is obtained.*

$P(z) * G(z)$ has distance $\geq 4 + (s - 1)$ when $P(z)$ is an information vector of support s.

Example 3.9. *Hamming convolutional code: Here the Hamming* $[7, 4, 3]$ *code is extended to a* $(7, 4, 3; 1, 6)$ *convolutional code. The distance is* 6 *which is twice that of the Hamming* $[7, 4, 3]$ *but it's also a convolutional code which has its own decoding techniques. The method is to look at the Hamming code as a unit-derived code and proceed from there by a general technique of constructing convolutional codes by the unit-derived method.*

The Hamming $[7, 4, 3]$ *is given with generator matrix $G = \begin{pmatrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{pmatrix}$ and check matrix $H = \begin{pmatrix} 1&1&0&1&1&0&0 \\ 1&0&1&1&0&1&0 \\ 0&1&1&1&0&0&1 \end{pmatrix}$. Now $\begin{pmatrix} G \\ H \end{pmatrix}$ is* not *invertible so this cannot be used for extending G to be a unit-derived code. For reasons that will appear later use $L = \begin{pmatrix} 1&1&1&1&1&1&1 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{pmatrix}$ as the generator matrix*[‖].

‖This is obtained by adding the other three rows to the first row to G above

Now complete L to a unit $U = \begin{pmatrix} 1&1&1&1&1&1&1\\ 0&1&0&0&1&0&1\\ 0&0&1&0&0&1&1\\ 0&0&0&1&1&1&1\\ 1&0&1&1&1&0&0\\ 0&1&0&0&1&1&1\\ 0&0&0&1&1&1&0 \end{pmatrix} = \binom{L}{K}$, *say. It is easy to check that K generates a* $[7,3,3]$ *code.*

U has inverse $V = \begin{pmatrix} 0&0&1&1&1&0&0\\ 1&1&0&1&1&1&1\\ 0&1&1&1&0&1&1\\ 1&1&0&0&1&0&1\\ 1&0&0&0&1&1&0\\ 0&1&0&0&0&1&0\\ 0&0&0&1&0&0&1 \end{pmatrix} = (\, C \;\; D\,)$, *where C is* 7×4 *and D is* 7×3. *Form* $G(z) = L + \binom{0}{K} z = \begin{pmatrix} 1&1&1&1&1&1&1\\ 0&1&0&0&1&0&1\\ 0&0&1&0&0&1&1\\ 0&0&0&1&1&1&1 \end{pmatrix} + \begin{pmatrix} 0&0&0&0&0&0&0\\ 1&0&1&1&1&0&0\\ 0&1&0&0&1&1&1\\ 0&0&0&1&1&1&0 \end{pmatrix} z.$

Precisely:

- $G(z)$ *generates a convolutional* $(7,4,3;1)$ *non-catastrophic code* $\mathcal{C}$.
- *The free distance of* $\mathcal{C}$ *is* 6 *so* $G(z)$ *generates a* $(7,4,3;1,6)$ *convolutional code.*
- *If* $P(z)$ *is an information vector then the distance of* $P(z) * G(z)$ *is* $\geq (6+d-1)$ *where d is the support of* $P(z)$.
- *The free distance may be shown from the following observations. Let* $(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$ *be a non-zero vector. Then* $(\alpha_1,\alpha_2,\alpha_3,\alpha_4) * \binom{0}{K}$ *has distance* 3 *except when* $\alpha_2 = 0 = \alpha_3 = \alpha_4$*; but in this case* $(\alpha_1,\alpha_2,\alpha_3,\alpha_4) * L = \alpha_1(1,1,1,1,1,1,1)$ *which has distance* 7.

Term the code generated by $G(z)$ *to be the* Hamming convolutional code.

4. Higher memory convolutional codes from units

The basic unit-derived scheme from $UV = I$ breaks $U = \binom{A}{B}$ to derive $\binom{A}{B}(\, C \;\; D\,)$ and linear convolutional codes of memory 1 from the scheme have been described and analysed.

Here the unit is broken into more than two blocks and linear convolutional codes of higher memory are derived and analysed.

Consider the case $UV = I$ with $U = \begin{pmatrix} A\\ B\\ C\\ D \end{pmatrix}$, $V = (\, E \;\; F \;\; G \;\; H\,)$ appropriately, giving another type of unit scheme:

$$\begin{pmatrix} A\\ B\\ C\\ D \end{pmatrix}(\, E \;\; F \;\; G \;\; H\,) = I.$$

First assume that A,B,C,D are of the same size. Thus U is a $4n\times 4n$ matrix. Three-quarter rate block linear codes are described by taking $\begin{pmatrix} A\\ B\\ C \end{pmatrix}$ as a generator matrix and then $(\, H\,)$ is a control matrix. More generally by choosing three of A,B,C,D to form the generator matrix for a code gives a $[4n,3n]$ three quarter rate code. The control matrix is immediately clear as one of E,F,G,H. When U is orthogonal, LCD block codes are obtained from which the convolutional codes described are DC when the characteristic is 2.

Memory 3 codes are described: $G(z) = A + Bz + Cz^2 + Dz^3$ is the generator of a $(4n,n,3n;3)$ code. This is non-catastrophic as $G(z)E = I_n$. The distance is d where d is a linear functional of the distances of the codes generated by A,B,C,D. Moreover $P(z)G(z)$ has distance $\geq (d+t-1)$ where $P(z)$ is an information vector and t is the support of $P(z)$.

Then $(A + Bz + Cz^2 + Dz^3)((F,G,H) - (E,H,G)z - (H,E,F)z^2 + (G,F,E)z^3 = 0$ and so $K^{\mathrm{T}}(z) =$

$(F, G, H) - (E, H, G)z - (H, E, F)z^2 + (G, F, E)z^3$ is a control matrix. The matrix of the dual is given by

$$K(z^{-1})z^3 = \begin{pmatrix} G^{\mathrm{T}} \\ F^{\mathrm{T}} \\ E^{\mathrm{T}} \end{pmatrix} - \begin{pmatrix} H^{\mathrm{T}} \\ E^{\mathrm{T}} \\ F^{\mathrm{T}} \end{pmatrix} z - \begin{pmatrix} E^{\mathrm{T}} \\ H^{\mathrm{T}} \\ G^{\mathrm{T}} \end{pmatrix} z^2 + \begin{pmatrix} F^{\mathrm{T}} \\ G^{\mathrm{T}} \\ H^{\mathrm{T}} \end{pmatrix} z^3.$$

This dual code is a $(4n, 3n, 9n; 3)$ code. When U is orthogonal, $A = E^{\mathrm{T}}, B = F^{\mathrm{T}}, C = G^{\mathrm{T}}, D = H^{\mathrm{T}}$.

Proposition 4.1. *Let* $\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \begin{pmatrix} E & F & G & H \end{pmatrix} = I_{4n}$ *be a unit scheme in which* $\{A, B, C, D\}$ *are of the same size. Then*

(i) $G(z) = A + Bz + Cz^2 + Dz^3$ *is a generator matrix of a* $(4n, n, 3n; 3)$ *convolutional code. The distance is a linear functional of the distances of the codes generated by* $\{A, B, C, D\}$.

(ii) $P(z)G(z)$ *has distance* $\geq (d + t - 1)$ *where t is the support of the information vector* $P(z)$.

(iii) The control matrix of C is $(F, G, H) - (E, H, G)z - (H, E, F)z^2 + (G, F, E)z^3$ *and the dual code of C is generated by* $\begin{pmatrix} G^T \\ F^T \\ E^T \end{pmatrix} - \begin{pmatrix} H^T \\ E^T \\ F^T \end{pmatrix} z - \begin{pmatrix} E^T \\ H^T \\ G^T \end{pmatrix} z^2 + \begin{pmatrix} F^T \\ G^T \\ H^T \end{pmatrix} z^3$.

(iv) When the full matrix is orthogonal the dual code of C is generated by

$$\begin{pmatrix} C \\ B \\ A \end{pmatrix} - \begin{pmatrix} D \\ A \\ B \end{pmatrix} z - \begin{pmatrix} A \\ D \\ C \end{pmatrix} z^2 + \begin{pmatrix} B \\ C \\ D \end{pmatrix} z^3.$$

(v) When the full matrix is orthogonal and the characteristic is 2 *the dual code is generated by* $\begin{pmatrix} C \\ B \\ A \end{pmatrix} + \begin{pmatrix} D \\ A \\ B \end{pmatrix} z + \begin{pmatrix} A \\ D \\ C \end{pmatrix} z^2 + \begin{pmatrix} B \\ C \\ D \end{pmatrix} z^3$. *In this case the dual code* $C^{\perp}$ *of C is a dual containing* $(4n, 3n, 9n; 3)$ *convolutional code. From this dual containing code of rate* $\frac{3}{4}$, *using the CSS construction, a quantum error correcting code of length* $4n$ *and rate* $\frac{1}{2}$ *is obtained.*

The Example 4.2 below is a very small prototype example with which to illustrate the general method.

Example 4.2. *Consider* $X = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ *over* $GF(2)$. *The matrix is orthogonal,* $XX^T = I_4$, *and also* $X = X^T$.[**] *Then* $XX^T = I$ *is broken up to give* $\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \begin{pmatrix} E & F & G & H \end{pmatrix} = I_4$ *where* $\{A, B, C, D\}$ *are row* 1×4 *vectors and* $E^T = A, F^T = B, G^T = C, H^T = D$ *as X is orthogonal. Define* $G(z) = A + Bz + Cz^2 + Dz^3$. *Then* $G(z)$ *generates a* $(4, 1, 1; 3, 12)$ *convolutional (non-catastrophic) code C. The distance is* 12 *as the code generated by each of* A, B, C, D *has distance* 3.

By Proposition 4.1 part (iv), the dual of C is generated by $\begin{pmatrix} C \\ B \\ A \end{pmatrix} + \begin{pmatrix} D \\ A \\ B \end{pmatrix} z + \begin{pmatrix} A \\ D \\ C \end{pmatrix} z^2 + \begin{pmatrix} B \\ C \\ D \end{pmatrix} z^3$.

Thus $C^{\perp}$ *is a dual-containing convolutional rate* $\frac{3}{4}$ *code of the form* $(4, 3, 9; 3)$. *From this a quantum error-correcting code of rate* $\frac{1}{2}$ *is formed.*

Example 4.3. *Golay binary code to convolutional rates* $\frac{3}{4}$ *and* $\frac{1}{4}$ *codes with memory* 3.

Consider the matrix X used in forming the self-dual Golay binary $[24, 12, 8]$ *code in the form* (I_{12}, X) *as in [32]. This X is the reverse circulant matrix with first row* $L = [0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0]$. *The X is symmetric and* $XX^T = X^2 = I_{12}$. *Here break X into four blocks,* X_1, X_2, X_3, X_4 *of equal size* 3×12. *The code generated by each* X_i *has distance* 5. *Then define* $G(z) = X_1 + X_2z + X_3z^2 + X_4z^3$ *and* $G(z)$ *generates a binary* $(12, 3, 9; 3, 20)$ *code C. Also* $P(z)G(z)$ *has distance* $\geq (20 + s - 1)$ *where* $P(z)$ *is an*

[**] (I_4, X) is a generator matrix for the extended Hamming $[8, 4, 4]$ code.

information vector and s is the support of $P(z)$. Note that since the rows of X are independent so any non-zero combination of the rows of $X_1 \cup X_2 \cup X_3 \cup X_4$ has distance ≥ 1.

As X is orthogonal the dual, $C^\perp$, of C is generated by $\begin{pmatrix} X_3 \\ X_2 \\ X_1 \end{pmatrix} + \begin{pmatrix} X_4 \\ X_1 \\ X_2 \end{pmatrix} z + \begin{pmatrix} X_1 \\ X_4 \\ X_3 \end{pmatrix} z^2 + \begin{pmatrix} X_2 \\ X_3 \\ X_4 \end{pmatrix} z^3$ by Proposition 4.1. Thus $C^\perp$ is a convolutional $(12, 9, 9; 3)$ code. It is seen that $C^\perp$ is a dual containing convolutional code of rate $\frac{3}{4}$ and is used to form a convolutional quantum error correcting code of rate $\frac{1}{2}$.

Example 4.4. *Let H be a Hadamard 12×12 matrix. Here the computer algebra system* GAP *[42] is used to generate H and the subsystem* GUAVA *is used to construct the codes and verify their distances in the linear block cases. The distances for the convolutional codes can be determined algebraically from the distances of the associated linear block codes.*

Thus H has the unit form $\binom{A}{B}(A^T \; B^T) = 12I_{12}$. In any field not of characteristic 2 or 3, A and B generate LCD codes.

- *Three rows of H generate a $[12, 3, 6]$ LCD code.*
- *Six rows generate a $[12, 6, 6]$ LCD code.*
- *Nine rows generate a $[12, 9, 2]$ LCD code.*
- *Let A be the first six rows of H and B be the last six rows of H. Define $G(z) = A + Bz$. Then $G(z)$ generates a $(12, 6, 6; 1, 12)$ convolutional code.*
- *Define $G(z) = A + iBz$ where i is a square root of (-1) in the field or in a quadratic extension of the field. Then $G(z)$ generates a self-dual convolutional $(12, 6, 6; 1, 12)$. From this a convolutional quantum code of type $[[12, 0, 12]]$ is formed.*
  $GF(5)$ has 2 as a square root of (-1) so over $GF(5)$, i can be taken to be 2. Arithmetic in $GF(5)$ is modular arithmetic. $GF(7)$ does not contain a square root of (-1) so it needs to be extended to $GF(7^2)$ in which to obtain a self-dual convolutional code.
- *Dual-containing convolutional codes of form $(12, 9, 3; 1, d)$ are obtained by letting A be the first nine rows of H and B the last three rows of H and defining $G(z) = A + iB_1 z$ where B_1 has first six rows consisting of zeros and last three rows consist of B. This gives rise to a quantum convolutional code of the form $[[12, 6, d]]$. The distance $d = 4$ but note that $P(z)G(z)$ has distance $\geq 5 + (s - 2)$ where s is the support of the information vector $P(z)$.*
- *Form (I_{12}, H). This is a $[24, 12, 8]$ code. Form (I_{12}, iH), where i is a square of (-1) in the field or in a quadratic extension of the field. This is a self-dual $[24, 12, 8]$ code.*
  In $GF(5)$ the element 2 is a square root of (-1) and thus $(I_{12}, 2X)$ gives a self-dual $[24, 12, 8]$ code in $GF(5)$. The field $GF(7)$ needs to be extended to $GF(7^2)$ and then a self-dual code is obtained.

4.1. Further block decomposition

1) Cases where the unit system is of the form A unit system $\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}(E \; F \; G \; H) = I$ where A, B, C have the same size $r \times n$ but D has size $s \times n$ with $s < n$ can be dealt in a similar but more complicated manner. The details are omitted.
2) Cases where the unit system is of size $3n \times 3n$ is dealt with by the following proposition:

   Proposition 4.5. *Let $\begin{pmatrix} A \\ B \\ C \end{pmatrix}(D \; E \; F) = I_{3n}$ be a unit scheme in which A, B, C are of the same size. Let $G(z) = A + Bz + Cz^2$. Then and then verifying that $(A + Bz + Cz^2)((E, F) - (D, F)z + (D, E)z^2 + (0, E - D)z^3) = 0$*

This allows the construction of rate $\frac{1}{3}$ and rate $\frac{2}{3}$ convolutional codes which are dual to one another similar to the method and results in Proposition 4.1.

3) A unit scheme which can be broken into blocks of 8 in an $8n \times 8n$ enables for example convolutional codes with memory 7 and rate $\frac{1}{8}$ and $\frac{7}{8}$ to be established. In special cases the $\frac{7}{8}$ rate code is dual containing establishing a rate $\frac{3}{4}$ quantum convolutional code of memory 8 similar to Proposition 4.1.
4) These types of constructions may be continued. For instance matrices with blocks of size n and matrix size $2^i n \times 2^i n$ are more amenable; when the matrix is orthogonal then dual-containing convolutional codes are obtainable from which quantum error correcting convolutional codes are formed with rate $\frac{2^{i-1}-1}{2^{i-1}}$. Details are omitted.

5. Low density parity check codes

A low density parity check, LDPC, code is one where the check/control matrix of the code has a small number of non-zero entries compared to its length, see for example [28].

The methods devised in previous sections for constructing linear block and convolutional codes are now used to construct LDPC linear and convolutional codes. What is required is the scheme produces a check/control matrix with low density compared to its length. It is known that for best performance of LDPC codes, there should be be no short cycles in the control matrix and this can be achieved by the methods.

Given a unit scheme $UV = I$ unit-derived codes are formed by taking any r rows of U as generator matrix and a check matrix is obtained by eliminating the corresponding columns of V. Thus if V itself is of low density then any such code formed is an LDPC code; if in addition V itself has no short cycles then any such code formed is an LDPC code with no short cycles.

Thus given a unit scheme $UV = I$, where V is of low density and has no short cycles, choose any r rows of U to form the generator matrix of an $[n, r]$ code C and deleting the corresponding r columns in V gives a check matrix for the code and the code C formed is an LDPC code with no short cycles in the check or control matrix. The code can also be specified by choosing the columns of the low density matrix V to form the control matrix and going to U to choose the rows which form a generator matrix.

This is done in [26] for linear block codes. Therein methods are derived using units in group rings to produce linear LDPC codes and to produce such LDPC codes with no short cycles in the the check matrices. Thus a unit system, $uv = 1$, is constructed in a group ring where one of the elements u, v, say v, has small support as a group ring element. The $uv = 1$ is then mapped to the corresponding matrix equation, $UV = I$, by a process given in [18], in which V has low density. Then using unit-derived codes leads to the construction of LDPC codes as required and when V has no short cycles it leads to the construction of LDPC codes with no short cycles in the check matrix. It can be ensured that V has no short cycles by a condition (see [26]) on the group elements with non-zero coefficient used in forming the group ring element v short cycles.

In that paper [26] simulations are made and the examples given are shown to outperform substantially previously constructed ones of the same size and rate. Randomly selected LDPC codes with no short cycles are produced from the same unit. The codes produced are of particular use in applications and industry where low storage and low power may be a requirement or necessary for better functioning.

Note that U, from which the generator matrix is derived, does not, necessarily, have low density which is good from the point of the minimum distance of the code; however as stated in Mackay [28], "Distance isn't everything".

Thus using group rings, systems are constructed $UV = I_n$ in which V has low density with no short cycles anywhere. This gives an enormous freedom in which to construct LDPC codes with no short cycles. Indeed eliminating any $(n - r)$ columns of V gives a control matrix, and the generator matrix is formed by using the rows from U corresponding to the eliminated columns of V; the result is an $[n, r]$ LDPC code. Thus given $UV = I_n$ where V has low density and no short cycles allows the construction of many LDPC codes with no short cycles.

In previous sections, methods are given for constructing convolutional codes from the unit-derived formula $\binom{A}{B}\begin{pmatrix} C & D \end{pmatrix} = I_n$ from $UV = I_n$ using all the components A, B, C, D. Convolutional codes of higher memory are obtained by further breaking up the unit system in blocks. These techniques may also be applied for constructing convolutional LDPC codes with no short cycles in the control matrix.

Considering the basic formula and when A, B have the same size, and n is even, then $G(z) = A + Bz$ generates a non-catastrophic convolutional code. The control matrix is $D - Cz$ and the dual code is generated by $C^{\mathrm{T}} - D^{\mathrm{T}}z$. If V is of low density and has no short cycles then $C^{\mathrm{T}} - D^{\mathrm{T}}z$ is of low density and has no short cycles. Thus the codes derived is a convolutional LDPC code with no short cycles.

It is difficult to describe explicitly the LDPC codes derived as for applications large lengths are required. Note that the method is very general and length and rate achieved can be decided in advance. We will concentrate on extending two of the examples in [26] to construct LDC convolutional codes with no short cycles.

Only basic information on group rings is required. A good nice book on group rings is [35] and also the basic information may be found online by a simple search.

Low density convolution codes and with no short in the control matrix are constructed by applying the methods in the previous sections together with the methods described in [26] for constructing LDPC linear codes with no short cycles. The following algorithm describes the constructions in general:

Algorithm 1. *(1) In a group ring with group size n find a unit and its inverse* $uv = 1$ *where v has small support and no short cycles. The size n of the group should be large and the support of v relatively small compared to n.*

(2) From $uv = 1$ *go over to the matrix embedding of the group ring in a ring of matrices of size* $n \times n$*, as in [18], to get a unit scheme* $UV = I_n$ *of matrices where V is of low density and has no short cycles.*

(3) Choose r columns of V to eliminate to form an $n \times (n - r)$ *matrix which will be a control matrix for a* $[n, r]$ *code. A generator matrix for this code is the* $r \times n$ *matrix formed by selecting the r rows from U corresponding in order to the r columns eliminated from V.*

(4) The unit scheme from item 1 may be presented as $\binom{A}{B}\begin{pmatrix} C & D \end{pmatrix}$ *where A has size* $r \times n$*, B has size* $(n - r) \times n$*, C has size* $n \times r$ *and D has size* $n \times (n - r)$*. Now here both C, D are of low density and have no short cycles. An LDPC code with no short cycles in the control matrix is given by* $AD = 0$ *as in [26]. But also notice* $BC = 0$ *gives a LDPC code in addition.*

(5) The unit scheme in item (4) as in previous sections is extended to $G(z) = A + Bz$ *when A, B have the same size (in which case the rate is* $1/2$*) or, when A has size heater than the size of B, to* $G(z) = A + B_1z$ *where* B_1 *is obtained by extending B with zero rows to be the size of A. Then* $G(z)$ *generates a convolutional memory* 1 *code which is non-catastrophic and has low density control*

matrix with no short cycles. The control matrix is $D - C_1 z$ *where* C_1 *is* C *or a submatrix of* C *as explained above.*

(6) *Obtaining memory greater than* 1 *from the unit matrix scheme* UV *derived from the group ring unit* uv *also follows in a similar manner as described earlier. Examples of such are given below.*

Examples must be of large length in order to satisfy the low density criterion. In general the examples in [26] are taken from unit-derived codes within $\mathbb{Z}_2(C_n \times C_4)$, where $\mathbb{Z}_2 = GF(2)$ is the field of two elements.

The matrices derived are then submatrices of circulant-by-circulant matrices and are easy to program. They are not circulant and thus are not cyclic codes. Having circulant-by-circulant rather than circulant allows a natural spreading of the non-zero coefficients and gives better distance and better performance.

Assume that C_n is generated by g and C_4 is generated by h. Every element in the group ring is then of the form:

$$\sum_{i=0}^{n-1} (\alpha_i g^i + h\beta_i g^i + h^2\gamma_i g^i + h^3\delta_i g^i),$$

with $\alpha_i, \beta_i, \gamma_i, \delta_i \in \mathbb{Z}_2$.

Example 5.1. *Examples of* [96, 48] *LDPC codes are given in Sections 3.2, 3.4 of [26]. These examples are derived from the group ring* $\mathbb{Z}_2(C_{24} \times C_4)$.

The check element $v = g^{24-9} + g^{24-15} + g^{24-19} + hg^{24-3} + hg^{24-20} + h^2 g^{24-22} + h^3 g^{24-22} + h^3 g^{24-12}$ *is used to define an LDPC linear code.*

Then v *has no short cycles in its matrix* V *and just* 8 *or less non-zero elements in each row and column. Any choice of columns of* V *will give an LDPC block linear code. A pattern to delete half the columns from the matrix* V *of* v *is chosen to produce a rate* 1/2 *code and is simulated and compared to other LDPC codes, outperforming these even when random columns are chosen.*

This selection is then presented as $\begin{pmatrix} A \\ B \end{pmatrix}\begin{pmatrix} C & D \end{pmatrix}$ *where* A *has size* 48×96, B *has size* 48×96, C *has size* 96×48 *and* D *has size* 96×48. *Then define* $G(z) = A + Bz$ *to obtain a convolutional* $[96, 48, 48; 1]$ *low density parity check code. The control matrix is* $D - Cz$ *which is* $D + Cz$ *as the characteristic is* 2.

In this case the matrix U *has the form* $\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}$ *and* V *has the form* $\begin{pmatrix} E & F & G & H \end{pmatrix}$ *where* A, B, C, D *have size* 24×96 *and* E, F, G, H *have size* 96×24 *and each of* E, F, G, H *have low density. Then as in Proposition 4.1,* $G(z) = A + Bz + Cz^2 + Dz^3$ *defines a convolutional* $(96, 24, 24 * 3; 1)$ *convolutional code which has low density check matrix.*

6. Conclusions

Using units to design and analyse both linear block and convolutional codes is a unique method in the important area of Coding Theory. The method is used to define codes to required length, rate and type by using appropriate units which give the required codes. Families of LCD (linear complementary dual) codes, families of self-dual codes, families of DC (dual-containing) codes, families of quantum codes and LDPC codes are designed from the method using appropriate units. The general method allow further new families of codes to be designed. New families of linear block codes, and with particular properties, can be designed from unit schemes; any linear block code over a field has such a designation. New families of convolutional codes of various memories can be mined from unit schemes; the question of which convolutional codes are unit-derived is left open.

Use of Generative-AI tools declaration

The author declares that he/she has not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The author declares no conflicts of interest in this paper.

References

1. P. Almeida, D. Napp, R. Pinto, A new class of superregular matrices and MDP convolutional codes, *Linear Algebra Appl.*, **439** (2013), 2145–2157. https://doi.org/10.1016/j.laa.2013.06.013
2. P. Almeida, D. Napp, R. Pinto, Superregular matrices and applications to convolutional codes, *Linear Algebra Appl.*, **499** (2016), 1–25. https://doi.org/10.1016/j.laa.2016.02.034
3. R. E. Blahut, *Algebraic codes for data transmission*, Cambridge: Cambridge University Press, 2003.
4. I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, Dual convolutional codes and the MacWilliams identities, *Probl. Inf. Transm.*, **48** (2012), 21–30. https://doi.org/10.1134/S0032946012010036
5. A. R. Calderbank, E. M. Rains, P. W. Shor, N. J. A. Sloane, Quantum error correction via codes over $GF(4)$, In: *Proceedings of IEEE international symposium on information theory*, Germany: IEEE, 1997. https://doi.org/10.1109/ISIT.1997.613213
6. A. R. Calderbank, P. W. Shor, Good quantum error-correcting codes exist, *Phys. Rev. A*, **54** (1996), 1098–1105. https://doi.org/10.1103/PhysRevA.54.1098
7. C. Carlet, S. Mesnager, C. Tang, Y. Qi, Euclidean and Hermitian LCD MDS codes, *Des. Codes Cryptogr.*, **86** (2018), 2605–2618. https://doi.org/10.1007/s10623-018-0463-8
8. C. Carlet, S. Mesnager, C. Tang, Y. Qi, R. Pelikaan, Linear codes over F_q are equivalent to LCD codes for $q > 3$, *IEEE Trans. Inform. Theory*, **64** (2018), 3010–3017. https://doi.org/10.1109/TIT.2018.2789347
9. C. Carlet, S. Mesnager, C. Tang, Y. Qi, New characterization and parametrization of LCD codes, *IEEE Trans. Inform. Theory*, **65** (2019), 39–49. https://doi.org/10.1109/TIT.2018.2829873
10. C. Carlet, Boolean functions for cryptography and error correcting codes, In: *Chapter of the monography Boolean models and methods in mathematics, computer science, and engineering*, Cambridge: Cambridge University Press, 2010, 257–397.
11. G. G. La Guardia, On negacyclic MDS-convolutional codes, *Linear Algebra Appl.*, **448** (2014), 85–96. https://doi.org/10.1016/j.laa.2014.01.033

12. H. Gluesing-Luerssen, G. Schneider, A MacWilliams identity for convolutional codes: The general case, *IEEE Trans. Inf. Theory*, **55** (2009), 2920–2930. https://doi.org/10.1109/TIT.2009.2021302
13. T. Hurley, D. Hurley, B. Hurley, Quantum error-correcting codes: The unit-derived strategy, *arXiv:1805.09053*, 2018. https://doi.org/10.48550/arXiv.1805.09053
14. T. Hurley, Linear block and convolutional MDS codes to required rate, distance and type, In: *Lecture notes in networks and systems*, Cham: Springer, **507** (2022), 129–157. https://doi.org/10.1007/978-3-031-10464-0_10
15. T. Hurley, Convolutional codes from unit schemes, *arXiv:1412.1695*, 2018. https://doi.org/10.48550/arXiv.1412.1695
16. T. Hurley, On codes induced from Hadamard matrices, *arXiv:2410.24027*, 2024. https://doi.org/10.48550/arXiv.2410.24027
17. T. Hurley, D. Hurley, B. Hurley, Maximum distance separable codes to order, *arXiv:1902.06624*, 2019. https://doi.org/10.48550/arXiv.1902.06624
18. T. Hurley, Group rings and rings of matrices, *Int. J. Pure Appl. Math.*, **31** (2006), 319–335.
19. T. Hurley, Solving underdetermined systems with error-correcting codes, *Int. J. Inf. Coding Theory*, **4** (2017), 201–221. http://dx.doi.org/10.1504/IJICOT.2017.10005825
20. T. Hurley, Convolutional codes from units in matrix and group rings, *Int. J. Pure Appl. Math.*, **50** (2009), 431–463.
21. P. Hurley, T. Hurley, Module codes in group rings, In: *2007 IEEE International symposium on information theory*, France: IEEE, 2007, 1981–1985. https://doi.org/10.1109/ISIT.2007.4557511
22. P. Hurley, T. Hurley, Codes from zero-divisors and units in group rings, *Int. J. Inform. Coding Theory*, **1** (2009), 57–87. https://doi.org/10.1504/IJICOT.2009.024047
23. P. Hurley, T. Hurley, Block codes from matrix and group rings, In: *Selected topics in information and coding theory*, World Scientific, 2010, 159–194. https://doi.org/10.1142/9789812837172_0004
24. P. Hurley, T. Hurley, LDPC and convolutional codes from matrix and group rings, In: *Selected topics in information and coding theory*, World Scientific, 2010, 195–237. https://doi.org/10.1142/9789812837172_0006
25. T. Hurley, D. Hurley, Coding theory: The unit-derived methodology, *Int. J. Inf. Coding Theory*, **5** (2018), 55–80. http://dx.doi.org/10.1504/IJICOT.2018.10013082
26. T. Hurley, P. McEvoy, J. Wenus, Algebraic constructions of LDPC codes with no short cycles, *Int. J. Inf. Coding Theory*, **1** (2010), 285–297. https://doi.org/10.1504/IJICoT.2010.032544
27. R. Johannesson, K. Sh. Zigangirov, *Fundamentals of convolutional coding*, Wiley-IEEE Press, 2015.
28. D. J. C. MacKay, *Information theory, inference, and learning algorithms*, Cambridge: Cambridge University Press, 2003.
29. F. J. MacWilliams, N. J. A. Sloane, *The theory of error-correcting codes*, Elsevier, 1977.
30. J. L. Massey, Linear codes with complementary duals, *Discrete Math.*, **106-107** (1992), 337–342. https://doi.org/10.1016/0012-365X(92)90563-U

31. J. L. Massey, Reversible codes, *Inf. Control*, **7** (1964), 369–380. https://doi.org/10.1016/S0019-9958(64)90438-3

32. I. McLoughlin, T. Hurley, A group ring construction of the extended binary Golay code, *IEEE Trans. Inform. Theory*, **54** (2008), 4381–4383. https://doi.org/10.1109/TIT.2008.928260

33. R. McEliece, *The theory of information and coding*, 2 Eds., Cambridge: Cambridge University Press, 2009. https://doi.org/10.1017/CBO9780511606267

34. S. Mesnager, C. Tang, Y. Qi, Complementary dual algebraic geometry codes, *IEEE Trans. Inf. Theory*, **64** (2018), 2390–2397. https://doi.org/10.1109/TIT.2017.2766075

35. C. P. Milies, S. K. Sehgal, An introduction to group rings, In: *Algebra and applications*, Dordrecht: Springer, **1** (2002).

36. R. Pellikaan, On decoding by error location and dependent sets of error positions, *Discrete Math.*, **106-107** (1992), 369–381. https://doi.org/10.1016/0012-365X(92)90567-Y

37. J. M. M. Porras, J. A. D. Pérez, J. I. I. Curto, J. S. Sotelo, Convolutional Goppa codes, *IEEE Trans. Inform. Theory*, **52** (2006), 340–344. https://doi.org/10.1109/TIT.2005.860447

38. J. Rosenthal, R. Smarandache, Maximum distance separable convolutional codes, *Appl. Algebra Engrg. Comm. Comput.*, **10** (1999), 15–32. https://doi.org/10.1007/s002000050120

39. J. Rosenthal, Connections between linear systems and convolutional codes, In: *The IMA volumes in mathematics and its applications*, New York: Springer, **123** (1999). https://doi.org/10.1007/978-1-4613-0165-3_2

40. A. M. Steane, Simple quantum error correcting codes, *Phys. Rev. A*, **54** (1996), 4741–4751.

41. R. Smarandache, H. Gluesing-Luerssen, J. Rosenthal, Constructions for MDS-convolutional codes, *IEEE Trans. Inform. Theory*, **47** (2001), 2045–2049. https://doi.org/10.1109/18.930938

42. The GAP group, GAP – groups, algorithms, and programming, Version 4.12.2, 2022. Available from: `https://www.gap-system.org`